Université Lumière Lyon2
Année 2003

Thèse
pour obtenir le grade de
Docteur
en
Informatique

présentée et soutenue publiquement par

Pierre-Emmanuel JOUVE
le 10 décembre 2003

Apprentissage Non Supervisé et Extraction de Connaissances à partir de Données

préparée au sein du laboratoire ERIC
Equipe de Recherche en Ingénierie des Connaissances

sous la direction de
Nicolas Nicoloyannis

devant le jury, composé de:

Jean-Paul Rasson, Rapporteur	Professeur, Facultés Universitaires N.D. de la Paix, Namur
Gilles Venturini, Rapporteur	Professeur, Université de Tours
Mohand-Saïd Hacid, Examinateur	Professeur, Université Claude Bernard-Lyon 1
Michel Lamure, Examinateur	Professeur, Université Claude Bernard-Lyon 1
Gilbert Ritschard, Examinateur	Professeur, Université de Genève
Nicolas Nicoloyannis, Directeur de thèse	Professeur, Université Lumière-Lyon 2

Remerciements

Ce travail a commencé –et continuera je l'espère– avec Nicolas NICOLOYANNIS. Le hasard a bien voulu que je le rencontre il y a maintenant plus de 5 ans. De cette rencontre et de différentes circonstances a germé l'idée puis le projet de mes études doctorales auxquelles je ne me destinais pas forcément. Ses qualités humaines et scientifiques, son soutien, son amitié, ..., m'ont permis de mener à bien et avec grand plaisir et liberté ces travaux et je l'en remercie très chaleureusement.
MERCI NICOLAS !

Bien d'autres personnes ont contribué à rendre possible ce projet, je pense notamment ici à Ricco RAKOTOMALALA et Djamel ZIGHED qui ont guidé mes premiers pas au laboratoire ERIC ; Michel LAMURE et les membres du département Recherche et Technologie de la région Rhône-Alpes, le premier pour m'avoir fait confiance et intégré au sein d'un projet de recherche soutenu par la région et les seconds pour m'avoir accordé un financement de thèse.

De ces années, je retiendrai, outre le plaisir de la recherche, le bonheur d'avoir rencontré de nouveaux amis : Gaëlle et Laurent à qui ce travail doit énormément –enfin surtout à Gaëlle parce que Laurent...–.
GAELLE, LAURENT MERCI !

Je tiens également à souligner combien il fut agréable de parcourir ce chemin au sein du laboratoire ERIC dont je remercie l'ensemble des membres, et plus particulièrement les adeptes de discussions footballistiques, les buveuses et buveurs de thé ou café, ainsi que Astrid, Valérie et Lydie qui m'ont facilité certaines démarches.

D'un point de vue scientifique et humain, je voudrais rappeler le plaisir et l'honneur que m'ont fait d'accepter d'être membres de mon jury de thèse les Professeurs Mohand-Saïd HACID, Michel LAMURE, Gilbert RITSCHARD, Jean-Paul RASSON et Gilles VENTURINI. Je remercie les rapporteurs Jean-Paul RASSON et Gilles VENTURINI pour l'oeil à la fois critique et bienveillant qu'ils ont bien voulu porter sur mes travaux. Je tiens à exprimer tout particulièrement ma reconnaissance envers :

– Jean-Paul RASSON, pour la précision de sa lecture de mon travail et la justesse des remarques qu'il m'a transmises ;

– Gilbert RITSCHARD, pour l'extrême et stupéfiante précision de sa lecture de mon travail ainsi que pour la justesse des remarques qu'il m'a transmises bien qu'il n'ait pas été rapporteur.

Vos remarques m'ont permis de poursuivre ma réflexion et d'améliorer ce document, même s'il est certain qu'il me faudrait un temps considérable pour pouvoir exploiter entièrement votre travail.

Je voudrais maintenant remercier de tout mon cœur Maman, Papa, Philippe et Emilie puisque je leur dois ce que je suis et bien plus encore...
MERCI, MERCI, MERCI ET ENCORE MERCI !

Bien que n'ayant pas véritablement contribué à un avancement raisonné de mes travaux, je remercie l'ensemble de mes amis pour avoir assuré "la préservation de ma santé mentale" (bien que...) et permis de vivre d'excellents moments. MERCI A TOUS !

Je voudrais également avoir ici une pensée des plus affectueuses pour mes grands-parents.

Enfin, terminons avec grâce et beauté : Karen ; Karen je te remercie le plus amoureusement possible pour tout ton amour, tout ce que tu m'apportes et m'apprends... KAREN, MERCI !

Je tiens finalement à demander des excuses pour ceux que j'oublie ou ne cite pas nommément, pour mon incapacité à remercier en mesure de ce qui m'a été donné, et pour ceux dont la lecture de cette dissertation ne constituera pas un moment agréable ou utile...

Voilà, Merci à tous, et aux autres !

Pierre,

Grand Croix, Janvier 2004

Résumé

Les travaux constituant cette dissertation concernent la classification non supervisée. Cette problématique, commune à de multiples domaines (et ainsi connue sous diverses acception : apprentissage/classification non supervisé(e) en reconnaissance de formes, taxonomie en sciences de la vie, typologie en sciences humaines...), est ici envisagée selon la perspective Ingénierie des Connaissances et plus spécifiquement dans le cadre de son intégration au sein du processus d'Extraction de Connaissances à partir de Données (ECD).

D'une part, nos travaux participent à l'*amélioration du processus de classification non supervisée*, et ce, selon divers axes propres ou non à l'ECD (coût calculatoire et utilisabilité des méthodes, formes et distribution des données traitées, forme des connaissances extraites, sélection de variables pour l'apprentissage non supervisé...) mais aussi à l'*évaluation de la qualité d'un processus de classification non supervisée* (estimation de la validité des résultats issus du processus). D'autre part ces travaux visent à *illustrer le lien très étroit unissant apprentissage non supervisé et apprentissage supervisé* et à montrer l'*intérêt d'une interaction entre ces deux types de processus*.

Concrètement, ces divers problèmes sont abordés et présentés au travers d'une nouvelle méthode de classification non supervisée, de deux nouveaux indices et d'une méthodologie dédiés à l'évaluation/comparaison de la validité de classification non supervisée, de méthodes de sélection de variables pour l'apprentissage non supervisé et l'apprentissage supervisé, de plusieurs méthodes pour l'agrégation de classifications non supervisées.

MOTS CLÉS : ECD, Apprentissage Non Supervisé/Supervisé/Semi-Supervisé, Sélection de Variables, Agrégation de Modèles...

Abstract

This dissertation deals with clustering. This problem, which is common to many fields (and thus may be found under different names such as : unsupervised learning in pattern recognition, taxonomy in life sciences, typology in human sciences..), is considered here through Knowledge Engineering perspective. More specifically, we consider clustering as an integrated step of a Knowledge Discovery in Databases (KDD) process.

On the one hand, our work contributes to *clustering process enhancement* according to several axis (computational cost and usability of clustering algorithms, type and distribution of treated data, materialization of extracted knowledge, feature selection for clustering...) but it also contributes to *Clustering Quality Checking* (Clustering Validity Checking). On the other hand, our work aims at *illustrating the link between supervised and unsupervised learning* and showing that *the interaction between these two kinds of learning is largely profitable*.

These problems are treated through the presentation of a new clustering method, as well as two new indices and a methodology dedicated to clustering validity comparison/assesment, two new methods for feature selection (in supervised learning and clustering contexts), and finally several methods for clustering combinations.

KEYWORDS : KDD, Clustering, Supervised Learning, Semi-Supervised Learning, Feature Selection, Clustering Combinations, Cluster Ensembles...

Table des matières

1 Introduction, Préambule

"Le meilleur (...) ce n'est pas le mal réel qu'on se donne pour accoler le mot au mot, pour entasser brique sur brique ; ce sont les préliminaires, le travail à la bêche que l'on fait en silence en toutes circonstances, que ce soit dans le rêve ou à l'état de veille. Bref, la période de gestation. Personne n'a jamais réussi à jeter sur le papier ce qu'il avait primitivement l'intention de dire (...)"

- Henry Miller -
Sexus, (édition Buchet-Chastel) (1949)

Le traitement automatique de l'information fait appel à différentes ressources techniques, technologiques et théoriques issues de domaines variés tels que l'informatique, l'intelligence artificielle, la statistique, la théorie des probabilités, l'analyse de données, l'optimisation... La fin des années 80 et le début des années 90 ont vu un faisceau de situations favorables[1] à l'émergence d'un domaine de recherche transdisciplinaire s'attachant spécifiquement au traitement de vastes volumes d'information et à leur valorisation sous forme de connaissances : le "Knowledge Discovery in Databases" [PSCF+89] (expression que la communauté scientifique francophone a traduit plus tard par Extraction de Connaissances à partir de Données (ECD)).

Une quinzaine d'années plus tard, la communauté mondiale en ECD s'est élargie, structurée et a très largement essaimé dans le monde industriel. L'interaction entre la recherche et l'industrie explique certainement d'ailleurs, la rapidité de cette croissance, tout comme l'enthousiasme et l'activité de cette jeune communauté ont également contribué à cette emergence.

Plus encore, nous pensons que l'agitation, l'ébullition autour du phénomène ECD provient de l'adéquation des problématiques industrielles et académiques : ainsi les promesses de l'ECD en terme de valorisation de l'information ne pouvaient laisser insensibles les acteurs industriels au moment où l'information apparaît comme un élément stratégique déterminant.

1. sur le plan technologique (avancées technologiques en informatique : accroissement des capacités de stockage et de calculs), économique (passage de l'économie de l'ère post-industrielle à l'ère informationnelle)... et peut être épistémologique : le traitement massif et informatisé de l'information souffre alors un peu moins de critiques touchant à la rigueur de cette approche

On peut également chercher des explications non conjoncturelles à ce succès comme la transdisciplinarité d'un domaine qui se fait fort de mettre à profit chacune de ses composantes ainsi que les synergies pouvant exister entre elles. Surtout, l'ECD, comme le rappelle sa définition francophone[2], est un processus anthropocentré tirant profit d'une interaction entre l'homme et la machine. Placer l'humain au centre du processus relève tout d'abord d'un intérêt pragmatique : les limites actuelles des systèmes automatiques peuvent être repoussées par utilisation de l'intelligence et de l'expertise humaine. On peut également se hasarder à avancer un intérêt "psychologique" : un utilisateur intégré au processus de traitement de l'information, non dépossédé de ses capacités d'analyse tend à mieux accepter, comprendre le processus d'ECD.

La tendance actuelle de la communauté ECD à se pencher sur ses échecs passés (afin de les féconder et de préparer les succès futurs...?) [LMF02] souligne d'ailleurs la grande nécessité d'intégrer l'homme au sein des systèmes et processus d'ECD.

Les travaux présentés dans cette thèse s'inscrivent au sein de la problématique ECD.

Aussi, l'intégration de l'expertise et des connaissances humaines, la définition de méthodes "permettant un échange", une interaction entre homme et machine ainsi que la prise en compte des contraintes d'utilisabilité pour les méthodes développées constituent des exigences fondamentales sous-jacentes aux travaux que nous présentons ici.

Préalablement évoquée, la relation forte entre recherche et industrie permet et nécessite la prise en compte des besoins issus de la pratique (tels que la limitation des coûts calculatoires, la limitation des coûts de stockage, la conformité à des exigences issues de la distribution de l'information, etc...). Proposer des solutions ne dérogeant pas non plus à ces nécessités constitue la deuxième exigence que nous imposons à nos travaux.

Plus précisément, il s'agit dans ce document de présenter un ensemble de contributions pour l'apprentissage non supervisé. De manière plus détaillée, nous tentons d'apporter ici un ensemble de nouvelles solutions pour l'intégration de l'apprentissage non supervisé au sein d'un processus ECD[3].

2. L'ECD est définie par la communauté scientifique francophone comme le processus itératif, anthropocentré (interactif), non trivial d'identification de connaissances valides, nouvelles, potentiellement utiles et intelligibles au sein d'un ensemble de données.

3. Le processus ECD est schématisé sur la figure 1.1. Ce processus comprend classiquement les étapes suivantes :

   1. comprehension du domaine étudié
   2. création et nettoyage de l'ensemble des données à traiter (Sélection, Construction de Variables....)

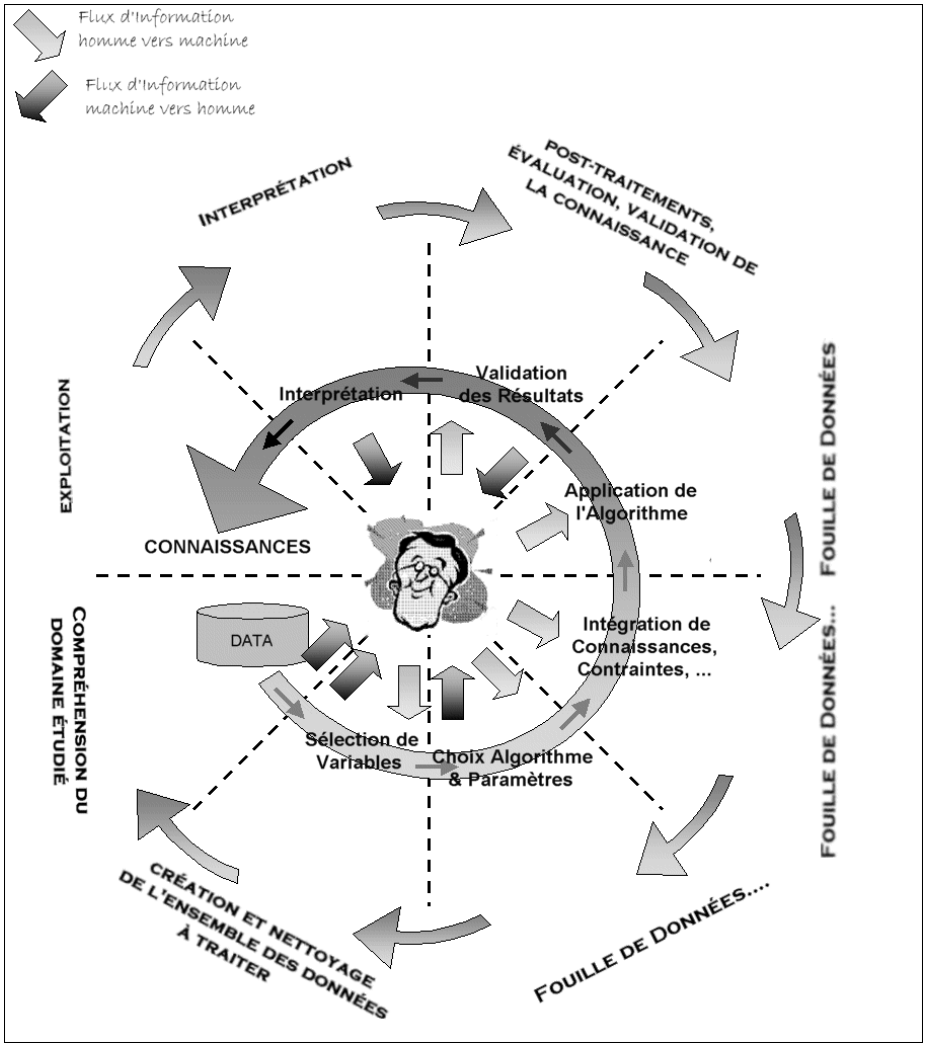


FIG. 1.1 –: *Eléments du Processus ECD*

Ces solutions concernent les différentes étapes clés de ce processus, à savoir, la sélection de variables (chapitre 5), l'application de la méthode d'apprentissage non supervisé 3, la validation / l'estimation de la qualité d'un modèle d'apprentissage non supervisé 4. Un chapitre est également consacré à l'agrégation de modèles d'apprentissage non supervisé et aux différents intérêts que

3. extraction des régularités cachées dans les données et formulation des connaissances mises à jour sous forme de modèles ou de règles (cette étape dans le processus global d'ECD est habituellement désignée sous le nom de fouille des données)
4. post-traitements, évaluation, validation de la connaissance découverte
5. interprétation des résultats
6. exploitation des résultats

cela revêt dans le cadre de la prise en compte des deux niveaux d'exigence évoqués plus tôt.

Notons enfin que le choix de l'apprentissage non supervisé s'explique, d'une part, par l'intégration de ces travaux au sein du projet de recherche universitaire BC3[4] dont l'un des objectifs était la mise au point de méthodes de fusion de données (l'apprentissage non supervisé constituant une piste envisagée), et d'autre part car nous pensons que l'apprentissage non supervisé occupe une place particulière et centrale au sein du processus ECD. Ainsi, la présentation de nos travaux sur l'apprentissage non supervisé est complétée par différentes contributions pour l'apprentissage supervisé exploitant justement nos contributions pour l'apprentissage non supervisé.

Ce document s'organise de la manière suivante : le premier chapitre introduit les éléments nécessaires pour la lecture du reste du document ; les chapitres 3 (présentation d'une nouvelle méthode d'apprentissage supervisé) et 4 (présentation d'une nouvelle méthodologie pour l'évaluation de la validité d'un modèle d'apprentissage non supervisé) peuvent être abordés de manière indépendante des autres alors que la lecture du chapitre 5 (présentation de méthodes de sélection de variables pour l'apprentissage supervisé et non supervisé) et du chapitre 6 (présentation de méthodes d'agrégation de modèles d'apprentissage non supervisé) nécessite respectivement la lecture préalable des chapitres 4 et 3. Notons enfin, que contrairement à la majorité des thèses,

4. Le projet BC3 (Projet Base de Connaissances Cœur-Cerveau, $http : //kbbrain.free.fr$) initié par différentes équipes universitaires et hospitalières vise à la création d'une base de connaissances (BDC) sur les pathologies organiques touchant deux organes vitaux : le cœur et le cerveau. Une telle BDC a pour objectif premier de servir de support pour le recueil et la conservation de données médicales, permettant ainsi la capitalisation de l'expérience et des connaissances de chercheurs et cliniciens des domaines concernés. La métaphore d'une encyclopédie médicale numérique ayant pour sujet les 2 organes vitaux que constituent le cœur et le cerveau peut dans un premier temps être adoptée pour décrire cet outil. Les travaux que nous menons dans le cadre de ce projet impliquent toutefois d'étendre cette métaphore à celle d'une encyclopédie stockant non seulement de l'information mais se voulant également "génératrice" de connaissances. Nous proposons en effet d'intégrer des méthodes de traitement de l'information à la BDC de manière à permettre l'exploitation de l'information stockée dans cette base et la découverte de nouvelles connaissances.
   Equipes Universitaires :
   - CREATIS : Centre de Recherche et d'Applications en Traitement de l'Image et du Signal, Université Claude Bernard Lyon 1, Hôpital Cardiologique L.Pradel Service de Radiologie
   - ERIC : Equipe de Recherche en Ingénierie des Connaissances, Université Lumière Lyon2
   - ISC : Institut des Sciences Cognitives, Université Claude Bernard Lyon1
   - LASS : Laboratoire d'Analyse des Systèmes de Santé, Université Claude Bernard Lyon1
   - TIMC : Techniques en Imagerie, Modélisation et Cognition, Faculté de Médecine de Grenoble

   Equipe hospitalière : Centre de Neuropsychologie de l'hôpital de la Pitié-Salpêtrière (Paris)

le début de ce document n'est pas consacré à un état de l'art général car chacun des chapitres le constituant est amorcé par un état de l'art spécifique.

Nous tenons également à signaler que les différentes expérimentations proposées dans cette dissertation ont été possibles grâce à l'utilisation des logiciels libres : Sipina développé au laboratoire ERIC[5], WEKA de l'Université de Waikato en Nouvelle-Zélande[6] et d'un logiciel mis au point au cours de cette thèse.

5. Sipina est disponible au téléchargement à l'adresse *http://eric.univ-lyon2.fr/%7Ericco/sipina.html*

6. WEKA est disponible au téléchargement à l'adresse *http://www.cs.waikato.ac.nz/ml/weka*

2 Concepts, Notions et Notations Utiles

"Un concept est une invention à laquelle rien ne correspond exactement mais à laquelle nombre de choses ressemblent."

- Friedrich Nietzsche -
Posthumes

Ce chapitre est l'occasion de présenter un ensemble de concepts, notions et notations qui seront utilisés tout au long de ce document. Les raisons sous-jacentes à la rédaction d'un tel chapitre sont doubles : il constituera une base de théorie nécessaire ultérieurement et son introduction préalable permettra une approche plus directe et intuitive des développements proposés plus tard. De plus, il nous semble utile et intéressant de regrouper l'ensemble de ces informations en une unique entité à laquelle on se référera facilement. Les notions introduites ici sont relatives tout d'abord au concept de données catégorielles, et enfin au Nouveau Critère de Condorcet.

2.1 Données Catégorielles

L'ensemble des terminologies et formalismes que nous utiliserons pour introduire les données catégorielles proviennent de multiples références de la littérature que nous ne manquerons pas d'évoquer. La forme de cette présentation s'inspire quant à elle de [Hua97].

Afin d'assurer une plus grande clarté nous nous appuierons sur des exemples basés sur un jeu de données décrivant un ensemble de 3 votes de motions différentes par 54 nations lors de sessions à l'O.N.U.(voir Tableau 2.1).

On définit (de manière tautologique) les données catégorielles comme les données décrivant des objets par l'intermédiaire de caractéristiques catégorielles. Les objets décrits par un ensemble de données catégorielles, sont nommés en conséquence objets catégoriels, ils correspondent à une version très simplifiée des objets symboliques définis dans [GD91]. Ces objets ne peuvent posséder de caractéristiques numériques (quantitatives), si tel est le cas on doit

Pays	M_1	M_2	M_3	Pays	M_1	M_2	M_3	Pays	M_1	M_2	M_3
DOMI	A	A	A	PANA	C	A	A	FRAN	C	B	C
POLA	A	A	C	VANE	C	A	A	SWED	C	B	C
HUNG	A	A	C	PERU	C	A	A	NORW	C	B	C
CZEC	A	A	C	CHIL	C	A	A	DENM	C	B	C
YUGO	A	A	C	ARGE	C	A	A	USA	C	C	A
BULG	A	A	C	GREE	C	A	A	UK	C	C	A
ROMA	A	A	C	CYPR	C	A	A	NETH	C	C	A
USSR	A	A	C	CANA	C	B	A	BELG	C	C	A
UKRA	A	A	C	HOND	C	B	A	LUXE	C	C	A
BYEL	A	A	C	ELS	C	B	A	URUG	C	D	A
CUBA	A	D	C	NICA	C	B	A	EQUA	C	D	B
ALBA	A	D	C	BRAZ	C	B	A	HAIT	D	A	A
FINL	B	B	C	PARA	C	B	A	GUYA	D	A	A
JAMA	C	A	A	IREL	C	B	A	BOLI	D	A	A
TRIN	C	A	A	SPAIN	C	B	A	BARB	D	A	B
MEXI	C	A	A	ITAL	C	B	A	COLU	D	B	A
GUAT	C	A	A	ICEL	C	B	A	PORT	D	C	B
COST	C	A	A	AUST	C	B	B	MALT	D	D	A

TAB. 2.1 –: *Votes à l'O.N.U. de 54 pays différents pour 3 motions différentes*

s'astreindre à une phase de discrétisation de ces caractéristiques afin d'uniformiser la description de ces objets[1].

2.1.1 Domaines et Attributs Catégoriels

Dans l'ensemble de ce document, nous considérons qu'un jeu de données est caractérisé par un ensemble de p variables notées V_1,V_2,...,V_p décrivant un espace EV ($EV = \{V_1,V_2,...,V_p\}$). Nous notons $Dom(V_1)$,$Dom(V_2)$,...,$Dom(V_p)$ les domaines respectifs des variables de EV.

EXEMPLE : $EV = \{V_1,V_2,V_3\}$ *(V_1, V_2, V_3 correspondent respectivement aux variables nommées M_1, M_2 et M_3).*

Définition 1 *Un domaine $Dom(V_j) = \{v_{j1},...,v_{jk}\}$, $(k \in N^*)$ est défini comme catégoriel s'il est fini, et non ordonné.*
Ainsi $\forall a,b \in Dom(V_j)$ les seules relations pouvant exister entre a et b sont : $a = b$ ou $a \neq b$.
V_j est en conséquence appelée variable catégorielle.
(concernant l'aspect non ordonné, il n'est pas nécessaire que l'aspect non ordonné soit réel mais on ne tiendra pas compte ultérieurement de cet ordre s'il existe)

1. Notons qu'une perte d'information plus ou moins forte est associée à ce processus

EXEMPLE : $Dom(V_1) = \{A,B,C,D\}$,$Dom(V_2) = \{A,B,C,D\}$,$Dom(V_3) = \{A,B,C\}$.

Définition 2 *$EV = \{V_1,...,V_p\}$ est un espace catégoriel si $\forall V_j$, $j \in 1..p$, V_j est une variable catégorielle.*

Notons, que les domaines catégoriels sont définis par des ensembles de singletons ainsi des valeurs provenant de combinaisons ne sont pas autorisées a contrario des travaux présentés dans [GD91].

Nous définissons un ensemble de valeurs additionelles spécifiques pour les domaines des variables catégorielles. Cet ensemble de valeurs (noté $E_\varepsilon = \{\varepsilon_i\}$) permet de représenter des cas particuliers comme ceux de la présence de valeurs manquantes (nous reviendrons ultérieurement sur les spécificités, l'intérêt et la signification de ces valeurs).

Afin de simplifier la présentation nous ne considérerons pas les relations d'inclusions conceptuelles pouvant exister au sein de bases de données provenant de la pratique contrairement aux travaux de Kodratoff et Tecuci [KT88].

2.1.2 Objets Catégoriels

Dans l'ensemble de ce document, nous considérons qu'un jeu de données est également caractérisé par un ensemble O de n objets, ces objets sont notés o_i ($O = \{o_1,...,o_n\}$). Ainsi un jeu de données est caractérisé par les ensembles $EV = \{V_1,V_2,...,V_p\}$ et $O = \{o_1,...,o_n\}$.

Cette section est consacrée à l'introduction des objets catégoriels, qui, comme dans [GD91], sont représentés par une conjonction logique de paires attributs-valeurs (Une paire attribut-valeur est dénommée sélecteur dans [MS83].). L'objet catégoriel o_i est ainsi décrit par la règle $[V_1 = o_{i_1}] \bigcap [V_2 = o_{i_2}] \bigcap ... [V_p = o_{i_p}]$.

EXEMPLE : $FRAN = [V_1 = C] \bigcap [V_2 = B] \bigcap [V_3 = C]$

En conséquence, nous représenterons chaque objet $o_i \in O$ par l'ensemble de p valeurs $\{o_{i_1},o_{i_2},...,o_{i_p}\}$ (chaque objet possède exactement p valeurs d'attributs).

EXEMPLE : $FRAN = [C,B,C]$.

REMARQUES :

- Si la valeur d'un attribut V_j est non disponible pour un objet o_i, on fixe alors $o_{i_j} = \varepsilon_k$, $\varepsilon_k \in E_\varepsilon$. Afin de simplifier ce chapitre introductif, nous considérerons en cas de valeur manquante pour un objet o_i qu'il lui est assigné la valeur ε_1 dont le comportement est identique aux autres modalités de cet attribut.
- $o_i = o_j$ si $\forall k, o_{i_k} = o_{j_k}$. Cette dernière relation n'implique toutefois pas que o_i et o_j représentent le même objet du jeu de données, mais elle signifie qu'ils possèdent les même valeurs catégorielles pour les attributs

V_1,V_2,...,V_p.
EXEMPLE : "$FRAN$" $\neq$ "$DENM$" *mais on a* $[C,B,C] = [C,B,C]$.

– Soient $C = \{o_1,o_2,...,o_n\}$ un ensemble de n objets catégoriels au sein duquel p objets sont distincts, N la cardinalité du produit cartésien $Dom(V_1) \bigotimes Dom(V_2) \bigotimes ... \bigotimes Dom(V_p)$. On a alors $p \leq N$, n peut quant à lui être inférieur, égal ou supérieur à N, ce dernier cas impliquant obligatoirement la présence de " doublons " dans C.

2.1.2.1 Similarités, Dissimilarités entre Objets Catégoriels

Nous définissons maintenant les notions de similarité et dissimilarité entre objets catégoriels. Nous venons d'indiquer qu'une notion de similarité (ou de dissimilarité) globale entre objets catégoriels existe : deux objets peuvent être similaires sans pour autant être les mêmes (cf. exemple précédent concernant les objets FRAN et DENM). Cette similarité globale implique un ensemble de similarités locales (au niveau de chaque variable), nous pouvons définir naturellement deux fonctions $\delta_{sim}(o_{a_i},o_{b_i})$ et $\delta_{dissim}(o_{a_i},o_{b_i})$ qui mesurent la similarité de deux objets o_{a_i} et o_{b_i} au niveau de la variable V_i :

$$\delta_{sim}(o_{a_i},o_{b_i}) = \begin{cases} 1 \text{ si } o_{a_i} = o_{b_i} \\ 0 \text{ si } o_{a_i} \neq o_{b_i} \end{cases} \tag{2.1}$$

$$\delta_{dissim}(o_{a_i},o_{b_i}) = \begin{cases} 0 \text{ si } o_{a_i} = o_{b_i} \\ 1 \text{ si } o_{a_i} \neq o_{b_i} \end{cases} \tag{2.2}$$

Ainsi, $\delta_{sim}(o_{a_i},o_{b_i})$ (resp. $\delta_{dissim}(o_{a_i},o_{b_i})$) vaut 1 si les objets o_{a_i} et o_{b_i} sont similaires (resp. dissimilaires) au niveau de la variable V_i.

REMARQUES :

– Les définitions de $\delta_{sim}(o_{a_i},o_{b_i})$ et $\delta_{dissim}(o_{a_i},o_{b_i})$ sont telles que $\delta_{sim}(o_{a_i},o_{b_i}) = 1 - \delta_{dissim}(o_{a_i},o_{b_i})$.
– Nous pourrions donc nous contenter de n'introduire qu'une seule de ces deux fonctions étant donnée la relation les unissant. Toutefois les développements futurs "cassant" cette relation, nous utiliserons tout au long du document ces deux fonctions afin de rendre plus intelligibles ces mêmes développements ultérieurs.

Ces mesures peuvent être étendues à l'ensemble des variables de EV de manière à rendre compte du degré de similarité globale entre les deux objets :

$$sim(o_a,o_b) = \sum_{i=1}^{p} \delta_{sim}(o_{a_i},o_{b_i}), \quad 0 \leq sim(o_a,o_b) \leq p \tag{2.3}$$

$$dissim(o_a,o_b) = \sum_{i=1}^{p} \delta_{dissim}(o_{a_i},o_{b_i}) \quad 0 \leq dissim(o_a,o_b) \leq p \tag{2.4}$$

Ainsi, plus $sim(o_a,o_b)$ (resp. $dissim(o_a,o_b)$) est proche de p plus o_a et o_b peuvent être considérés comme similaires (resp.dissimilaires).

REMARQUES :
- Les définitions de $sim(o_{a_i},o_{b_i})$ et $dissim(o_{a_i},o_{b_i})$ sont telles que $sim(o_{a_i},o_{b_i}) = p - dissim(o_{a_i},o_{b_i})$.
- Nous pourrions donc nous contenter de n'introduire qu'une seule de ces deux fonctions étant donnée la relation les unissant. Toutefois les développements futurs "cassant" cette relation, nous utiliserons tout au long du document ces deux fonctions afin de rendre plus intelligible ces mêmes développements ultérieurs.

2.1.3 Ensemble d'Objets Catégoriels

Nous introduisons maintenant un ensemble de notions relatives aux ensembles d'objets catégoriels.
Soit $C = \{o_a,o_b,...,o_h\}$ un ensemble de h objets catégoriels ($C \subseteq O$).

2.1.3.1 Mode d'un Ensemble d'Objets Catégoriels

Nous notons :
- $n_{C_{k,j}}$ le nombre d'objets de C ayant la valeur v_{jk} pour la variable $V_j \in EV$
- $f_r(V_j = v_{jk}|C) = n_{C_{k,j}}/card(C)$ la fréquence relative de la valeur v_{jk} pour V_j au sein de l'ensemble d'objets C.

Définition 3 *[Hua97] Le mode d'un ensemble d'objet C est l'objet virtuel $mode^C$ ($mode^C = [mode_j^C, j = 1..p]$) tel que pour toute variable $V_j \in EV$ la valeur d'attribut de $mode^C$ est, celle, la plus représentée pour cette variable au sein de la classe C: $\forall j = 1..p, \forall o_i \in C, f_r(V_j = mode_j^C|C) \geq f_r(V_j = o_{i_j}|C)$.*

En clair, le mode d'un ensemble d'objet C correspond au profil de cet ensemble, à l'objet type de cet ensemble

.

REMARQUES : Cette définition implique que :
- le mode d'un ensemble d'objets n'est pas forcément unique
  EXEMPLE : *le mode de l'ensemble $C = \{BARB,COLU,PORT,MALT\} = \{\{D,A,B\},\{D,B,A\},\{D,C,B\},\{D,D,A\}\}$ peut être $mode^C = [D,A,A]$, ou $mode^C = [D,B,A]$, ou bien $mode^C = [D,C,A]$, ou $mode^C = [D,D,A]$, ou $mode^C = [D,A,B]$, ou bien $mode^C = [D,B,B]$, ou encore $mode^C = [D,C,B]$, ou finalement $mode^C = [D,D,B]$.*
- le mode d'un ensemble d'objets n'est pas forcément un élément de cet ensemble, EXEMPLE : *le mode de $C = \{COLU,FRAN,US\} = \{[D,B,A],[C,B,C],[C,C,A]\}$ est $[C,B,A]$.*

2.1.3.2 Similarités et Dissimilarités entre Ensembles d'Objets Catégoriels

Tout comme nous l'avons fait pour des couples d'objets catégoriels, nous introduisons maintenant des fonctions permettant de rendre de compte du degré de similarité ou de dissimilarité de deux ensembles d'objets catégoriels.

Afin de traduire le niveau de similarité (resp. dissimilarité) entre deux ensembles d'objets catégoriels, nous utiliserons la fonction $Sim(C_i,C_j)$ (resp. $Dissim(C_i,C_j)$) qui détermine le nombre de similarités (resp. dissimilarités) entre deux ensembles d'objets différents C_i et C_j ($C_i \neq C_j$).

$$\begin{aligned} Sim(C_i,C_j) &= \sum_{o_a \in C_i, o_b \in C_j} sim(o_a,o_b) \\ 0 \leq \quad Sim(C_i,C_j) &\leq card(C_i) \times card(C_j) \times p \end{aligned} \tag{2.5}$$

$$\begin{aligned} Dissim(C_i,C_j) &= \sum_{o_a \in C_i, o_b \in C_j} dissim(o_a,o_b) \\ 0 \leq \quad Dissim(C_i,C_j) &\leq card(C_i) \times card(C_j) \times p \end{aligned} \tag{2.6}$$

Ainsi, plus $Sim(C_i,C_j)$ (resp. $Dissim(C_i,C_j)$) est proche de $card(C_i) \times card(C_j) \times p$ plus C_i et C_j peuvent être considérés comme similaires (resp. dissimilaires).

2.1.3.3 Similarités et Dissimilarités au sein d'un Ensemble d'Objets Catégoriels

Les notions de similarités (resp. dissimilarités) au sein d'un ensemble d'objets catégoriels correspondent au nombre de similarités (resp. dissimilarités) entre objets d'un ensemble C_i et constituent une extension des définitions préalablement établies pour les similarités et dissimilarités entre ensembles d'objets catégoriels.

$$\begin{aligned} Sim(C_i) &= \sum_{o_a \in C_i, o_b \in C_i, a<b} sim(o_a,o_b), \\ 0 \leq \quad Sim(C_i) &\leq \frac{card(C_i) \times (card(C_i)-1) \times p}{2} \end{aligned} \tag{2.7}$$

$$\begin{aligned} Dissim(C_i) &= \sum_{o_a \in C_i, o_b \in C_i, a<b} dissim(o_a,o_b) \\ 0 \leq \quad Dissim(C_i) &\leq \frac{card(C_i) \times (card(C_i)-1) \times p}{2} \end{aligned} \tag{2.8}$$

Ainsi, plus $Sim(C_i)$ (resp. $Dissim(C_i)$) est proche de $\frac{card(C_i) \times (card(C_i)-1) \times p}{2}$ plus C_i présente une forte homogénéité (resp. hétérogénéité) interne.

2.1.3.4 Voisinage d'une Partition d'un Ensemble d'Objets Catégoriels

Nous notons : $P_h = \{C_1,...,C_z\}$ une partition de O en z classes et P_g une partition de O en y classes.

Définition 4 *Nous dirons qu'une partition P_g appartient à $Vois(P_h)$ le voisinage d'une partition P_h ou encore l'ensemble des partitions voisines d'une partition P_h si :*

- *P_g peut être obtenue à partir de P_h par segmentation d'une classe C_j de P_h selon une variable V_i (processus équivalent à la segmentation des arbres de décision)*
- *P_g peut être obtenue à partir de P_h par fusion de deux classes de P_h*
- *$P_g = P_h$*

EXEMPLE : *Soient $O = \{BARB,COLU,EQUA,MALT\}$;*
$P_1 = \{\{COLU,MALT\},\{BARB,EQUA\}\}$.
Le voisinage de P_1 est alors constitué des partitions de O provenant de :

- *la fusion de deux classes de P_1, il s'agit uniquement ici de la partition $P_2 = \{BARB,COLU,EQUA,MALT\}$;*
- *la segmentation d'une classe de P_1 selon une des variables catégorielles, il s'agit ici des partitions :*
  $P_3 = \{\{COLU\},\{MALT\},\{BARB,EQUA\}\}$ (segmentation de la classe $\{COLU,MALT\}$ selon M_2),
  $P_4 = \{\{COLU,MALT\},\{BARB\},\{EQUA\}\}$ (segmentation de la classe $\{BARB,EQUA\}$ selon M_1 ou M_2) ;
- *P_1 elle même.*

Ainsi $vois(P_1) = \{P_1,P_2,P_3,P_4\}$.

2.2 Le Nouveau Critère de Condorcet

Nous présentons maintenant une mesure particulière pour l'évaluation de la qualité d'une partition : le Nouveau Critère de Condorcet[2] (NCC) introduit par Pierre Michaud. Le lecteur désirant une description de ce critère plus complète que celle que nous proposons peut se référer aux travaux de Michaud suivants [Mic82], [Mic83], [Mic85], [Mic87], [Mic97].

Dans le cadre de ces travaux, Michaud a proposé une nouvelle mesure de l'aspect naturel d'une partition : le NCC. Cette mesure est dérivée de la théorie de l'agrégation de votes, ce qui apparaît plus clairement si l'on considère la correspondance entre :

- une variable catégorielle définissant une partition sur un ensemble d'objets, et
- un juge donnant son opinion sur la ressemblance/similarité d'objets.

2. Le nom de ce critère est motivé par l'existence d'un critère similaire dans le cadre de l'agrégation de rangs, critère qui fut proposé par Condorcet en 1785

EXEMPLE : *Considérons, par exemple, l'ensemble d'objets O ainsi que la variable* M_1 *(*$O = \{DOMI,COLU,EQUA,MALT\}$*). Cette variable définit la partition suivante de* O *:* $\{\{DOMI\},\{MALT\},\{COLU,EQUA\}\}$*. On peut ainsi l'associer naturellement à un juge pour qui l'ensemble d'objets s'organiserait selon 3 groupes d'objets similaires.*

Les variables dans le cadre de la classification non supervisé (cns[3]), ou les juges dans l'optique agrégation de votes, caractérisant les objets sont multiples. La recherche de la bonne classification s'apparente alors à un problème d'agrégation d'opinions des juges. Définir une méthode idéale pour l'agrégation des opinions peut être impossible à réaliser, cependant divers auteurs, tel K.J. Arrow [Arr63], ont axiomatisé les propriétés désirables pour une bonne agrégation. Le NCC vérifie certaines de ces propriétés que nous ne détaillons pas ici.

La mesure de l'aspect naturel d'une partition P_h (Nous notons $P_h = \{C_1,...,C_z\}$ une partition de O en z classes) $NCC(P_h)$ est définie comme suit :

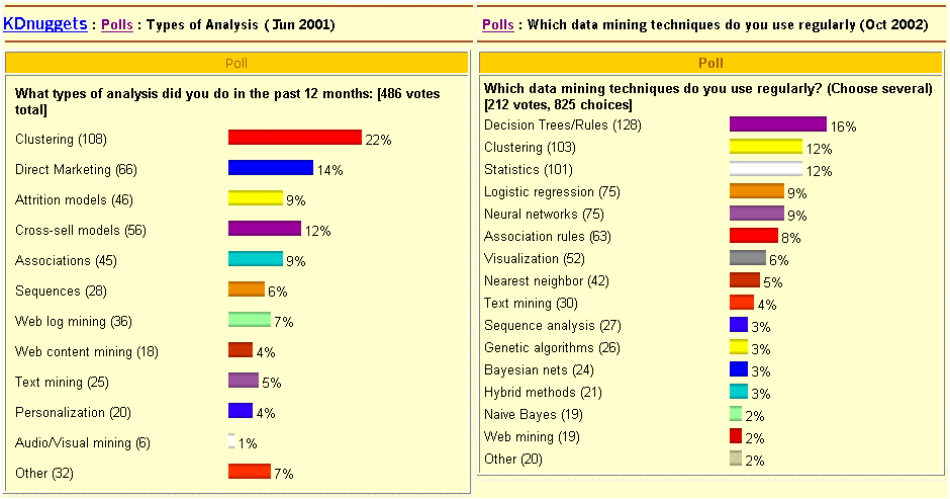
$$NCC(P_h) = \sum_{i=1..z,j=1..z,i<j} Sim(C_i,C_j) + \sum_{i=1}^{z} Dissim(C_i) \qquad (2.9)$$

$0 \leq NCC(P_h) \leq \sum_{i=1..z,j=1..z,i<j} card(C_i) \times card(C_j) \times p + \sum_{i=1}^{z} \frac{card(C_i) \times (card(C_i)-1) \times p}{2}$

Le critère $NCC(P_h)$ comptabilise donc le nombre de dissimilarités internes à chacune des classes de la partition P_h ainsi que le nombre de similarités entre classes de P_h. Ainsi, plus il est faible (proche de 0) plus cela signifie que la partition présente un faible nombre de dissimilarités internes pour chacune des classes et un faible nombre de similarités entre classes ; cela signifie donc que plus la valeur du critère NCC est proche de 0 plus la partition apparaît comme naturelle et semblant traduire la structure interne des données.

3. Par la suite, l'acronyme cns remplacera l'expression classification non supervisée d'une part pour évoquer le processus de classification non supervisée et d'autre part pour évoquer le résultat de ce processus (i.e. une partition de l'ensemble des objets). Notons que cet acronyme est invariablement utilisé pour rendre compte d'un ou plusieurs processus ou de un ou plusieurs résultats de processus...

3 Classification Non Supervisée



Résultats de Sondages du Site Web KDNuggets (juin 2001 et octobre 2002) (KDNuggets Polls : www.kdnuggets.com/polls/)

3.1 Introduction

La classification non supervisée (cns) constitue l'un des outils les plus populaires de la fouille de données. Son utilisation permet la découverte de groupes, de motifs, d'éléments structurels à l'intérieur d'un jeu de données. La problématique sous-jacente à la cns consiste donc en le partitionnement de l'ensemble des objets du jeu de données considéré en des groupes tels que les objets d'un même groupe sont relativement similaires entre eux alors que ces objets sont relativement différents des objets des autres groupes [GRS98] [JMF99]. Ainsi, l'idée est de mettre à jour la structure interne des données en révélant l'organisation des objets par l'intermédiaire de groupes. L'analyse de ces groupes permettra alors de dériver un ensemble de connaissances sur le jeu de données.

La cns est utilisée dans divers domaines tels la biologie, la médecine, l'ingénierie... D'ailleurs, à l'appellation cns peut être substitué un ensemble d'autres

termes selon le contexte d'utilisation : apprentissage non supervisé (en reconnaissance de formes), taxonomie (en sciences de la vie), typologie (en sciences humaines)...

Nous proposons ici une rapide et bien évidemment non exhaustive présentation du processus de classification non supervisée. Le domaine est si vaste qu'envisager une présentation intégrale du domaine semble non réaliste, toutefois nous recommandons la lecture des travaux de [JMF99] [Ber02] [JD88] [HBV01] qui permettent une approche relativement complète du domaine.

3.1.1 Méthodologie Générale de la Classification Non Supervisée

La méthodologie générale de la cns est largement semblable à celle de l'ECD : elle est constituée d'un ensemble de tâches s'enchaînant séquentiellement et pouvant être intégrées au sein d'un processus itératif. Ces étapes sont les suivantes [FPSSU96] (voir figure 3.1) :

1. **Sélection de Variables** (SdV) : l'objectif est essentiellement ici de procéder à la sélection des variables encodant le mieux l'information que l'on veut soumettre à l'analyse. Plus rarement, la SdV pour la cns à un objectif identique à celui de la SdV dans le cadre de l'apprentissage supervisé : l'élimination de l'information non pertinente. Notons que dans ce dernier cas la nature non supervisée et l'utilisation à but exploratoire de la cns confère à la SdV un aspect paradoxal : "Dans quelle mesure puis-je conjecturer que cette information n'est pas pertinente alors que je n'ai aucune connaissance, idée, sur ce que je désire découvrir?". Nous proposons toutefois au chapitre 5 une méthode apparemment efficiente pour la SdV dans ce cadre particulier.
2. **Application de l'algorithme de cns** : cette étape centrale concerne le choix de la méthode de cns à utiliser, sa paramétrisation, le choix des mesures de similarités/dissimilarités à employer.
3. **Validation des Résultats** : de par la nature non supervisée de la cns, l'évaluation de la validité des structures mises à jour est essentielle mais relativement ardue. Nous abordons en détail ce point au chapitre 4.
4. **Interprétation des Résultats** : Dans de nombreux cas des experts doivent intégrer les résultats de la cns dans un ensemble plus vaste d'analyses afin d'aboutir à des conclusions intéressantes et valides.

3.1.2 Applications de la Classification Non Supervisée

Comme le montre les sondages réalisés sur le site KDNUGGETS[1] la cns est un outil majeur en ECD (voir images page 15). Nous résumons ici les applications de la cns en nous référant à [HBV01] :

– *Réalisation de Taxonomies* : il s'agit évidemment de l'application principale de la cns, les taxonomies générées pouvant alors être utilisées telles

1. www.kdnuggets.com

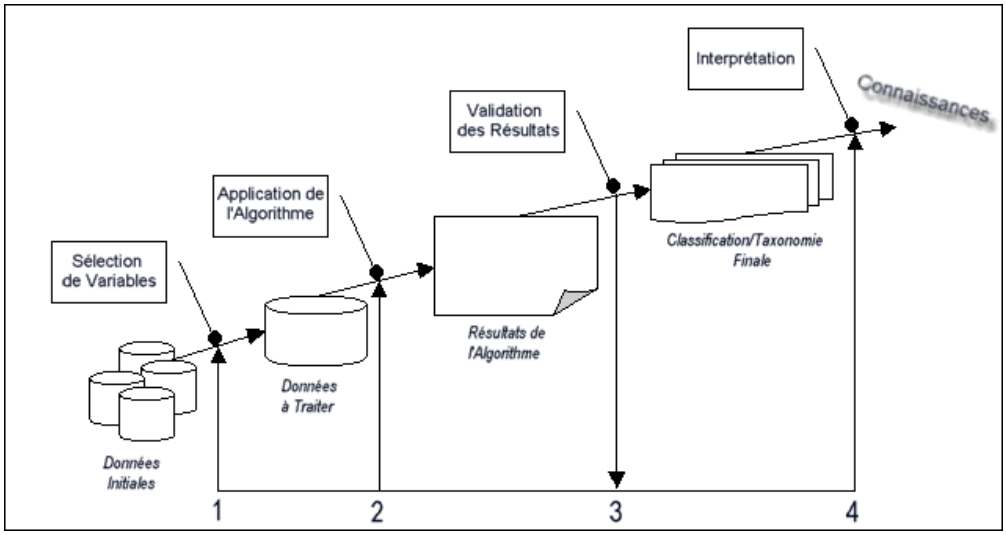


FIG. 3.1 –: *Etapes du Processus de Classification Non Supervisée*

qu'elles ou au sein d'analyses plus élaborées (la cns peut, par exemple, constituer une étape de pré-traitement de l'information préalable à l'utilisation d'autres algorithmes comme ceux d'apprentissage supervisé).

– *Résumer l'information* : la cns peut contribuer à la compression de l'information comprise dans un jeu de données, et permettre une présentation synthétique des éléments informationnels les plus marquants d'un jeu de données.

– *Génération/Test d'hypothèses* : La cns peut être utilisée afin d'inférer des hypothèses sur l'organisation, les relations existant parmi les objets d'un jeu de données ; de manière opposée, la cns peut être utilisée dans le but de confirmer/rejeter un ensemble d'hypothèses préalablement émises au sujet des objets d'un jeu de données.

3.1.3 Taxonomies des Méthodes de Classification Non Supervisée

Une multitude de méthodes de cns sont proposées dans la littérature, on classifie traditionnellement ces méthodes selon :

– Le type de données qu'elles peuvent traiter

– La mesure employée pour rendre compte de la similarité / dissimilarité entre objets

– Les éléments théoriques et les concepts fondamentaux sous-jacents à la méthode.

Les méthodes de cns sont ainsi généralement classées selon les groupes suivants [JMF99] :

- les *méthodes "partitionnelles"* déterminent une partition des objets du jeu de données ; pour cela elles procèdent à une recherche itérative de la partition optimisant un critère particulier.
  Méthode(s) Représentative(s) : K-Means [Mac67], Fuzzy C-Means [BEF84], K-Modes [Hua97], PAM [NH94], CLARA [NH94], CLARANS [NH94]....
- les *méthodes hiérarchiques* procèdent, quant à elles, par fusion ou segmentation de classes d'une partition en initialisant la partition initiale soit à la partition grossière (une unique classe) soit à la partition la plus fine (chaque objet constitue alors une classe). Dans le premier cas, la méthode procède par segmentation de classes, on parle alors de méthode divisive ou encore descendante ; dans le second, cas la méthode procède par fusion de classes et l'on parle cette fois de méthode agglomérative ou ascendante. Le résultat est alors un arbre appelé dendrogramme dont chaque niveau correspond à une partition particulière du jeu de données.
  Méthode(s) Représentative(s) : BIRCH [ZRL96], CURE [GRS98], ROCK [GRS00], Méthode de Williams et Lambert [WL59], Méthode de Fages....
- les *méthodes basées sur la densité* pour lesquelles l'idée clé est de procéder à des regroupements d'objets en groupes selon le voisinage des objets et son niveau de densité.
  Méthode(s) Représentative(s) : DBSCAN [EKSX96] [EKS+98], DENCLUE [EKS+98]...
- les *méthodes à base de grilles* sont essentiellement utilisées pour des données spatiales, elles procèdent par discrétisation de l'espace en un certains nombre de cellules et effectuent ensuite l'ensemble des traitements sur l'espace discrétisé.
  Méthode(s) Représentative(s) : Wave-Cluster [SCZ98], STING [HK01] [WYM97] [WYM99].

D'autres auteurs [Hua97] [GRS00] [RLR98a] proposent également une classification selon le type de données traitées par la méthode :

- les *méthodes dites statistiques* basées sur des concepts statistiques et traitant les données quantitatives qui utilisent des mesures de similarité pour partitionner les objets,
- les *méthodes conceptuelles* utilisées pour les données catégorielles classifient quant à elles les objets selon le concept qu'ils véhiculent.

On peut également classer les méthodes de cns selon leur façon d'intégrer l'incertitude :

- les *méthodes floues* utilisent la théorie des ensembles flous afin de mettre à jour, non pas une partition de l'ensemble des objets, mais un recouvrement de cet ensemble. Ainsi, un objet peut appartenir à plusieurs classes

et un degré d'appartenance à chacune des classes auxquelles il peut appartenir lui est asssigné. Ce type de méthodes peut donc permettre de transcrire les cas pratiques où l'incertitude doit être intégrée.
Méthode(s) Représentative(s) : Fuzzy C-Means [BEF84]...

– à l'opposé les méthodes classiques, les plus nombreuses, considèrent des partitions de l'ensemble des objets sans aucun recouvrement.

D'autres éléments permettent d'effectuer des distinctions entre méthodes, ainsi on peut opposer les méthodes par :

– leurs approches d'optimisation qui peuvent être *déterministes* ou *stochastiques*,
– leur *incrémentalité* ou *non-incrémentalité*,
– leur aspect *monothétique* ou *polythétique*
– l'utilisation de paradigmes particuliers, on peut citer :
  – les *méthodes neuronales* comme les cartes de Kohonen [Koh89] qui présentent de manière graphique des partitions de l'ensemble des objets ce qui permet d'obtenir des connaissances sur la proximité entre classes de la partition,
  – *les méthodes évolutionnaires* utilisant les algorithmes génétiques (telle la méthode proposée par Cristofor et Simovici [CS02]) ou la programmation génétique...
– ...

3.1.4 Méthodes de Classification Non Supervisée pour Données Catégorielles

Les travaux présentés ultérieurement concernant la cns pour données catégorielles, nous donnons maintenant les informations majeures (type de méthode, complexité algorithmique, géométrie des groupes extraits, faculté à gérer les outliers, paramètres, forme des résultats, critère sous-jacent à la méthode, publication de référence) concernant quelques une des méthodes de cns pour données catégorielles les plus récentes. Nous abordons ainsi successivement les méthodes **K-Modes**, **ROCK**, **RDA/AREVOMS**, une méthode proposée par **Cristofor et Simovici**, et enfin **STIRR**.

La plus connue et la plus populaire des méthodes de cns est certainement la méthode des K-Means [Mac67] (qui possède plusieurs évolutions comme les méthodes de nuées dynamiques). Cette méthode vise à déterminer la partition de l'ensemble des objets du jeu de données considéré telle qu'elle optimise (minimise) la fonction objectif $E = \sum_{i=1..nc} \sum_{x \in C_i} d(x,m_i)$. Ici, m_i est le centre de la classe C_i, nc est le nombre de classes de la partition, et $d(x,m_i)$ est le carré de la distance euclidienne entre l'objet x et le centre de la classe à laquelle il appartient. L'objectif de cet algorithme est donc de déterminer la partition (possédant un nombre de classes fixé a priori) telle que l'homogénéité des classes

soit la plus forte possible. Plus spécifiquement, l'algorithme débute par l'initialisation du centre des nc classes, puis assigne séquentiellement chaque objet à la classe dont le centre est le plus proche, ce dernier processus étant réitéré tant que des mouvements d'objets d'une classe à une autre ont lieu.

L'algorithme des **K-Modes** [Hua97] constitue l'adaptation des K-Means au cadre des données catégorielles. Pour ce faire, la notion de mode d'une classe (voir chapitre 2) est substituée à celle de centre d'une classe. Le critère à optimiser (à minimiser plus précisément) est donc le suivant :
$QKM = \sum_{i=1..nc} \sum_{x \in C_i} d(x,mode^{C_i})$, $d(x,mode^{C_i}) = dissim(x,mode^{C_i})$.

Ses principales caractéristiques sont :

- **Complexité :** $O(n)$,
- **Géométrie des Classes :** Forme non-concave,
- **Gestion des Outliers :** Non,
- **Paramètres :** Nombre de Classes,
- **Résultats :** Composition et Mode de chaque classe,
- **Critère :** Minimisation du critère QKM

La méthode **ROCK** [GRS00] est, elle, à classer dans la catégorie des méthodes hiérarchiques. La particularité principale de cette méthode est d'employer deux nouveaux concepts :

- Le concept de voisins d'un objet, que l'on peut décrire grossièrement comme les objets significativement similaires à cet objet. En fait, une mesure de similarité/proximité (qui peut être ou non une métrique) est utilisée afin de rendre compte de la proximité d'un couple d'objets, et si la valeur de cette mesure remplit certaines conditions alors les objets seront considérés comme voisins.
- Le concept de liens entre deux objets est utilisé pour mesurer la similarité/proximité entre deux objets. Le nombre de liens entre deux objets est défini comme le nombre de voisins en commun que possèdent ces deux objets.

Ainsi, l'algorithme hiérarchique agglomératif ROCK emploie ces concepts de voisinage et de liens (et non le concept de distance) afin de procéder aux fusions de classes. Notons que des étapes d'échantillonnage des objets peuvent être inclues dans le processus afin d'en réduire le temps d'exécution.

Ses principales caractéristiques sont :

- **Complexité :** $O(n^2 + nm_m m_a + n^2 logn)$ avec m_m le nombre maximal de voisins pour un objet, m_a le nombre moyen de voisins pour un objet,
- **Géométrie des Classes :** Pas d'a priori particulier pour la forme des classes,
- **Gestion des Outliers :** Oui
- **Paramètres :** Nombre de classes et θ une valeur de seuil pour la définition des voisins,
- **Résultats :** Composition de chaque classe,

– **Critère :** Maximisation du critère g ($g = sum_{i=1..nc} ni \times \sum_{o_a,o_b \in C_i} \frac{link(o_a,o_b)}{n_i^{1+2f(\theta)}}$, avec n_i le nombre d'objets de la classe C_i, $link(o_a,o_b)$ le nombre de liens entre o_a et o_b, $n_i^{1+2f(\theta)}$ une fonction visant à "estimer le nombre de liens que doit comprendre C_i").

Michaud décrit dans [Mic97] une méthode dénommée **RDA/AREVOMS** (Relational Data Analysis/Analyse du REsultat d'un VOte Majoritaire et S-théorie). L'originalité principale de cette méthode est l'utilisation d'un critère particulier : le Nouveau Critère de Condorcet (NCC) (voir chapitre 2). Ce critère issu de l'analyse des préférences permet une détermination automatique du nombre de classes de la partition résultant du processus de cns. Initialement, le problème d'optimisation sous-jacent à cette méthode était résolu par la programmation en nombres entiers ce qui conférait à cette méthode une non applicabilité sur des jeux de données de taille importante. Une première évolution de la méthode a alors consisté à substituer à la méthode d'optimisation par programmation en nombres entiers une méthode d'optimisation par programmation linéaire [MM81]. Le coût calculatoire est dans ce cas réduit mais l'optimalité du résultat n'est plus assurée. D'autres tentatives pour la mise au point d'une méthode plus rapide ont ensuite débouché sur la mise au point d'une méthode utilisant une heuristique de complexité en $O(n^2)$ [Mic91]. Finalement, une nouvelle heuristique de complexité en $O(n)$ a été développée par Michaud en s'appuyant sur une transformation du problème par le biais de sa S-théorie. Notons la remarquable accélération du processus due à cette méthode, ainsi que le fait que les solutions apportées par cette méthode sont très proches de l'optimalité et qu'il est également possible de fixer, si on le désire, le nombre de classes que doit posséder la partition résultat.

Ses principales caractéristiques sont :

– **Complexité :** $O(n)$,
– **Géométrie des Classes :** Pas d'a priori particulier pour la forme des classes,
– **Gestion des Outliers :** oui
– **Paramètres :** Aucun ou nombre de classes,
– **Résultats :** Composition de chaque classe,
– **Critère :** Minimiser le Nouveau Critère de Condorcet.

Cristofor et Simovici présentent quant à eux dans [CS02] une méthode visant à surpasser le problème difficile de la définition d'une mesure de similarité naturelle entre objets catégoriels. Pour cela, ils utilisent une mesure de dissimilarités entre partitions d'objets catégoriels (ces partitions étant en définitive les partitions déterminées par les variables catégorielles du jeu de données considéré). Cette mesure est basée sur la théorie de l'information et plus particulièrement sur l'entropie généralisée. Le processus de cns correspond

alors à un problème d'optimisation résolu par l'intermédiaire d'un algorithme génétique. Ses principales caractéristiques sont :

- **Complexité :** $O(gpn)$ avec g le nombre de générations de l'algorithme génétique, p le nombre de chromosomes d'une génération de l'algorithme génétique,
- **Géométrie des Classes :** Pas d'a priori particulier pour la forme des classes,
- **Gestion des Outliers :** Oui
- **Paramètres :** paramètres classiques pour un algorithme génétique, nombre maximal de générations sans amélioration de la fonction objectif,
- **Résultats :** Composition de chaque classe, poids décrivant l'influence de chacune des variables catégorielles dans la constitution du la cns,
- **Critère :** voir l'article de référence [CS02].

La méthode **STIRR** [GKR00] proposée par Gibson, Kleinberg et Raghavan est, elle, basée sur une approche itérative visant à associer des poids aux variables catégorielles d'un jeu de données afin de faciliter l'utilisation d'un certain type de mesures de similarité basées sur la cooccurrence de valeurs. Cette méthode peut ainsi être vue comme utilisant un type particulier de système dynamique non linéaire et généralisant les méthodes de partitionnement spectral de graphes. Notons que les résultats nécessitent ici une interprétation relativement ardue.

- **Complexité :** non évoquée par les auteurs, les évaluations expérimentales qu'ils proposent semblent montrer une complexité en $O(n)$,
- **Géométrie des Classes :** Pas d'a priori particulier pour la forme des classes[2],
- **Gestion des Outliers :** Oui[3]
- **Paramètres :** nombre d'itérations du processus itératif, pour le reste voir l'article de référence [GKR00]
- **Résultats :** Un graphique à interpréter,
- **Critère :** voir l'article de référence [GKR00].

3.1.5 Challenges Actuels en Classification Non Supervisée

De nombreux algorithmes classiques de cns ne permettent pas un traitement réellement satisfaisant des données pour de multiples raisons. Ces raisons peuvent être classées selon qu'elles sont inhérentes aux données traitées ou liées à des contraintes dues au domaine dans lequel est utilisé l'algorithme de cns :

- **Problèmes inhérents aux données traitées :**
  - **Très grand nombre d'objets.** Si le nombre d'objets à traiter est très élevé, les algorithmes utilisés se doivent de posséder une complexité

2. dans la mesure où les classes sont déterminées par interprétation d'un graphique

algorithmique théorique relativement faible, ou plutôt, exhiber une bonne scalabilité. En effet, comme de nombreux problèmes intéressants, la cns mène à la résolution de problèmes généralement NP-difficiles. Il est généralement admis que des algorithmes "valables" en terme de coût calculatoire doivent posséder une complexité algorithmique linéaire ou log-linéaire, dans quelques cas particuliers une complexité quadratique ou cubique peut être acceptée.

- **Dimensionnalité élevée.** Dans certains cas, le nombre de descripteurs (variables) est très élevé (parfois supérieur au nombre d'objets). Ainsi, un algorithme de cns doit pouvoir affronter "le fléau de la dimensionnalité"...[3]
- **Autres éléments problématiques.** Le problème de la *"vacuité"* ("sparsity") des données peut affecter la complexité algorithmique ainsi que le choix de la mesure de similarité à utiliser, la présence de *données manquantes* soulève également des questions pour le choix de cette mesure. La présence d'*outliers* et leur détection est un problème non trivial, ainsi il est parfois nécessaire de posséder un algorithme relativement insensible à leur présence. (Dans le cas de données catégorielles, la notion d'outliers fait référence aux objets présentant une description particulièrement différentes de l'ensemble des autres objets du jeu de données.)

- **Problèmes inhérents à des contraintes applicatives :**
  - **Necessité d'intégrer des connaissances, des contraintes dans le processus.** Le processus d'extraction des connaissances dans lequel s'insère le processus de cns est essentiellement anthropocentré et itératif. Donner la possibilité à l'utilisateur de "jouer" avec le processus de cns en y intégrant ses connaissances ou d'éventuelles contraintes est donc souvent nécessaire.
  - **Bonne utilisabilité.** Souvent l'utilisateur final d'un algorithme de cns ne s'avère pas un expert du domaine : simplicité d'utilisation (paramétrage facile et intelligible de l'algorithme), présentation explicite et intelligible des résultats et connaissances extraites par le processus de cns sont alors des éléments indispensables pour une utilisation profitable et pertinente d'un algorithme.
  - **Données distribuées.** Les grands entrepôts de données proposent le plus souvent des sources de données distribuées. Les modèles de cns que l'on peut bâtir localement nécessitent parfois d'être intégrés dans un modèle global/holistique.

Cette liste résume partiellement les problèmes en cns constituant les principales préoccupations actuelles des chercheurs dans ce domaine. Nous évalue-

3. curse of dimensionality [Fri94]

rons, en conclusion, dans quelle mesure la méthode de cns que nous proposons apporte ou non des solutions à chacun d'entre eux.

3.2 Une Nouvelle Méthode de Classification Non Supervisée "Orientée Utilisateur"

Nous proposons ici une méthode de classification non supervisée (pour données catégorielles) basée sur le concept de la comparaison par paires et plus spécifiquement sur le Nouveau Critère de Condorcet (*NCC*).

Les idées mises en œuvre pour la mise au point de cette méthode ne sont pas totalement nouvelles puisqu'il s'agit d'une généralisation (non hiérarchique) de la méthode d'analyse des associations de Williams et Lambert [WL59]. De plus, l'idée d'utiliser le concept de la comparaison par paires pour élargir la problématique de Williams et Lambert aux cas où les variables nominales ont plus de deux modalités est également évoquée dans [Mar84a] [Mar84b]. Enfin, l'utilisation du *NCC* au sein d'un algorithme de cns a été proposée par P.Michaud [Mic97] [Mic91] [Mic87] .

Nous pensons cependant que les travaux proposés ici présentent plusieurs intérêts :

- revivifier l'intérêt de la communauté ECD pour des travaux en agrégation des préférences et comparaison par paires à l'intérêt indéniable (les travaux de Michaud et Marcotorchino sont selon nous des sources de réflexions et d'inspirations immenses),
- proposer une méthode de cns possédant de multiples avantages pour l'utilisateur,
- proposer une méthode permettant de dériver une méthode d'apprentissage semi-supervisé...

Nous utilisons pour notre méthode de cns un critère dérivé du Nouveau Critère de Condorcet (NCC), la présentation de ce critère constitue le point de départ de la présentation de la méthode proposée, puis suivent la présentation de l'algorithme et son évaluation expérimentale. Nous concluons par l'introduction de plusieurs éléments additionnels conférant notamment à la méthode la possibilité d'être "transformée" en une méthode d'apprentissage semi-supervisée dont la présentation et l'évaluation achève ce chapitre.

3.2.1 Critère d'Évaluation de l'Aspect Naturel d'une Partition d'Objets

Le problème sous-jacent à la cns est : "étant donné un ensemble d'objets O, déterminer une partition P_{nat} de O que l'on dénommera naturelle et qui reflète la structure interne des données". Cette partition doit être telle que ses classes sont constituées d'objets présentant une relative forte similarité et que les objets de classes différentes présentent une relative forte dissimilarité. Pour

ce faire, on doit disposer d'un critère permettant de capturer l'aspect naturel d'une partition, nous utilisons ici $NCC^\star$ une version modifiée du Nouveau Critère de Condorcet. Le principal avantage de ce critère est, tout comme pour le critère NCC, une détermination automatique du nombre final de classes pour la cns. De plus, l'introduction d'un nouvel élément dit "facteur de granularité" permet également à l'utilisateur d'influer, de "piloter" la résolution de la partition résultant du processus de cns (i.e. d'influer sur le nombre de classes de cette partition). Voici la définition formelle de $NCC^\star(P_h)$ la mesure de l'aspect naturel d'une partition $P_h = \{C_1,...,C_z\}$:

$$NCC^\star(P_h) = \sum_{i=1..z,j=1..z,j>i} Sim(C_i,C_j) + \alpha \times \sum_{i=1}^{z} Dissim(C_i) \tag{3.1}$$

α scalaire appelé facteur de granularité ($\alpha \geq 0$), si $\alpha = 1$ alors $NCC^\star(P_h) = NCC(P_h)$

Nous rappelons ici quelques définitions données au chapitre précédent :

$$\begin{aligned} Sim(C_i,C_j) &= \sum_{o_a \in C_i, o_b \in C_j} sim(o_a,o_b) \\ Dissim(C_i) &= \sum_{o_a \in C_i, o_b \in C_i} dissim(o_a,o_b) \\ sim(o_a,o_b) &= \sum_{i=1}^{p} \delta_{sim}(o_{a_i},o_{b_i}) \\ dissim(o_a,o_b) &= \sum_{i=1}^{p} \delta_{dissim}(o_{a_i},o_{b_i}) \end{aligned} \tag{3.2}$$

$$\delta_{sim}(o_{a_i},o_{b_i}) = 1 - \delta_{dissim}(o_{a_i},o_{b_i}) = \begin{cases} 1 \text{ si } o_{a_i} = o_{b_i} \\ 0 \text{ si } o_{a_i} \neq o_{b_i} \end{cases} \tag{3.3}$$

Ainsi, $NCC^\star(P_h)$ mesure simultanément les dissimilarités entre objets de même classe de la partition P_h, et les similarités entre objets de classes différentes (on peut donc dire que $NCC^\star(P_h)$ est défini comme une fonction de l'homogénéité interne des classes et de l'hétérogénéité entre classes). Donc, les partitions présentant une forte homogénéité intra-classe et une forte disparité inter-classes posséderont une faible valeur pour $NCC^\star$ et constitueront les partitions apparaissant comme les plus naturelles.

Définition 5 *Une partition P_1 est dite plus naturelle qu'une partition P_2 (ou encore représentant mieux la structure interne des données) si $NCC^\star(P_1) < NCC^\star(P_2)$.*

Définition 6 *Une partition d'un ensemble d'objets O est nommée Partition Naturelle de O et est notée P_{nat} si elle minimise $NCC^\star$:*
$\wp$ ensemble des partitions de O, $\forall P_i \in \wp, NCC^\star(P_{nat}) \leq NCC^\star(P_i)$.

REMARQUES :

- Il peut y avoir plusieurs partitions naturelles différentes pour un ensemble d'objets; par exemple si $\alpha = 1$,
  $O = \{ICEL,FRAN,SWED,NORW,DENM,USA,UK,NETH,BELG\}$,
  $P_1 = \{\{ICEL,FRAN,SWED,NORW,DENM\},\{USA,UK,NETH,BELG\}\}$,
  $P_2 = \{\{FRAN,SWED,NORW,DENM\},\{ICEL,USA,UK,NETH,BELG\}\}$,
  $NCC^\star(P_1) = NCC^\star(P_2) = 28$, et $\wp$ ensemble des partitions de O,
  $\forall P_i \in \wp, 28 \leq NCC^\star(P_i)$
- Notons le rôle du facteur de granularité α (non présent dans la définition initiale du NCC) : celui ci permet soit de privilégier l'influence de l'homogénéité intra-classe ou de la disparité inter-classes pour la détermination de l'aspect naturel d'une partition. En effet, plus α est élevé (resp. faible) plus une partition doit présenter une forte homogénéité intra-classe (resp. une forte disparité inter-classes) pour apparaître naturelle. Par exemple, si
  $O = \{ICEL,FRAN,SWED,NORW,DENM,USA,UK,NETH,BELG\}$,
  $P_1 = \{\{ICEL,FRAN,SWED,NORW,DENM\},\{USA,UK,NETH,BELG\}\}$,
  $P_3 = \{\{FRAN,SWED,NORW,DENM\},\{ICEL\},\{USA,UK,NETH,BELG\}\}$,
  $NCC^\star(P_1) = 24 + \alpha \times 4$, $NCC^\star(P_3) = 32 + \alpha \times 0$, ainsi
  pour $\alpha = 1$ $NCC^\star(P_1) < NCC^\star(P_3)$,
  pour $\alpha = 2$ $NCC^\star(P_1) = NCC^\star(P_3)$,
  pour $\alpha = 3$ $NCC^\star(P_1) > NCC^\star(P_3)$

3.2.2 La Méthode de Classification Non Supervisée "Orientée Utilisateur"

3.2.2.1 Travaux Liés et Spécificités du Travail

La cns pour données catégorielles a été l'objet de multiples travaux ces dernières années, les principaux challenges associés à ces recherches sont :

- **la définition de critères bien adaptés à ce cadre particulier** :
  - dans ROCK [GRS00] les auteurs utilisent une mesure basée sur le nombre de voisins communs que possèdent deux objets,
  - Cristofor et Simovici [CS02] utilisent quant à eux une mesure basée sur l'entropie généralisée,
  - Michaud [Mic97] utilise quant à lui le NCC ;
- **la mise au point d'algorithmes au coût calculatoire relativement faible** :
  - Huang [Hua97] propose les K-Modes une adaption de la méthode des K-Means dans le cas de données catégorielles,
  - Gibson et al. [GKR00] utilisent les systèmes dynamiques pour STIRR,
  - Michaud s'appuie sur la S-théorie [Mic87] [Mic91] pour dériver un algorithme efficace [Mic97].

Notre propos n'est pas ici de poursuivre dans ces directions de recherche mais plutôt de s'appuyer sur ces travaux afin de proposer une méthode efficace exhibant de nombreux avantages pour l'utilisateur et utilisable par un non spécialiste. En effet, l'aspect utilisabilité est trop peu abordé dans la littérature. On peut ainsi lister un ensemble de qualités pratiques désirables par un utilisateur et qui distinguerait largement une méthode les possédant des approches existantes :

- générer une description des classes définies aisément compréhensible et ne nécessitant aucun post-traitement contrairement à [GRS00], [CS02], [GKR00] [Mic97]
- coût calculatoire relativement faible contrairement à [GRS00]
- ne pas nécessiter de fixer a priori le nombre final de classes contrairement à [Hua97] mais toutefois permettre à l'utilisateur de jouer sur la finesse de la partition s'il considère que le nombre de classes obtenues est trop élevé ou trop faible.
- permettre la découverte de structures ne présentant pas une régularité dans le nombre d'objets par classe contrairement à [Hua97]
- permettre la gestion de données manquantes, l'introduction de contraintes et l'intervention de l'utilisateur au cours du processus de cns et ce de manière aisée.

L'objectif du travail mené est donc la mise au point d'une méthode de cns permettant la mise à jour de partitions pertinentes et possédant ces qualités pratiques. La méthode ainsi mise au point a été nommée KEROUAC, pour sa présentation au Workshop International DATA MINING FOR ACTIONABLE KNOWLEDGE ("Fouille de Données pour Connaissances Utilisables") mené conjointement à la conférence PAKDD2003 (Pacific-Asia Conference on Knowledge Discovery and Data Mining)[4]. KEROUAC correspond à un acronyme pour les termes anglais **K**nowledge **E**xplicit, **R**apid, **O**ff-beat, **U**ser-centered, **A**lgorithm for **C**lustering, ces termes faisant référence aux principales qualités de la méthode : caractérisation explicite des classes mises à jour (**K**nowledge **E**xplicit), coût calculatoire relativement faible (**R**apid), bonne utilisabilté (**U**ser-centered).

3.2.2.2 L'Algorithme de Classification Non Supervisée

L'algorithme que nous proposons consiste en une mise en œuvre astucieuse et nouvelle de principes et techniques existants afin d'atteindre les objectifs détaillés précédemment. Pour cela nous utilisons le critère $NCC^{\star}$, et une technique de type graphes d'induction pour découvrir une partition $P_{\sim nat}$ proche ou égale à P_{nat} ce qui conférera un aspect non hiérarchique à la méthode

4. Ce travail a également fait l'objets de deux autres publications [JN03a] et [JN03b]

([WL59] avaient proposé une technique de type arbre d'induction ce qui conférait un aspect hiérarchique à la méthode et utilisaient un critère de type khi2).

REMARQUES :

- La découverte de P_{nat} (problème combinatoire) peut être résolue par des méthodes au coût calculatoire élevée : par une approche de type Programmation en nombre entier, par une approche basée sur les méthodes de Plans de Coupe et Branch and Bound [GW89].
- La découverte de $P_{\sim nat}$ peut être effectuée grâce à des heuristiques de coût calculatoire en $O(n^2)$: Nicoloyannis a proposé une approche itérative basée sur la prétopologie [Nic88], Nicoloyannis et al. utilisent une méthode itérative utilisant le recuit-simulé permettant la découverte de $P_{\sim nat}$ [NTT98], Michaud et Marcotorchino [MM81] utilisent une approche de type programmation linéaire, plus tard Michaud propose une méthode au coût calculatoire en $O(n)$ [Mic97].
- Ces méthodes ne possèdent pas les éléments d'utilisabillité désirés.

Nous avons donc adopté une heuristique gloutonne de type graphe d'induction (on procède par segmentation/fusion successives de classes de partitions). En définitive, à partir de la partition grossière de O, l'algorithme va évoluer itérativement de partition en partition en suivant le principe d'évolution suivant : on passe d'une partition P_i vers la partition P_{i+1} appartenant à son voisinage (cf. Définition 4 page 13) telle qu'elle réduise au plus le $NCC^\star$. L'algorithme s'achève lorsque $P_i = P_{i+1}$. Le pseudo-code de l'algorithme est donc :

Algorithme 1 Algorithme de Classification Non Supervisée

Paramètres : α (le facteur de granularité)

1. soit P_0 la partition grossière de O

2. i:=0

3. Déterminer $Vois(P_i)$

4. Déterminer $P_{i+1} \in Vois(P_i)$ la meilleure partition de $Vois(P_i)$ selon le $NCC^\star$

5. Si $P_{i+1} = P_i$ aller en 6), sinon i:=i+1 et aller en 3)

6. $P_{\sim nat} = P_i$

REMARQUES : Pour éviter des minima locaux, on peut autoriser dans le cas $P_{i+1} = P_i$, un nombre fixé a priori d'évolutions vers la partition du voisinage de P_i obtenue par segmentation selon une variable et impliquant la plus faible augmentation du $NCC^\star$.

3.2.2.3 Complexité de l'Algorithme

La complexité de l'algorithme est équivalente à celle des graphes d'induction. En effet, bien que la définition du critère $NCC^{\star}$ semble impliquer une comparaison de l'ensemble des couples d'objets (soit une complexité quadratique selon le nombre d'objets du jeu de données pour l'évaluation de la valeur du critère $NCC^{\star}$ pour une partition donnée), des astuces calculatoires (illustrées par la suite) permettent de réduire le coût calculatoire associé à l'évaluation de la valeur du critère $NCC^{\star}$ pour une partition. Ce coût n'est alors que linéaire selon le nombre d'objets du jeu de données.

En effet, la seule connaissance de la composition de chacune des classes d'une partition permet de déterminer sa valeur pour $NCC^{\star}$.

Considérons par exemple la partition P_h :
$P_h = \{C_1,C_2,C_3\} = \{\{ICEL,FRAN,SWED\},\{USA,UK\},\{POLA,USSR,CUBA\}\}$.
Soient :

- $n_{C_{i_{k,j}}}$ le nombre d'objets de la classe C_i ayant la valeur v_{jk} pour la variable $V_j \in EV$
- $n_{O_{k,j}}$ le nombre d'objets de O ayant la valeur v_{jk} pour la variable $V_j \in EV$.

On peut pour chaque valeur v_{jk} de chacune des variables $V_j \in EV$ déterminer $n_{O_{k,j}}$; pour chaque valeur v_{jk} de chacune des variables $V_j \in EV$ et chaque classe C_i de P_h déterminer $n_{C_{i_{k,j}}}$. Ces valeurs sont données dans le tableau 3.1.

	$M1$				$M2$				$M3$		
	A	B	C	D	A	B	C	D	A	B	C
C_1	0	0	3	0	0	3	0	0	1	0	2
C_2	0	0	2	0	0	0	2	0	2	0	0
C_3	3	0	0	0	2	0	0	1	0	0	3
O	3	0	5	0	2	3	2	1	3	0	5

TAB. 3.1 –: *Elements d'illustration de la complexité algorithmique*

On peut montrer que :

$$Sim(C_i,C_j) = \sum_{l=1}^{p} \sum_{k=1}^{card(Dom(V_l))} n_{C_{i_{k,l}}} \times n_{C_{j_{k,l}}} \quad (3.4)$$

$$Dissim(C_i) = \sum_{l=1}^{p} \sum_{k=1}^{card(Dom(V_l))} \frac{n_{C_{i_{k,l}}} \times (card(C_i) - n_{C_{i_{k,l}}})}{2} \quad (3.5)$$

Ce qui permet de calculer la valeur du critère $NCC^{\star}$. Ainsi,
$Sim(C_1,C_2) = (0\times0+0\times0+3\times2+0\times0)+(0\times0+3\times0+0\times2+0\times0)+(1\times2+0\times0+2\times0) = 8$
$Sim(C_1,C_3) = (0\times3+0\times0+3\times0+0\times0)+(0\times2+3\times0+0\times0+0\times0)+(1\times0+0\times0+2\times3) = 6$

$Sim(C_2,C_3) = (0\times3+0\times0+2\times0+0\times0)+(0\times2+0\times0+2\times0+0\times1)+(2\times0+0\times0+0\times3) = 0$
$Dissim(C_1) = (\frac{0\times3}{2}+\frac{0\times3}{2}+\frac{3\times0}{2}+\frac{0\times3}{2})+(\frac{0\times3}{2}+\frac{3\times0}{2}+\frac{0\times3}{2}+\frac{0\times3}{2})+(\frac{1\times2}{2}+\frac{0\times3}{2}+\frac{2\times1}{2}) = 2$
$Dissim(C_2) = (\frac{0\times2}{2}+\frac{0\times2}{2}+\frac{2\times0}{2}+\frac{0\times2}{2})+(\frac{0\times2}{2}+\frac{0\times2}{2}+\frac{2\times0}{2}+\frac{0\times2}{2})+\frac{2\times0}{2}+\frac{0\times2}{2}+\frac{0\times2}{2}) = 2$
$Dissim(C_3) = (\frac{3\times0}{2}+\frac{0\times3}{2}+\frac{0\times3}{2}+\frac{0\times3}{2})+(\frac{2\times1}{2}+\frac{0\times3}{2}+\frac{0\times3}{2}+\frac{1\times2}{2})+(\frac{0\times3}{2}+\frac{0\times3}{2}+\frac{3\times0}{2}) = 2$
d'où
$NCC(P_h) = 8+6+0+2+2+2$

Le calcul de la valeur du critère $NCC^\star$ possédant une complexité seulement linéaire dans le nombre d'objets du jeu de données, notre méthode possède donc bien une complexité équivalente à celle des graphes d'induction.

3.2.2.4 Qualités de la Méthode pour l'Utilisateur

- Chacune des classes de la partition résultat $(P_{\sim nat})$ est caractérisée par une règle logique (règle formée de disjonctions de conjonctions de sélecteurs) correspondant à la suite de mécanismes (fusion / segmentation) lui ayant donné le jour. Cette règle correspond alors pour un objet de O à une condition nécessaire et suffisante pour appartenir à la classe qu'elle décrit. (voir l'exemple page suivante)
- On associera également à chaque classe de $P_{\sim nat}$ son mode, celui-ci correspondant en définitive à une sorte de profil ou individu type de la classe.
- Ces deux premiers points simplifient largement la compréhension et l'interprétation des résultats.
- La description de chaque classe par une règle logique peut également permettre l'assignation d'un nouvel objet à l'une des classes sans pour autant connaître l'ensemble de ses caractéristiques.
- le nombre de classes est déterminé automatiquement par la méthode, toutefois l'utilisateur peut influer sur la finesse de la partition finalement produite par l'intermédiaire du facteur de granularité. (Si le nombre de classes apparaît trop faible (resp. trop fort) à l'utilisateur, ce dernier peut procéder à une nouvelle cns en augmentant (resp. en diminuant) la valeur du facteur de granularité).

3.2.2.5 Illustration du Fonctionnement de l'Algorithme

Considérons un jeu de données classique : les données mushrooms[MM96]. Ce jeu de données est composé de 8124 objets (en l'occurrence des champignons), chacun décrit par 23 variables. Chaque objet est, de plus, identifié par sa comestibilité (voir page 217 pour une présentation complète du jeu de données). Le jeu de données est ainsi composé de champignons comestibles et de champignons vénéneux.

La figure 3.2 présente le processus de cns sur le jeu de données "Mushrooms" pour un facteur de granularité α valant 1. Elle détaille ainsi l'ensemble des processus de segmentation/fusion, permettant l'obtention de la cns. Voici pour exemple les règles logiques caractérisant les classes C8 et C10 (rappelons que

chacune de ces règles détermine l'appartenance ou la non appartenance de tout objet à la classe à laquelle elle est associée) :

- C8 [Ring_Type = evanescent] ET [Gill_Size = broad] ET [Bruises? = bruises]
- C10 [Ring_Type = pendant] ET [[Bruises? = bruises] OU [[Bruises? = no] ET [stalk-color-above-ring = white] ET [Gill_Size = narrow]]]

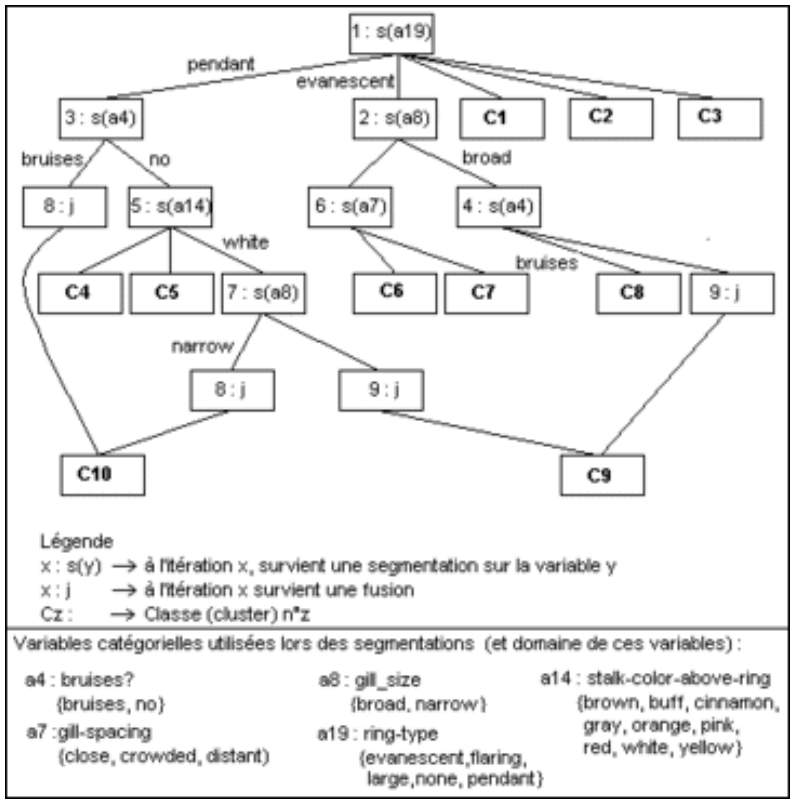


FIG. 3.2 –: *Illustration du Fonctionnement de l'Algorithme*

3.2.3 Evaluation de l'Algorithme de Classification non Supervisée

Classiquement, une méthode de cns s'évalue selon :

- la validité des classifications qu'elle propose,
- la stabilité des classifications qu'elle propose,
- son efficacité algorithmique.

Nous proposons maintenant une évaluation expérimentale de chacun de ces points.

3.2.3.1 Evaluation de la Validité des Classifications

La nature non supervisée de la cns n'autorise pas une définition claire et directe de ce que sont des structures/organisations valides. Ainsi, les mul-

tiples algorithmes de cns se caractérisent par l'ensemble d'hypothèses qu'ils emploient afin de définir les propriétés devant être satisfaites par une structure valide. L'ensemble d'hypothèses déterminant la validité d'une structure n'étant pas universel et différant selon les méthodes, les résultats varient selon la méthode utilisée. Conséquemment, il est essentiel de définir une méthode d'évaluation des structures resultant d'un processus de cns. Ce type d'évaluation est nommé évaluation de la validité d'une cns.

L'évaluation de la validité d'une cns est généralement réalisée par l'utilisation de mesures de validité de cns (voir [HBV01] et le chapitre 4 pour des présentations de l'évaluation de la validité des cns). Ces mesures sont de deux types : externes et internes (les modes d'évaluation habituels correspondant en définitive à l'utilisation implicite d'une mesure externe). Les critères externes de validité évaluent dans quelle mesure le résultat du processus de cns correspond à des connaissances avérées sur les données. De manière assez générale, on admet que ces informations ne sont pas calculables à partir des données. La forme la plus commune de données de ce type est un ensemble d'étiquettes que l'on associe à chacun des objets (ce dernier type d'information peut éventuellement être obtenu par une classification manuelle). Les critères internes de validité consistent quant à eux en une mesure basée uniquement sur le traitement des données servant au processus de cns. Le choix de l'utilisation de l'un ou l'autre type de mesures (pour procéder à l'évaluation de la validité) est essentiellement pragmatique : si l'on dispose d'informations permettant de caractériser la structure devant être extraite, l'utilisation de mesures externes est alors privilégiée tandis qu'en cas d'absence de ce type d'informations l'unique moyen disponible pour l'évaluation de la validité est l'utilisation de mesure internes.

REMARQUES : Si l'utilisation d'une mesure externe n'est pas envisageable en pratique, i.e. lorsqu'il s'agit de traiter un jeu de données dont on ne connaît pas la structure sous-jacente, l'utilisation de ce type de mesure sur un jeux de de données dont on connaît la structure constitue par contre une solution pour l'évaluation de la capacité d'un algorithme donné à découvrir cette structure ou encore pour déterminer la validité des cns obtenues par un algorithme donné.)

Expérience #1 Nous avons utilisé ici une évaluation de type mesure externe largement utilisée dans la littérature. Nous avons considéré le jeu de données mushrooms (composé de champignons comestibles et de champignons vénéneux) et avons réalisé plusieurs cns, pour des valeurs de α différentes, enfin nous avons utilisé le concept "comestibilité" et le taux de correction (T.C.) des cns par rapport à ce concept afin de caractériser la qualité de la cns résultant (la variable définissant la "comestibilité" n'étant évidemment pas introduite dans le processus de cns) (le taux de correction d'une cns par rapport à un concept donné correspond à la pureté de cette cns par rapport à ce concept).

Les résultats obtenus pour 3 valeurs différentes de α ($\alpha = 1,2,3$) sont présentés dans la figure 3.3 ; ils montrent que :

- les différentes classifications réalisées permettent bien d'obtenir des partitions reflétant correctement la structure impliquée par le concept comestibilité,
- la capacité de l'algorithme à déterminer une structure présentant des irrégularités dans le nombre d'objets par classe (voir figures 3.3, 3.4).

(Notons de plus que pour des valeurs de α supérieures à 3, les classifications obtenues présentaient un nombre de classes strictement supérieur à 24, chacune étant homogène du point de vue de la comestibilité des champignons la constituant.)

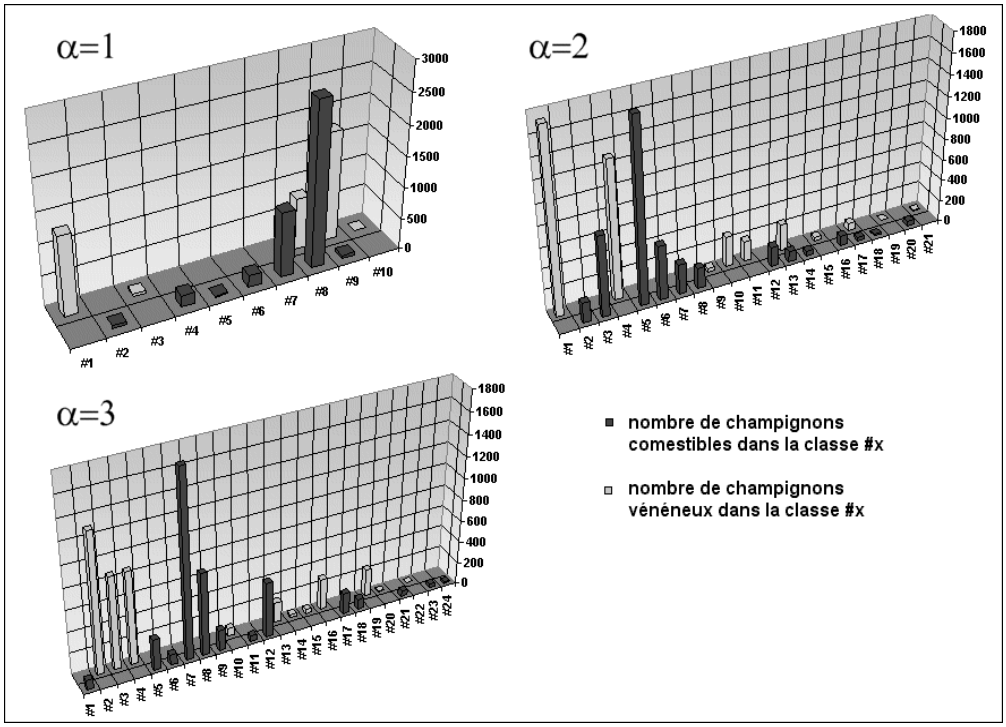


FIG. 3.3 –: *Composition en terme de comestibilité des classes de 3 cns différentes ($\alpha = 1$, $\alpha = 2$, $\alpha = 3$) du jeu de données Mushrooms*

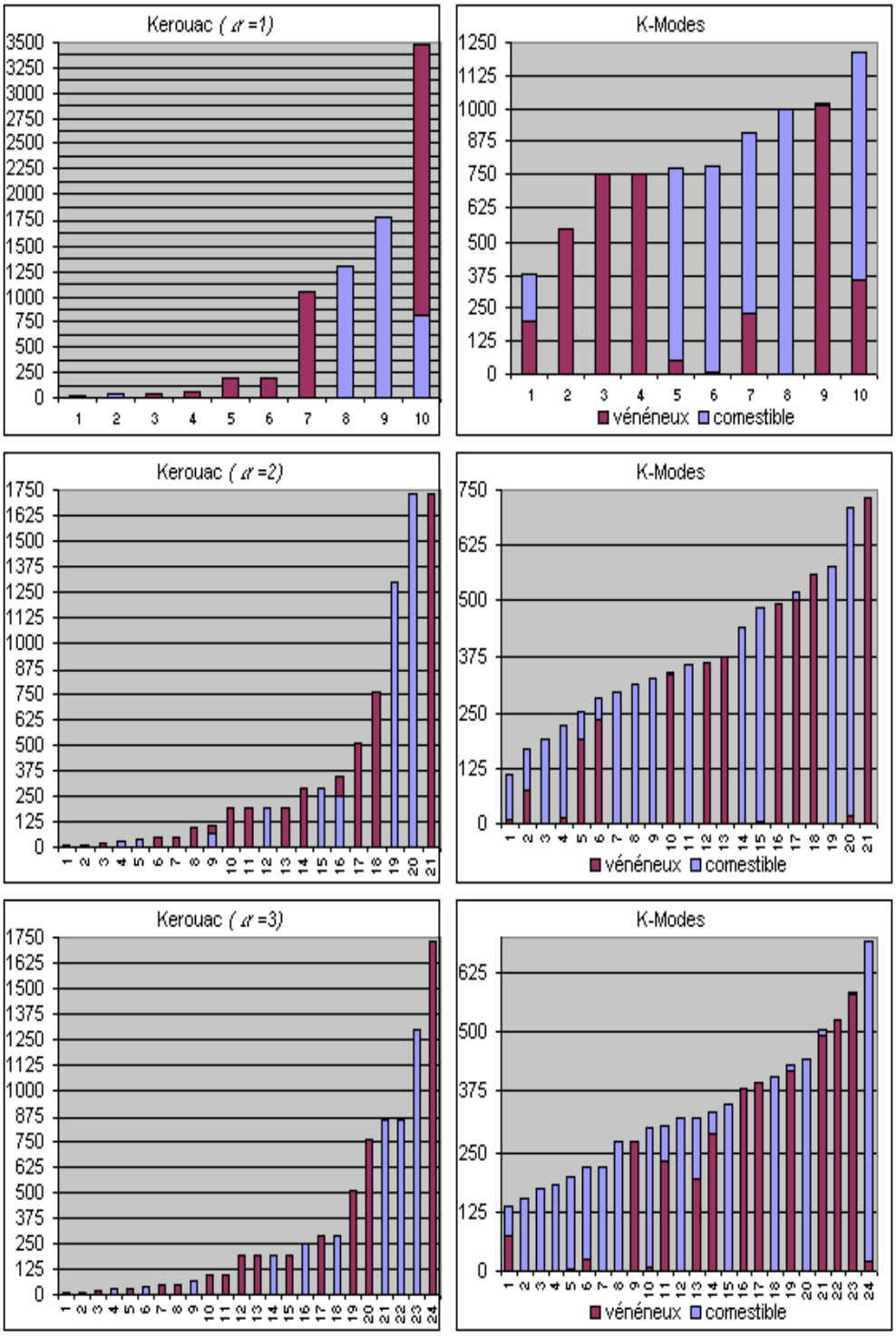


FIG. 3.4 –: *Composition en terme de comestibilité des classes de 6 cns différentes du jeu de données Mushrooms ($\alpha = 1$, $\alpha = 2$, $\alpha = 3$ pour* KEROUAC, *nombre de classes = 10, 21, 24 pour les K-Modes)*

Expérience #2 Nous présentons également une comparaison des résultats obtenus par notre méthode et ceux obtenus par les k-modes pour différentes cns. Nous avons, pour cela, lancé plusieurs processus de cns avec notre méthode en utilisant des facteurs de granularité différents. Cela nous a permis d'obtenir des cns en 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 58, 141 et 276 classes. Les taux de correction de ces cns par rapport au concept "comestibilité" sont ainsi reportés sur la figure 3.5.

Ensuite nous avons lancé des séries de 10 cns en utilisant les k-modes paramétrés de manière telle qu'on obtienne des cns en 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 58, 141 et 276 classes. Les taux de corrections concernant chacune de ces cns sont détaillés sur la figure 3.5. (Notons que nous donnons le taux de correction pour chacune des cns possédant entre 2 et 25 classes et que nous n'indiquons que la moyenne du taux de correction des séries de 10 cns possédant un nombre de classes strictement supérieur à 25. De plus, nous indiquons pour chaque série de 10 cns possédant entre 2 et 25 classes le taux de correction de la "meilleure" cns de la série (i.e. la cns possédant la plus faible valeur pour le critère QKM, voir page 20) pour la définition du critère QKM).

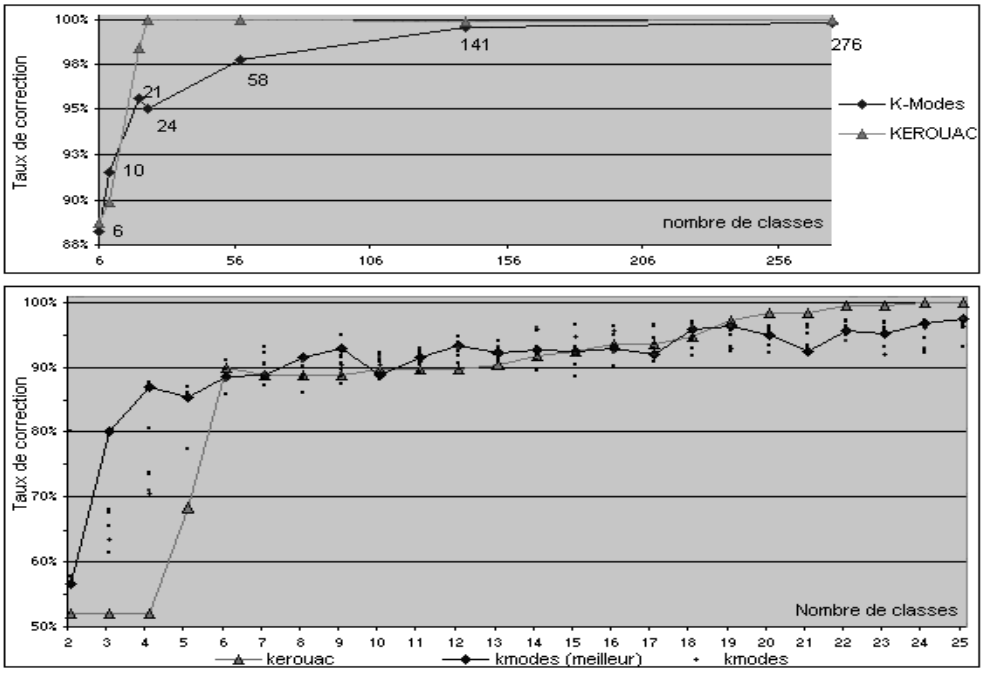


FIG. 3.5 –: *Taux de correction par rapport au concept "comestibilité" pour différentes cns obtenues par application des K-Modes, ou de* KEROUAC

L'ensemble de ces tests montrent une qualité légèrement supérieure pour les cns obtenues par l'intermédiaire de notre méthode. (Ils mettent également en évidence la sensibilité de la méthode K-Modes à l'initialisation : le taux de

correction d'une cns en un nombre donné de classes est susceptible de varier plus ou moins fortement selon l'initialisation de l'algorithme.)

Expérience #3 Des tests ont également été menés sur le jeu de données Soybean Disease [MM96]. Soybean Disease est un jeu de données standard en apprentissage symbolique (machine learning) composé de 47 objets, chacun étant décrit par 35 variables catégorielles. Chaque objet est caractérisé par une des 4 pathologies suivantes : Diaporthe Stem Canker (D1), Charcoal Rot (D2), Rhizoctonia Root Rot(D3), and Phytophthora Rot(D4). A l'exception de D4 qui est représentée par 17 objets, toutes les autres pathologies sont représentées par 10 objets chacune (voir page 217 pour une présentation complète du jeu de données).

Nous avons mené plusieurs cns pour différentes valeurs de α et utilisé le concept "pathologie" pour caractériser la qualité des cns obtenues (la variable "pathologie" n'étant évidemment pas introduite dans le processus de cns). Les résultats obtenus pour 4 valeurs différentes de α ($\alpha = 1$,1.5,2,3) présentés dans la figure 3.6 montrent que les cns obtenues reflètent correctement le concept "pathologie".

Pour $\alpha \geq 3$, les cns ont un nombre de classes strictement supérieur à 4, chaque classe étant homogène du point de vue du concept "pathologie". Nos résultats pour une cns en 4 classes (taux de correction égal à 100%) sont meilleurs que ceux des k-modes reportés dans [Hua97], (taux de correction à peu près égal à 96% pour les k-modes) et à ceux des expérimentations que nous avons menées dont les résultats sont reportés dans la figure 3.7.

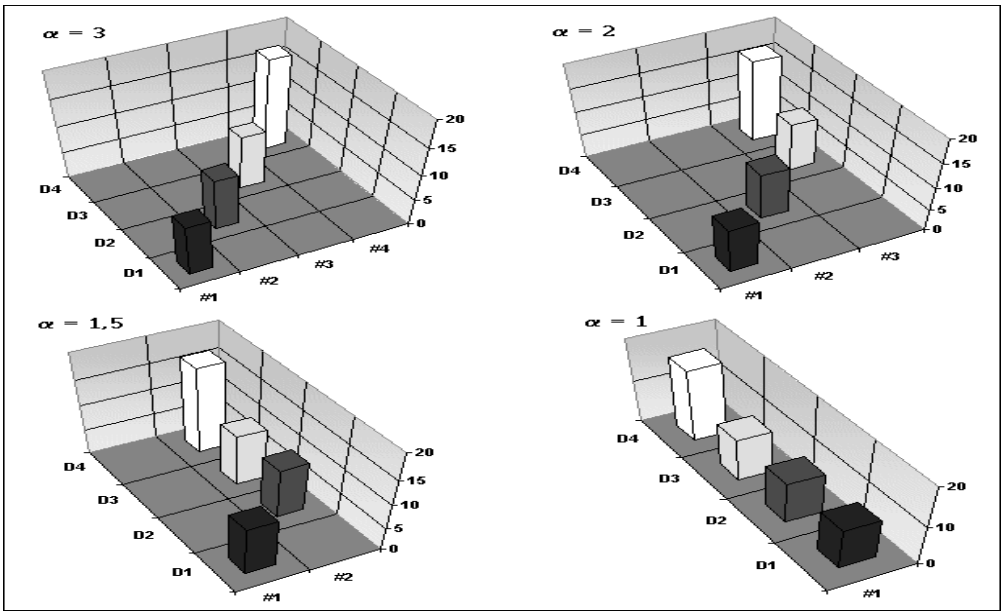


FIG. 3.6 –: *Composition en terme de pathologie des classes de 4 cns différentes ($\alpha = 1$, $\alpha = 1.5$, $\alpha = 2$, $\alpha = 3$) du jeu de données Soybean Diseases*

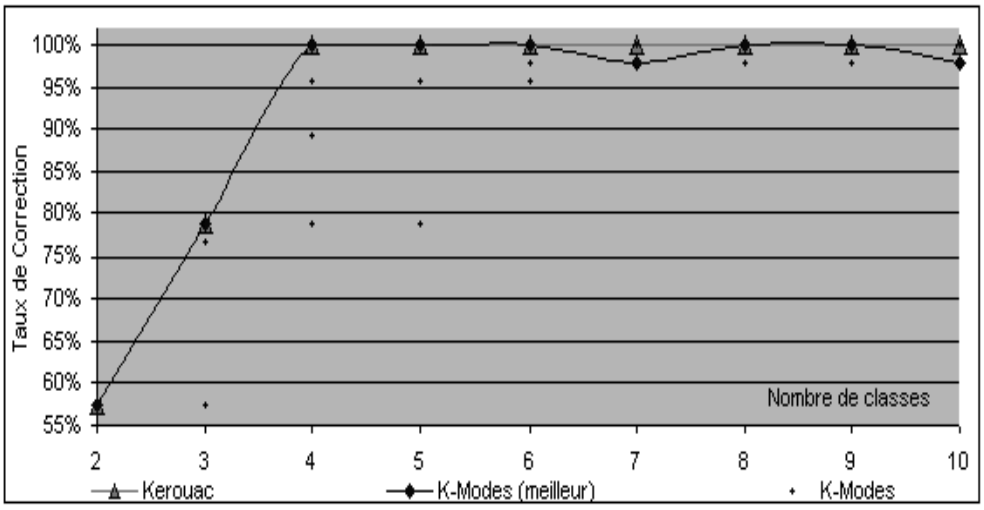


FIG. 3.7 –: *Taux de correction par rapport au concept "pathologie" pour différentes cns obtenues par application des K-Modes, ou de* KEROUAC

Expérience #4 Des expériences impliquant une méthode d'évaluation que nous avons mise au point et présentée au chapitre suivant nous ont permis d'évaluer et de comparer de manière globale la validité de cns obtenues par notre méthode et par la méthode K-Modes. Ces résultats (présentés page 95) semblent indiquer que notre méthode fournit généralement des cns possédant un niveau de validité supérieur à celles obtenues par application des K-Modes[5].

3.2.3.2 Evaluation de la Stabilité

Un autre point d'évaluation d'un algorithme de cns, est l'évaluation de sa stabilité (plus précisément de la stabilité de ses résultats), ce qui signifie à répondre à la question "aurai je obtenu une organisation des objets similaire ou très proche si l'ensemble d'objets que j'avais utilisé avait été légèrement différent (quelques objets supplémentaires ou en moins)?".

Afin de répondre à cette question, des méthodes d'échantillonnage et de comparaison des partitions obtenues ont été présentées, nous utiliserons ici celle présentée dans [LD01], son mode de fonctionnement est le suivant :

– on considère le jeu de données dans son intégralité et l'on réalise une première cns qui constituera la classification de référence P_{Ref} (le nombre d'objets du jeu de données est noté n).

5. Nous ne détaillons pas ici l'analyse des résultats de ces expériences, car ceux-ci sont présentés ultérieurement et nécessitent la présentation de la méthodologie utilisée pour l'évaluation/comparaison de validité de cns

- On réalise ensuite un ensemble EC de p cns (P_i,$i = 1..p$) sur des échantillons (tirés au hasard) du jeu de données de taille $\mu \times n$ ($\mu \in]0,1]$, μ est appelé facteur de dilution).
- Pour chaque cns P_i, $i = 1..p$ on procède à une comparaison avec P_{Ref} afin de calculer la proportion de paires d'objets (notée $prop_i$) traitées différemment par P_{Ref}. On dit qu'une paire d'objets est traitée différemment par P_{Ref} et P_i, si les deux objets sont présents dans l'échantillon ayant permis de bâtir P_i et si ces deux objets sont regroupés au sein d'une même classe dans P_i alors qu'ils ne l'étaient pas pour P_{Ref} ou si ces deux objets ne sont pas regroupés au sein d'une même classe dans P_i alors qu'ils l'étaient pour P_{Ref}. ($prop_i \in [0,1]$)
- On calcule la valeur d'un indicateur de stabilité de la cns $Stab$ ($Stab \in [0,1]$)qui correspond à la moyenne des $prop_i$.

Ainsi, une valeur élevée de $Stab$ (relativement proche de 1) correspondra à une forte différence entre les cns de EC et P_{Ref}, et donc une valeur faible de $Stab$ (proche de 0) correspondra à une faible différence entre les cns de EC et P_{Ref}. La valeur de $Stab$ permet alors de savoir si les résultats de l'algorithme de cns peuvent être considérés comme stables et l'utilisation de l'algorithme valable (la non stabilité impliquant la non utilisabilité de la méthode ou une recherche en profondeur des causes de la non stabilité).

Les tests de stabilité de l'algorithme réalisés sur le jeu de données "Mushrooms" sont présentés dans la figure 3.8 et permettent d'une part d'évaluer la stabilité des cns proposées par KEROUAC et d'autre part d'évaluer la stabilité des cns proposées par la méthode K-Modes. Plus précisément, ces tests correspondent aux expérimentations suivantes :

Pour KEROUAC :

- 11 séries de 25 cns ont été réalisées sur des échantillons aléatoires du jeu de données (un échantillon différent pour chaque cns de chaque série). Chacune des séries correspond à une taille d'échantillon particulière (et donc à une valeur du facteur de dilution). Les différentes tailles d'échantillons sont respectivement 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%, 5%, 1% du jeu de données (correspondant respectivement à une valeur de 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.01 pour le facteur de dilution). Pour chaque cns le facteur de granularité α a été fixé à 1.
- Une cns de référence a été obtenue en appliquant Kerouac au jeu de données "complet" , la cns ainsi obtenue possède 10 classes. (Notons que, pour un même jeu de données et une même valeur du facteur de granularité, KEROUAC fournit toujours la même cns. Ainsi, pour un facteur de dilution valant 1, la valeur de l'indice de stabilité est donc 0.).
- Puis la valeur de l'indice de stabilité a été évaluée pour chacune de ces cns. Ainsi, pour chaque série de 25 cns, la valeur moyenne de l'indice de stabilité ainsi que son écart-type sont calculés.
- De plus, pour chaque série de 25 cns, le nombre moyen de classes par cns ainsi que son écart-type sont calculés.

Pour la méthode des K-Modes :

- 12 séries de 25 cns ont été réalisées sur des échantillons aléatoires du jeu de données (un échantillon différent pour chaque cns de chaque série). Chacune des séries correspond à une taille d'échantillon particulière (et donc à une valeur du facteur de dilution). Les différentes tailles d'échantillons sont respectivement 100%, 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%, 5%, 1% du jeu de données (correspondant respectivement à une valeur de 1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.01 pour le facteur de dilution). Pour chaque cns le nombre de classes a été fixé à 10 de manière à obtenir des cns possédant le même nombre de classes que la cns de référence obtenue pour les K-Modes. (Le facteur de granularité a, en fait, été fixé à 1 pour Kerouac afin d'obtenir une cns en 10 classes, car pour ce jeu de données les auteurs utilisant les K-Modes indiquent qu'il s'agit du "meilleur" nombre de classes.)
- Une cns de référence a été obtenue en sélectionnant parmi les 25 cns de la série de cns sur le jeu de données "complet" (taille de l'échantillon = 100% du jeu de données), la cns possédant la plus faible valeur du critère QKM (critère à minimiser sous-jacent à la méthode des K-Modes) (Cette cns peut donc être considérée comme la meilleure cns obtenue, par application des K-Modes, sur le jeu de données complet).
- Puis la valeur de l'indice de stabilité a été évaluée pour chacune de ces cns. Ainsi, pour chaque série de 25 cns, la valeur moyenne de l'indice de stabilité ainsi que son écart-type est calculée.
- De plus, pour chaque série de 25 cns, nous avons recherché la cns correspondant à la "meilleure" cns (i.e. celle possédant la plus faible valeur pour le critère QKM) et indiqué sa valeur pour l'indice de stabilité.

Les résultats de ces expériences sont présentés sur la figure 3.8. Notons que la valeur de l'indice de stabilité est indiquée en pourcentage. Ainsi exprimée, elle indique le pourcentage de couples d'objets traités différemment par la cns de référence et les cns pour lesquelles la stabilité est évaluée.

Ces tests montrent une excellente stabilité des cns obtenues par notre méthode : quel que soit le niveau d'échantillonnage ($\geq 1\%$), moins de 0.5% des couples d'objets sont, en moyenne, traités différemment par la cns de référence et par les cns évaluées. Comparativement, la méthode des K-Modes est, quant à elle, à peu près 20 fois moins efficace...

La moindre efficacité des K-Modes peut s'expliquer d'une part par sa sensibilité au processus d'initialisation et d'autre part par une efficacité intrinsèque moins bonne due à la mesure de similarité qu'elle utilise. Cette dernière raison est sans doute la principale : si l'on étudie la stabilité associée aux "meilleures" cns de chaque série on peut observer que bien que l'on observe un accroissement global du niveau de stabilité, celui-ci reste toutefois à peu près 16 fois moins bon que le niveau de stabilité de notre méthode.

Notons également que, naturellement le niveau de stabilité est sensible au niveau de dilution mais qu'il apparaît toutefois très stable. Enfin, la stabilité du

point de vue du nombre de classes de chacune des cns obtenues par KEROUAC est également excellente :

- pour des niveaux de dilution supérieurs à 0.2 le nombre de classes de chacune des cns obtenues est quasiment toujours égal au nombre de classes de la cns de référence ;
- pour des niveaux de dilution inférieurs à 0.2 le nombre de classes de chacune des cns obtenues est en général quelque peu plus faible que le nombre de classes de la cns de référence, cela pouvant s'expliquer par l'éventuelle quasi-absence dans l'échantillon traité d'objets appartenant à l'une des 10 classes de la cns de référence (en effet, pour ces niveaux de dilution la taille des échantillons d'objets traités est inférieure à 20% du jeu de données complet, or 6 classes de la cns de référence comporte moins de 2.5% des individus).

Ce dernier point (la stabilité du nombre de classes par cns) montre la réelle supériorité de notre méthode par rapport à celle des K-Modes : en effet si l'on avait pu penser que la différence de stabilité entre les méthodes pouvait peut-être s'expliquer par la forme de la cns de référence (qui dans le cas de KEROUAC présente 4 classes regroupant la quasi-totalité des individus, voir figure 3.4), la stabilité du nombre de classes par cns permet de rejeter cet argument.

REMARQUES : Il est clair que les tests de stabilité des algorithmes ne donnent pas uniquement lieu à une évaluation de la stabilité des algorithmes et qu'ils prennent également en compte la nature intrinsèque des données à permettre la mise à jour de structure stable. Cependant, la différence de stabilité des algorithmes KEROUAC et K-Modes sur un même jeu de données (Mushrooms) (voir figure 3.4) montre bien que ce type de test permet véritablement de caractériser la plus grande d'un stabilité d'un algorithme par rapport à un autre...

3.2.3.3 Evaluation de l'Efficacité Algorithmique

Nous ne présentons pas ici une étude poussée du coût calculatoire de la méthode, nous nous contentons de préciser que la complexité algorithmique de notre méthode est équivalente à celle des graphes d'induction largement utilisés en E.C.D.[6] et présentons le temps de calcul associé à différentes cns pour le jeu de données "mushrooms". En fait, nous ne présentons pas explicitement les temps de calcul associés aux cns mais le rapport suivant :
$R = \frac{temps\ de\ calcul\ associé\ à\ la\ cns}{temps\ de\ calcul\ associé\ à\ la\ cns\ en\ 6\ classes\ par\ les\ K-modes}$.

Les rapports présentés pour les K-modes sont les valeurs moyennes obtenues pour des séries de 10 cns. Ces résultats montrent que les K-Modes (qui sont reconnus comme une méthode rapide et possédant une bonne scalabilité) sont plus rapides pour de faibles nombres de classes mais que les 2 méthodes semblent se comporter de manière similaire pour des nombres de classes plus

6. la méthode possède donc une complexité log-linéaire selon le nombre d'objets du jeu de données traité, et linéaire selon le nombre de variables du jeu de données

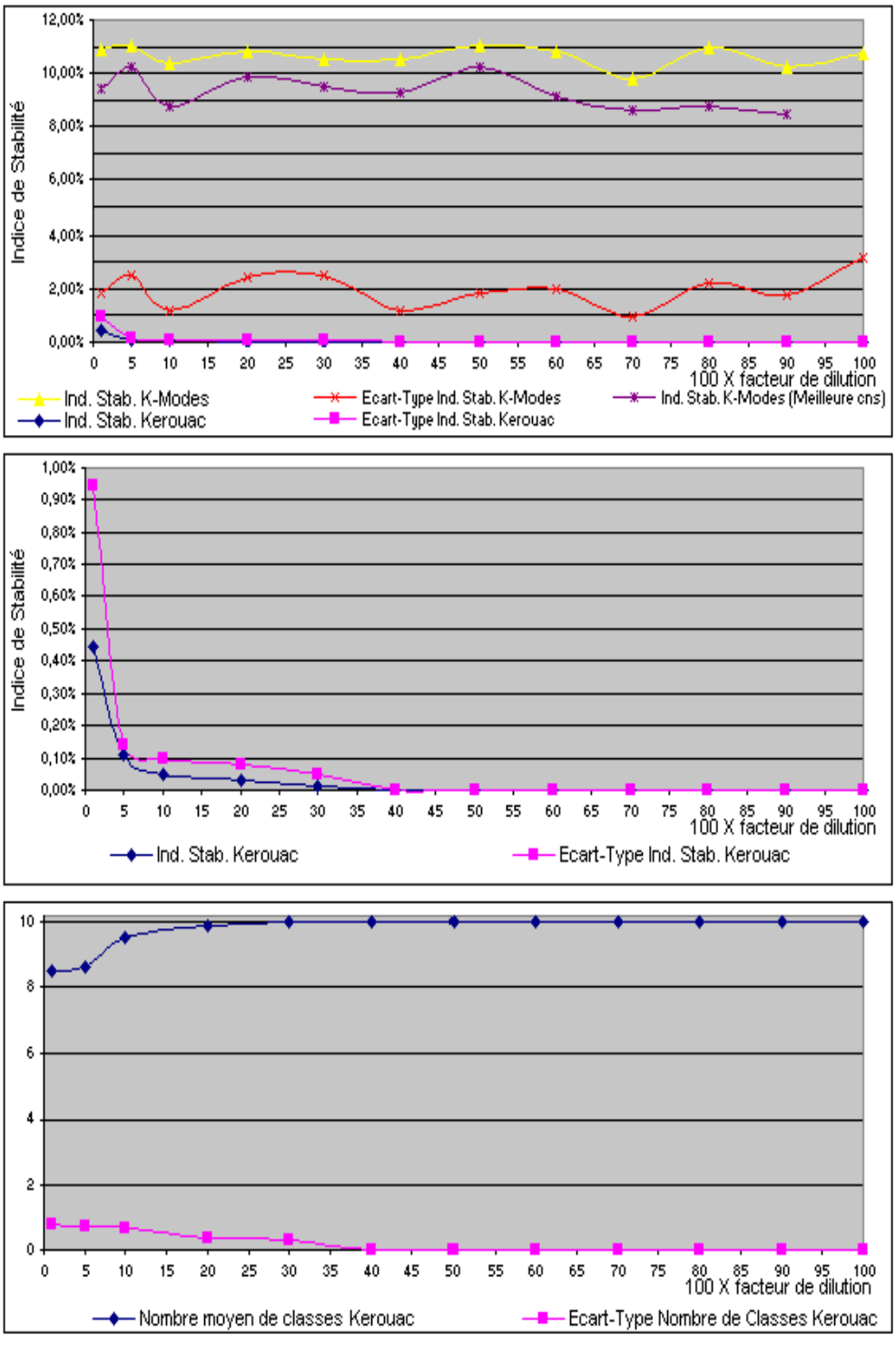


FIG. 3.8 –: *Evaluation de la stabilité pour le jeu de données Mushrooms*

élevés. En définitive, en observant les résultats expérimentaux et en y associant la complexité algorithmique théorique, on peut conclure que KEROUAC présente une efficacité algorithmique tout à fait correcte.

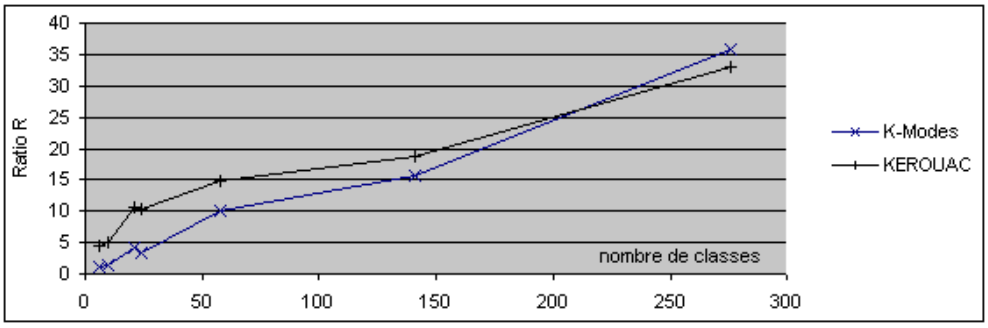


FIG. 3.9 –: *Rapports R associés à différentes cns*

3.2.4 Eléments Additionnels

Nous venons de procéder à l'évaluation de notre méthode de cns qui nous a permis de mettre en avant la validité des structures découvertes, la très bonne stabilité et le coût calculatoire relativement faible de la méthode. Ces différents points constituent plusieurs points forts, tout comme les avantages concernant l'utilisabilité cité à la section 3.2.2.4. Nous listons un ensemble d'autres points que nous allons maintenant aborder et qui participent également à rendre cette méthode très attrayante du point de vue de l'utilisateur :

- la présence de données manquantes n'est pas gênante : leurs conséquences sur la classification est complètement paramétrable en codant l'implication particulière de la présence de données manquantes sur l'aspect naturel d'une partition,
- l'introduction de contraintes est possible par l'intermédiaire de l'utilisation de variables supplémentaires sur lesquelles on n'autorisera pas de segmentation lors du processus de recherche de la partition naturelle approchée (Ainsi, l'interactivité entre utilisateur et processus de cns est possible, par introduction de contraintes).

Nous explicitons maintenant comment les traitements des données manquantes et contraintes sont facilités grâce à notre méthode et par l'intermédiaire de l'introduction de valeurs spécifiques dans les domaines des variables catégorielles.

3.2.4.1 Valeurs Spécifiques pour le Domaine des Variables Catégorielles

Dans certaines situations, il peut apparaître nécessaire de modifier l'implication sur l'aspect naturel d'une partition que peut avoir une valeur particulière

du domaine d'une variable catégorielle (i.e. il peut être parfois intéressant que les opérateurs δ_{sim} et δ_{dissim} aient un comportement spécifique en cas de présence de valeurs particulières du domaine d'une variable catégorielle).

Considérons par exemple le cas de valeurs manquantes (notée ?) pour une variable catégorielle V_i. On peut déterminer plusieurs types d'implication sur l'aspect naturel d'une partition pour cette valeur particulière du domaine de V_i selon "le type d'information qu'elle véhicule", par exemple :

- Considérons qu'une variable décrive le nombre de roues d'un véhicule. Une valeur manquante pour cette variable signifie alors l'absence de roue, ce qui est vecteur d'information, si bien que cette valeur manquante peut être considérée comme une valeur classique. Dès lors si $o_{l_i} =?$ et $o_{m_i} =?$ alors $\delta_{sim}(o_{a_i},o_{b_i}) = 1 - \delta_{dissim}(o_{a_i},o_{b_i}) = 1$; par contre si $o_{l_i} =?$ et $o_{m_i} \neq?$ alors $\delta_{sim}(o_{a_i},o_{b_i}) = 1 - \delta_{dissim}(o_{a_i},o_{b_i}) = 0$. Ainsi l'implication sur l'aspect naturelle de cette valeur particulière est identique au cas classique.
- Considérons maintenant une variable quelconque et que la présence d'une valeur manquante soit la conséquence d'une erreur de saisie. Pourquoi considérerions nous alors que un objet présentant une valeur manquante pour cette variable est similaire à un autre objet présentant également une valeur manquante pour cette variable... ou encore, quelle bonne raison pourrait nous pousser à le considérer dissimilaire d'un objet pour lequel il n'y a pas eu d'erreur de saisie ? La valeur manquante ne peut alors être considérée comme une valeur classique, on peut ainsi vouloir qu'un objet présentant cette valeur pour V_i, ne soit considéré ni similaire ni dissimilaire des autres objets du point de vue de V_i. Dans ce cas, si deux objets o_a et o_b sont tels que $o_{a_i} =?$ et $o_{b_i} \neq?$ on obtient : $\delta_{sim}(o_{a_i},o_{b_i}) = \delta_{dissim}(o_{a_i},o_{b_i}) = 0$; de même si deux objets o_l et o_m sont tels que $o_{a_i} =?$ et $o_{b_i} =?$ on obtient : $\delta_{sim}(o_{a_i},o_{b_i}) = \delta_{dissim}(o_{a_i},o_{b_i}) = 0$. Ainsi l'implication sur l'aspect naturel est particulière dans la mesure où la présence de valeurs manquantes aura pour conséquence de ne pas impliquer le regroupement ou la séparation des objets présentant cette valeur particulière et de ne pas impliquer la séparation ou le les regroupement des objets ne la présentant pas.
- ... (on peut imaginer d'autres cas)

Nous avons défini (au chapitre 2 un ensemble de valeurs additionnelles particulières pouvant être ajoutées aux domaines des variables catégorielles de manière à "coder" des cas particuliers comme celui de la présence de valeurs manquantes. Ces valeurs particulières $\varepsilon_1,\varepsilon_2,\varepsilon_3,\varepsilon_4,\varepsilon_5,\varepsilon_6,\varepsilon_7$ sont inclues dans l'ensemble E_ε, elles permettent de coder un ensemble relativement vaste de cas particuliers dont nous allons donner quelques exemples dans le cas de leur utilisation pour le codage de données manquantes ou de contraintes (il est toutefois tout à fait possible d'étendre l'ensemble avec d'autres valeurs).

Nous présentons dans les tableaux 3.2, 3.3, 3.4, les valeurs pour $\delta_{sim}(o_{a_i},o_{b_i})$, $(1-\delta_{dissim}(o_{a_i},o_{b_i}))$, $\delta_{dissim}(o_{a_i},o_{b_i})$ dans différents cas. Tout d'abord dans le cas

classique où les valeurs prises par o_{a_i} et o_{b_i} sont des valeurs ne correspondant pas à des valeurs spécifiques mais aux différentes modalités (A,B,C,D) d'une variable V_i (voir tableau 3.2). Puis dans le cas où les valeurs prises par o_{a_i} et o_{b_i} sont soit l'une des 4 modalités possibles de V_i que l'on note λ ou bien l'une des valeurs spécifiques que nous introduisons (voir tableaux 3.3, 3.4).

Le tableau 3.2 expose le "comportement" classique de $\delta_{dissim}(o_{a_i},o_{b_i})$ et $\delta_{sim}(o_{a_i},o_{b_i})$ pour une variable V_i c'est à dire lorsque les valeurs de cette variable sont disponibles à la fois pour o_a et o_b (i.e. $\forall i$, $o_{a_i} \neq \varepsilon_i$ et $\forall i$, $o_{b_i} \neq \varepsilon_i$; les tableaux 3.3 et 3.4) montrent, quant à eux, le comportement de $\delta_{dissim}(o_{a_i},o_{b_i})$ et $\delta_{sim}(o_{a_i},o_{b_i})$ pour une variable V_i lorsque l'une ou les 2 valeurs de V_i est inclue dans E_ε.

3.2.4.2 Gestion des Valeurs Manquantes :

Nous l'avons illustré, il peut être intéressant de réserver un traitement particulier aux valeurs manquantes et cela peut être réalisé grâce à l'utilisation des valeurs spécifiques introduites précédemment (ε_1,ε_2,ε_3,ε_4,ε_5,$\varepsilon 6$,ε_7). Ainsi, si l'on veut, par exemple, qu'un objet o_a présentant une valeur manquante pour une variable V_i, ne soit considéré ni similaire ni dissimilaire des autres objets du point de vue de V_i (i.e. quel que soit un autre objet quelconque et quelle que soit sa valeur pour V_i, cet objet ne sera considéré ni comme similaire ni comme dissimilaire de o_a) on peut utiliser la valeur spécifique ε_4 pour coder la valeur manquante. En effet dans ce cas, si $o_{a_i} = \varepsilon_4$, alors quelle que soit la valeur o_{b_i} on obtient : $\delta_{sim}(o_{a_i},o_{b_i}) = \delta_{dissim}(o_{a_i},o_{b_i}) = 0$. Ainsi l'implication sur l'aspect naturel est particulière dans la mesure où la présence de valeurs manquantes aura pour conséquence de ne pas impliquer le regroupement ou la séparation des objets présentant cette valeur particulière et de ne pas impliquer la séparation ou le regroupement des objets ne la présentant pas.

3.2.4.3 Introduction de Contraintes :

La cns est une tâche subjective par nature : le même jeu de données peut nécessiter différents partitionnements afin de répondre à diverses attentes de l'utilisateur. Considérons par exemple les animaux suivants : éléphants, baleines et thons [Wat85]. Les éléphants tout comme les baleines sont des mammifères et peuvent par conséquents former une classe. Cependant, si l'utilisateur est intéressé par une classification basée sur l'habitat naturel de ces animaux, baleines et thons formeront alors une classe. Il peut donc parfois être utile d'introduire cette notion de subjectivité dans le processus de cns.

Ainsi, si lorsque l'on procède à une cns on ne sait pas par avance le résultat que l'on désire obtenir, il peut arriver que l'on ait toutefois des idées sur la forme du résultat voulu. On peut en effet posséder un ensemble de connaissances qui nous poussent par exemple à vouloir que plusieurs objets soient regroupés au sein d'une même classe, ou, a contrario, que certains objets appartiennent à des classes différentes. Afin d'obtenir un résultat conforme à

$\delta_{sim}(o_{a_i},o_{b_i})$

$o_a \backslash o_b$	A	B	C	D
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

$\delta_{dissim}(o_{a_i},o_{b_i})$

$o_a \backslash o_b$	A	B	C	D
A	0	1	1	1
B	1	0	1	1
C	1	1	0	1
D	1	1	1	0

$1 - \delta_{dissim}(o_{a_i},o_{b_i})$

$o_a \backslash o_b$	A	B	C	D
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

TAB. 3.2 –: *Comportement des opérateurs $\delta_{sim}(o_{a_i},o_{b_i})$, $\delta_{dissim}(o_{a_i},o_{b_i})$, $1 - \delta_{dissim}(o_{a_i},o_{b_i})$ pour des valeurs classiques*

$o_a \backslash o_b$	λ	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ε_7
λ	1	0	0	0	0	0	0	0
ε_1	0	1	0	0	0	0	0	0
ε_2	0	0	$0^\star$	0	0	0	0	0
ε_3	0	0	0	1	0	0	0	0
ε_4	0	0	0	0	$0^\star$	0	0	0
ε_5	0	0	0	0	0	$0^\star$	0	0
ε_6	0	0	0	0	0	0	$0^\star$	0
ε_7	0	0	0	0	0	0	0	$0^\star$

TAB. 3.3 –: *Comportement de l'opérateur $\delta_{sim}(o_{a_i},o_{b_i})$ pour des valeurs particulières*

o_a o_b	λ	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ε_7
λ	0	1	1	$0^\star$	$0^\star$	1	$0^\star$	1
ε_1	1	0	1	$0^\star$	$0^\star$	1	$0^\star$	1
ε_2	1	1	0	1	$0^\star$	1	$0^\star$	$0^\star$
ε_3	$0^\star$	$0^\star$	1	0	$0^\star$	1	$0^\star$	1
ε_4	$0^\star$	$0^\star$	$0^\star$	$0^\star$	0	1	$0^\star$	$0^\star$
ε_5	1	1	1	1	1	$1^\star$	$0^\star$	1
ε_6	$0^\star$	$0^\star$	$0^\star$	$0^\star$	$0^\star$	$0^\star$	$1^\star$	$0^\star$
ε_7	1	$0^\star$	1	1	$0^\star$	1	$0^\star$	0

TAB. 3.4 –: *Comp. de l'opérateur $\delta_{dissim}(o_{a_i},o_{b_i})$ pour des valeurs particulières*

	V_E	V_1	V_2	V_3	V_4	O_{app}	O_{cont}	O_{cont_1}	O_{cont_2}	V_{cont_1}	V_{cont_2}
1	A	A	A	A	A	x	x	x	-	ε_2	ε_7
2	A	A	A	B	A	x	-	-	-	ε_4	ε_4
3	A	A	C	B	B	-	-	-	-	ε_4	ε_4
4	A	A	A	A	A	-	-	-	-	ε_4	ε_4
5	A	A	C	A	B	x	x	x	-	ε_2	ε_7
6	A	A	C	B	B	-	-	-	-	ε_4	ε_4
7	A	B	B	A	A	x	x	x	-	ε_2	ε_7
8	A	B	A	A	A	-	-	-	-	ε_4	ε_4
9	A	B	C	A	A	x	x	x	-	ε_2	ε_7
10	A	B	B	A	A	x	-	-	-	ε_4	ε_4
11	B	A	B	A	B	x	x	-	x	ε_7	ε_2
12	B	A	B	B	A	-	-	-	-	ε_4	ε_4
13	B	A	B	A	B	-	-	-	-	ε_4	ε_4
14	B	B	C	B	A	x	x	-	x	ε_7	ε_2
15	B	B	C	B	B	-	-	-	-	ε_4	ε_4
16	B	B	C	B	B	-	-	-	-	ε_4	ε_4
17	B	B	A	B	A	-	-	-	-	ε_4	ε_4
18	B	B	B	A	B	x	x	-	x	ε_7	ε_2
19	B	B	B	A	B	-	-	-	-	ε_4	ε_4
20	B	B	B	A	B	x	x	-	x	ε_7	ε_2

TAB. 3.5 –: *Jeu de données synthétique*

ces attentes imposer des contraintes que doit respecter la cns résultat constitue une bonne solution. Toutefois, nombre des méthodes existantes s'adaptent mal à l'adjonction de contraintes. Nous montrons que la méthode que nous proposons peut, elle, utiliser la notion de contraintes sans grande difficulté.

En fait, nous allons montrer qu'une contrainte peut être modélisée sous la forme d'une variable dont on pondère l'influence sur la valeur de l'aspect naturel d'une partition. Nous commençons par présenter un exemple illustrant comment nous pourrons introduire des contraintes dans le processus de clustering, puis formalisons cette notion de contrainte.

Illustration Considérons le jeu de données du tableau 3.5, (ce jeu de données est constitué de 20 objets, de 4 variables exogènes (V_1,V_2,V_3,V_4) et d'une variable endogène (V_E). Admettons que l'on veuille réaliser une cns et que nous désirions que cette cns présente la particularité de ne pas réunir d'objets de $O_{cont1} = \{o_1,o_5,o_7,o_9\}$ avec des objets de $O_{cont2} = \{o_{11},o_{14},o_{18},o_{20}\}$ (voir tableau 3.5). Pour cela, nous allons introduire une contrainte, et ce, sous la forme d'une nouvelle variable V_{cont_1} (voir tableau 3.5) à laquelle on associera une pondération p_1 afin de mettre en adéquation son effet sur la valeur de l'aspect naturel d'une partition et son objectif de séparation des objets de O_{cont1}

et de ceux de O_{cont2}. Cette variable possède 3 modalités différentes ε_4, ε_2, ε_7, on assigne aux objets de O_{cont1} la valeur de modalité ε_2, on assigne aux objets de O_{cont2} la valeur de modalité ε_7, on assigne aux objets restants la valeur de modalité ε_4 (voir tableau 3.5). Ainsi, lors du calcul du critère NCC :

- les partitions présentant des objets de O_{cont1} et de O_{cont2} au sein d'une même classe seront pénalisées du point de vue de cette nouvelle variable (par exemple, si o_1 et o_{11} sont réunis au sein d'une même classe, l'homogénéité interne de la classe est pénalisée puisque :
  $1 - \delta_{dissim}(o1_{V_{cont_1}}, o11_{V_{cont_1}}) = 1)$;
- par contre mélanger des objets de O_{cont1} avec des objets n'appartenant ni à O_{cont1} ni à O_{cont2} ne sera pas pénalisant , enfin séparer les objets de O_{cont1} (resp. O_{cont2}) (resp. les objets restants) n'est absolument pas pénalisant du point de vue de cette variable (par exemple si o_1 et o_5, appartiennent à des classes différentes l'hétérogénéité entre ces deux classes n'est pas pénalisée puisque :
  $\delta_{sim}(o1_{V_{cont_1}}, o5_{V_{cont_1}}) = \delta_{sim}(\varepsilon_2, \varepsilon_2) = 0)$
  (resp. si o_{11} et o_{14} appartiennent à des classes différentes l'hétérogénéité entre ces deux classes n'est pas pénalisée puisque :
  $\delta_{sim}(o11_{V_{cont_1}}, o14_{V_{cont_1}}) = \delta_{sim}(\varepsilon_7, \varepsilon_7) = 0)$
  (resp. si o_2 et o_{12} appartiennent à des classes différentes l'hétérogénéité entre ces deux classes n'est pas pénalisée puisque :
  $\delta_{sim}(o2_{V_{cont_1}}, o12_{V_{cont_1}}) = \delta_{sim}(\varepsilon_4, \varepsilon_4) = 0)$).

La pondération p_1 correspond à un facteur multiplicateur de l'effet de la variable sur le critère NCC : son effet sera alors équivalent à celui de p_1 variables identiques à elle. Notons enfin que cette variable ne sera pas utilisée pour les processus de segmentation ayant lieu lors du processus de cns. (nous fixons ici $p_1 = 64$).

Le résultat du processus de cns avec adjonction des contraintes (et $\alpha = 1$) est présenté sur la figure 3.11, le résultat du processus de cns sans contrainte (et $\alpha = 1$) est présenté sur la figure 3.10.

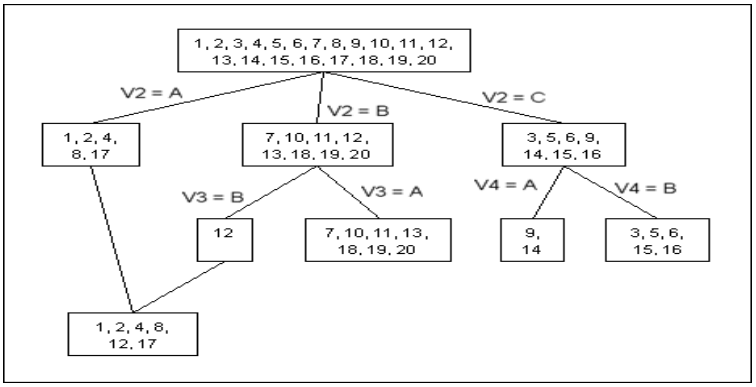


FIG. 3.10 –: *Illustration du processus de cns sans contrainte*

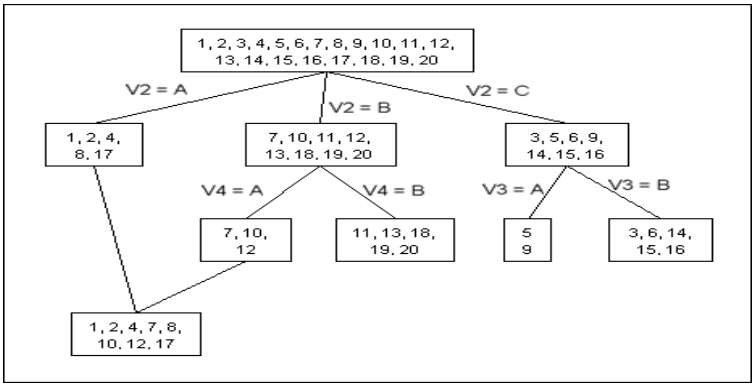


FIG. 3.11 –: *Illustration du processus de cns avec contrainte*

Formalisation Nous formalisons maintenant l'ensemble des éléments qui permettent l'introduction de contraintes dans un processus de cns.

Définition 7 *Contrainte : Une contrainte est modélisée par une variable supplémentaire, qui ne peut servir pour la segmentation de classes lors du processus de segmentation et dont l'influence dans la valeur de l'aspect naturel est pondérée. Ainsi, le calcul de l'aspect naturel d'une partition est le suivant dans le cas de la présence de contraintes.*

Soient

$O = \{o_i, i = 1..n\}$ *l'ensemble des objets du jeu de données*

$EV = \{V_j, j = 1..p\}$ *l'ensemble des variables du jeu de données*

$EV_C = \{Vc_j, j = 1..pc\}$ *l'ensemble des contraintes auxquelles sont associées une pondération* p_j *pour chacune d'entre elles*

$o_i = \{o_{ij}, j = 1,..,p,p+1,...,p+pc\}$ *un objet de O (les p premières valeurs d'attributs correspondant aux variables les pc suivantes aux contraintes)*

C_k *un ensemble d'objets de O*

$P_h = \{C_1,...,C_z\}$ *une partition de O en z groupes*

$Q(P_h)$ *la mesure de l'aspect naturel d'une partition* P_h

$$Q(P_h) = \sum_{i=1..z, j=1..z, j<i} Sim(C_i, C_j) + \alpha \times \sum_{i=1}^{z} Dissim(C_i) \tag{3.6}$$

α *un scalaire (fixé par l'utilisateur)*

$$Sim(C_i, C_j) = \sum_{o_a \in C_i, o_b \in C_j} sim(o_a, o_b) \tag{3.7}$$

$$Dissim(P_i) = \sum_{o_a \in C_i, o_b \in C_i} dissim(o_a, o_b) \tag{3.8}$$

$$sim(o_a,o_b) = \sum_{i=1}^{p} \delta_{sim}(o_{a_i},o_{b_i}) + \sum_{i=p+1}^{p+pc} p_i \delta_{sim}(o_{a_i},o_{b_i}) \tag{3.9}$$

$$dissim(o_a,o_b) = \sum_{i=1}^{p} \delta_{dissim}(o_{a_i},o_{b_i}) + \sum_{i=p+1}^{p+pc} p_i(\delta_{dissim}(o_{a_i},o_{b_i})) \tag{3.10}$$

$$\delta_{sim}(o_{a_i},o_{b_i}) = \begin{cases} 1 \textit{ si } o_{a_i} = o_{b_i} \\ 0 \textit{ si } o_{a_i} \neq o_{b_i} \\ \textit{cas particuliers si } o_{a_i} \in E_\varepsilon \textit{ et/ou } o_{b_i} \in E_\varepsilon \end{cases} \tag{3.11}$$

$$\delta_{dissim}(o_{a_i},o_{b_i}) = \begin{cases} 0 \textit{ si } o_{a_i} = o_{b_i} \\ 1 \textit{ si } o_{a_i} \neq o_{b_i} \\ \textit{cas particuliers si } o_{a_i} \in E_\varepsilon \textit{ et/ou } o_{b_i} \in E_\varepsilon \end{cases} \tag{3.12}$$

Définition 8 *Pondération d'une contrainte : lors de l'établissement d'une contrainte on peut désirer que celle ci soit toujours respectée, parfois respectée, ou si possible respectée ; en clair, on peut définir une sorte de niveau d'exigence du respect d'une contrainte. Ce niveau d'exigence du respect d'une contrainte sera modélisé par ce que nous appelons pondération d'une contrainte. Cette pondération est un scalaire* $p \in [0,+\propto [$*. Plus le scalaire est grand plus l'exigence de respect de la contrainte est forte.*

Exemples de Contraintes Nous donnons ici quelques exemples de type de contraintes ainsi que leur modélisation :

- *Contraintes du type "Aucun objet de l'ensemble* O_1 *ne doit se retrouver dans la même classe qu'un objet de l'ensemble* O_2 *"*. Ces contraintes se modélisent de la manière suivantes : soit O l'ensemble des objets du jeu de données $O = O_1 \bigcup O_2 \bigcup O_r$ ($O_r = \{o_i \in O, o_i \notin O_1, o_i \notin O_2\}$), on ajoute au jeu de données une variable contrainte V_c associée à une pondération p_c telle que :
  - les objets de O_1 aient ε_2 pour valeur pour V_c,
  - que les objets de O_2 aient ε_7 pour valeur pour V_c
  - les objets de O_r aient ε_4 pour valeur pour V_c.
- *Contraintes du type "Les objets de l'ensemble* O_1 *doivent se retrouver dans la même classe que les objets de l'ensemble* O_2 *"*. Ces contraintes se modélisent de la manière suivantes :soit O l'ensemble des objets du jeu de données $O = O_1 \bigcup O_2 \bigcup O_r$, on ajoute au jeu de données une variable contrainte V_c associée à une pondération p_c telle que :
  - les objets de O_1 et O_2 aient une valeur classique commune pour V_c,
  - les objets de O_r aient ε_4 pour valeur pour V_c.

3.2.4.4 De l'Apprentissage Non Supervisé à l'Apprentissage Supervisé : l'Apprentissage Non Supervisé sous Contraintes

L'intégration de contraintes au processus d'apprentissage non supervisé peut mener à l'établissement d'une méthode d'apprentissage supervisé ou plutôt semi-supervisé [CCM00] [DBE99] dont l'intérêt est notamment de pouvoir permettre la réalisation d'apprentissage supervisé lorsque l'on dispose uniquement d'un faible nombre d'exemples étiquetés mais également d'assurer dans certains cas une plus grande stabilité au modèle d'apprentissage bâti [CCM00].

Nous exposons rapidement comment la méthode d'apprentissage non supervisé que nous venons de proposer peut être transformée en une méthode d'apprentissage semi-supervisé relativement efficace[7].

L'idée sous jacente est simple : "pourquoi ne pas réaliser un apprentissage non supervisé par l'intermédiaire de notre méthode en lui adjoignant un certain nombre de contraintes basées sur la connaissance de l'étiquette d'un certain nombre d'individus". Le processus d'apprentissage ainsi réalisé mènera à la découverte d'une partition (et de la structure du graphe d'induction ayant mené à cette partition) telle qu'elle respecte les contraintes et qu'elle soit la plus "naturelle" possible.

Ainsi, par réalisation d'un apprentissage non supervisé sous contraintes nous obtenons une partition pour laquelle chaque classe est caractérisée par une règle logique, ces règles permettent donc d'associer à tout individu une classe de la partition. De plus, pour chaque classe de la partition on peut déterminer la modalité (de la variable endogène) la plus représentée parmi les individus dont on connaît l'étiquette et ainsi l'associer à la classe. On dispose alors de règles permettant non seulement d'associer à tout individu une classe de la partition découverte mais aussi une modalité de la variable endogène, cela correspond bien à un modèle d'apprentissage supervisé.

Notons que nous avons utilisé ici uniquement des contraintes visant à séparer des individus.

L'algorithme que nous proposons est présenté en page 51. Nous ne détaillons pas plus en détail la description formelle de cette méthode qui s'apparente aux graphes d'induction, toutefois nous proposons un exemple illustratif sur le jeu de données du tableau 3.5.

7. notons qu'il ne s'agit ici que d'un prototype de méthode d'apprentissage...

Algorithme 2 Algorithme d'Apprentissage Semi-Supervisé

Données :
l'ensemble des objets : $O = \{o_1,..,o_n\}$;
les variables exogènes $EV = \{V_i, i = 1..p\}$
la variable endogène V_E (qui possède k modalités ($v_{E1},..,v_{Ek}$))

1. Séparer l'ensemble des objets O en 2 échantillons : l'échantillon d'apprentissage O_{app} et l'échantillon de validation O_{val} ($O = O_{app} \bigcup O_{val}$).

2. Fixer le pourcentage d'individus ($pourc$) de l'échantillon d'apprentissage sur lesquels on impose des contraintes.

3. Séparer l'échantillon d'apprentissage O_{app} en 2 ensembles, l'un (O_{cont}) comprenant les individus sur lesquels des contraintes sont imposées ($card(O_{cont}) = pourc \times card(O_{app})$) et l'autre ($O_{rest}$) les objets restants ($O_{app} = O_{cont} \bigcup O_{rest}$).

4. Séparer l'échantillon O_{cont} en autant d'échantillons qu'il existe de modalités pour la variable endogène (c'est à dire séparer O_{cont} en k échantillons, chacun de ces échantillons est noté O_{cont_i} et est pur du point de vue de la variable endogène (il ne contient que des objets ayant la modalité v_{Ei} pour la variable endogène)) ($O_{cont} = \bigcup_{i=1..k} O_{cont_i}$).

5. Créer les contraintes telles qu'elles imposent aux individus de chacun des échantillons O_{cont_i} à ne pas être regroupés au sein d'une même classe qu'un individu appartenant à l'un des échantillons O_{cont_j}, $j = 1..k$, $j \neq i$. Cela mène à la création de k contraintes : $V_{cont_1},..,V_{cont_k}$ dont les pondération $p_1,..,p_k$ doivent être très forte, $\forall i = 1..k,\ p_i \rightarrow \infty$ (nous fixons $\forall i = 1..k\ p_i = p^3$). Chaque contrainte possède 3 modalités : ε_2, ε_4, ε_7, on assigne ces valeurs de modalités aux objets de O selon les règles suivantes : la valeur prise par l'objet o_a pour la variable V_{cont_i} est : ε_4 si $o_a \notin O_{cont}$; ε_2 si $o_a \in O_{cont_i}$; ε_7 si $o_a \in O_{cont_j}$, $j \in 1,..,k$, $j \neq i$

6. Lancer l'algorithme d'apprentissage non supervisé KEROUAC (avec $\alpha = 1$) sur le jeu de données considéré dans son intégralité (i.e. l'ensemble des objets considérés est bien O et non pas O_{app} en lui ayant au préalable intégré les contraintes (i.e. l'ensemble des variables considérées est constitué de V et des k contraintes modélisées sous la forme des variables $V_{cont_1},..,V_{cont_k}$; nous rappelons toutefois qu'aucun processus de segmentation ne peut être obtenue sur la base de ces dernières variables).

7. Obtention d'une partition $P_h = \{C_i, i = 1..z\}$ et de la structure du graphe d'induction lui ayant donné le jour.

8. Pour chaque classe de P_h on peut associer une modalité de la variable endogène, cette modalité étant celle la plus représentée parmi les objets de la classe appartenant à l'échantillon d'apprentissage, si toutefois aucun objet de l'échantillon d'apprentissage n'est présent dans la classe alors aucune modalité de la variable endogène n'est associé à la classe.

9. Obtention d'un ensemble de règles permettant éventuellement d'associer à un objet une modalité de la variable endogène. Si une règle ne permet pas d'associer une modalité de la variable endogène à un individu, cet individu est alors dit indéterminé.

EXEMPLE : *Considérons donc le jeu de données du tableau 3.5. Admettons que l'on veuille réaliser un apprentissage sur 50% des objets et que l'échantillon d'apprentissage (resp. de validation) soit composé de la manière suivante :*
$O_{app} = \{o_1,o_2,o_5,o_7,o_9,o_{10},o_{11},o_{14},o_{18},o_{20}\}$
(resp. $O_{val} = \{o_3,o_4,o_6,o_8,o_{12},o_{13},o_{15},o_{16},o_{17},o_{19}\}$*).*

Admettons maintenant que l'on fixe le paramètre $pourc$ *à 80%, et que* O_{cont} *(resp.* O_{rest}*) soit composé comme suit* $O_{cont} = \{o_1,o_5,o_7,o_9,o_{11},o_{14},o_{18},o_{20}\}$ *(resp.* $O_{rest} = \{o_2,o_{10}\}$*). On obtient donc* $O_{cont1} = \{o_1,o_5,o_7,o_9\}$, $O_{cont2} = \{o_{11},o_{14},o_{18},o_{20}\}$*. Cela permet de générer les variables* V_{cont_1} *et* V_{cont_2} *qui modélisent les contraintes impliquant que les objets de* O_{cont1} *et ceux de* O_{cont2} *ne doivent pas être réunis au sein d'une même classe. Notons que les pondérations (*p_1*,* p_2*) de ces deux variables doivent être très fortes (*$p_1 \to \infty$*,* $p_2 \to \infty$*). Si l'on lance l'algorithme* KEROUAC *(avec* $\alpha = 1$*) on obtient alors la partition suivante*
$\{\{o_1,o_2,o_4,o_7,o_8,o_{10},o_{12},o_{17}\}, \{o_5,o_9\}, \{o_3,o_6,o_{14},o_{15},o_{16}\}, \{o_{11},o_{13},o_{18},o_{19},o_{20}\}\}$*, le graphe d'induction de la figure 3.11, ainsi que les règles logiques suivantes* :

- $[V_2 = A] OU [V_2 = B\ ET\ V_4 = A] \Rightarrow Classe1$
- $[V_2 = C\ ET\ V_3 = A] \Rightarrow Classe2$
- $[V_2 = C\ ET\ V_3 = B] \Rightarrow Classe3$
- $[V_2 = B\ ET\ V_4 = B] \Rightarrow Classe4$

Par comptabilisation dans chaque classe des valeurs pour la variable endogène V_E *des objets de* O_{app} *appartenant à la classe, on peut associer la modalité la plus représentée à la classe et ainsi obtenir les règles logiques suivantes :*

- $[V_2 = A] OU [V_2 = B\ ET\ V_4 = A] \Rightarrow V_E = A$
- $[V_2 = C\ ET\ V_3 = A] \Rightarrow V_E = A$
- $[V_2 = C\ ET\ V_3 = B] \Rightarrow V_E = B$
- $[V_2 = B\ ET\ V_4 = B] \Rightarrow V_E = B$

On peut par la suite utiliser ces règles pour assigner une modalité de la variable endogène aux divers objets du jeu de données ce qui implique les associations suivantes : $o_1 \to V_E = A$, $o_2 \to V_E = A$, $o_3 \to V_E = B$, $o_4 \to V_E = A$, $o_5 \to V_E = A$, $o_6 \to V_E = B$, $o_7 \to V_E = A$, $o_8 \to V_E = A$, $o_9 \to V_E = A$, $o_{10} \to V_E = A$, $o_{11} \to V_E = B$, $o_{12} \to V_E = A$, $o_{13} \to V_E = B$, $o_{14} \to V_E = B$, $o_{15} \to V_E = B$, $o_{16} \to V_E = B$, $o_{17} \to V_E = A$, $o_{18} \to V_E = B$, $o_{19} \to V_E = B$, $o_{20} \to V_E = B$, .

Ici le taux de correction sur l'échantillon d'apprentissage (resp. de validation) est donc de 100% (resp. 60%).

Evaluation Expérimentale L'évaluation expérimentale a été réalisée sur 14 jeux de données (issus de la collection de l'université de Californie à Irvine [MM96], voir page 217 pour plus d'informations sur ces jeux de données) sur lesquels ont été menés divers apprentissages mettant en œuvre 6 méthodes d'apprentissage différentes : ID3, C4.5, Sipina et notre méthode pour les méthodes de type arbres/graphes d'induction, 1-plus proche voisins et bayésiens naïfs pour les autres méthodes d'apprentissage. Ces divers apprentissages ont

permis la réalisation d'une étude comparative concernant le taux de correction selon la méthode employée. L'évaluation du taux de correction est réalisée pour une 10-cross-validation ainsi que pour cinq 2-cross-validations. Pour ces évaluations, les paramètres suivants ont été adoptés : $pourc = 90\%$, $\alpha = 1$.

	ID3	C4.5	SIPINA	B.Naïfs	1-PPV	KEROUAC
GERMAN	$31.2^{3.37}$	$29^{4.1}$	$29.9^{4.5}$	$24^{3.6}$	$31.6^{6.07}$	$33.7^{3.74}$
MUSH.	$0.07^{0.13}$	0^{0}	$0.62^{0.17}$	$0.31^{0.19}$	0^{0}	0^{0}
SICK	$2.36^{1.11}$	$2.14^{0.81}$	$2.64^{0.99}$	$2.82^{1.45}$	$2.79^{0.91}$	0^{0}
VEHICLE	$33.7^{4.64}$	$32.4^{4.67}$	$43.51^{1.81}$	$33.32^{4.56}$	$33.21^{2.73}$	$35.11^{5.5}$
MONKS 3	0^{0}	0^{0}	0^{0}	$2.79^{2.71}$	$13.66^{4.68}$	0^{0}
FLAGS	$30.89^{7.23}$	$34.03^{10.6}$	$46.37^{11.4}$	$34.73^{11.8}$	$46.95^{10.2}$	$35.13^{9.55}$
BREAST	$9.3^{2.23}$	$5.43^{2.28}$	$6.87^{3.07}$	$3^{2.16}$	$5.58^{3.1}$	$4.44^{2.25}$
ZOO	$26.64^{10.62}$	$7^{6.4}$	$13.82^{7.88}$	$14^{11.14}$	$3.91^{6.56}$	$3.91^{4.58}$
WINE	$8.46^{3.83}$	$6.14^{6.31}$	$7.22^{7.88}$	$3.4^{4.56}$	$4.48^{4.17}$	$2.23^{2.72}$
CANCER	$7.47^{2.73}$	$5.27^{2.64}$	$4.68^{3.44}$	$2.49^{2.08}$	$5.42^{2.54}$	$5.42^{2.62}$
PIMA	$25.26^{3.22}$	$26.3^{5.07}$	$26.05^{5.41}$	$22.26^{4.13}$	$31.65^{4.64}$	$33.95^{2.68}$
CONTRA.	$52^{3.83}$	$50.71^{3.6}$	$56.29^{5.52}$	$50.16^{4.49}$	$56.82^{4.49}$	$43.74^{2.79}$
ION	$10.83^{6.98}$	$8.55^{3.13}$	$12.26^{5.14}$	$5.42^{2.99}$	$13.97^{5.35}$	$11.38^{4.89}$
HVOTES	$4.37^{3.62}$	$6.22^{3.88}$	$4.38^{2.18}$	$10.34^{4.15}$	$13.76^{4.37}$	$6.21^{3.34}$

Légende : Taux d'erreur Moyen $^{Ecart\ Type\ pour\ le\ taux\ d'erreur}$

TAB. 3.6 –: *Taux d'Erreur Moyen en Validation pour une 10-Cross-Validation*

Dans la mesure où la méthode présentée ici n'est en défintive qu'un prototype et que, nous le verrons, un certains nombre de questions restent à aborder nous ne détaillons pas ici les résultats de ces évaluations. Cependant, il apparaît clairement que les modèles d'apprentissage bâtis par le biais de cette méthode présentent une qualité très intéressante puisqu'elle est relativement similaire à celle des méthodes de références utilisées...

Ces résultats tendent à étayer la proposition de création de modèles d'apprentissage supervisé par des techniques d'apprentissage semi-supervisé et, plus précisément, dans le cas présent, de techniques d'apprentissage non supervisé sous contraintes.

Notons que plusieurs points mériteraient d'être approfondis tels que le type et le nombre de contraintes à intégrer. Ainsi, si les contraintes employées ici visent à séparer un certain nombre d'objets sélectionnés aléatoirement, on pourrait envisager une étude poussée concernant le nombre d'objets à assujetir à des contraintes ainsi qu'une étude concernant une sélection "intelligente" et non aléatoire de ces mêmes objets. De même, l'intégration d'autres types de contraintes doit être vecteur d'amélioration de qualité des modèles élaborés dans certaines situations.

Nous pensons donc que ces différents points soulignent plus encore l'intérêt de l'approche apprentissage non supervisé sous contraintes en démontrant

	ID3	C4.5	SIPINA	B.Naïfs	1-PPV	KEROUAC
GERMAN	$30.54^{0.47}$	$28.92^{1.09}$	$29.86^{0.71}$	$24.44^{0.74}$	$32.92^{1.49}$	$32.28^{1.69}$
MUSH.	$0.26^{0.05}$	$0.02^{0.03}$	$0.48^{0.16}$	$0.32^{0.02}$	0^{0}	0^{0}
SICK	$3.19^{0.17}$	$2.48^{0.18}$	$2.41^{0.04}$	$3.28^{0.24}$	$3.09^{0.43}$	$2.98^{0.34}$
VEHICLE	$37.64^{0.62}$	$34.18^{1.97}$	$46.48^{0.96}$	$34.54^{1.01}$	$36.41^{0.91}$	$35.62^{1.85}$
MONKS 3	$5^{0.06}$	0^{0}	$3.19^{1.13}$	2.78^{0}	$12.13^{1.45}$	0^{0}
FLAGS	$52.99^{1.71}$	$38.04^{4.81}$	$54.85^{1.06}$	$44.43^{2.77}$	$48.14^{3.29}$	$38.45^{4.66}$
BREAST	$8.56^{0.48}$	$5.41^{0.7}$	$6.32^{0.6}$	$3^{0.37}$	$5.58^{0.6}$	$5.81^{1.31}$
ZOO	$19^{1.47}$	$13.85^{2.44}$	$18.23^{0.8}$	$25.7^{4.39}$	$7.13^{1.6}$	$12.68^{7.19}$
WINE	$14.49^{1.15}$	$8.43^{2.59}$	$18.76^{2.09}$	$9.55^{1.88}$	$8.88^{1.96}$	$7.8^{3.3}$
CANCER	$8.37^{0.55}$	$6.76^{0.3}$	$6.53^{0.73}$	$2.9^{0.32}$	$5.04^{0.48}$	$5.27^{0.81}$
PIMA	$26.43^{1.44}$	$25.96^{0.92}$	$26.12^{1.09}$	$22.5^{0.1}$	$31.87^{1.23}$	$34.09^{2.85}$
CONTRA.	$53.22^{0.39}$	$51.54^{1.39}$	$58.38^{0.62}$	$49.75^{0.87}$	$59.02^{0.34}$	$55.12^{2.71}$
ION	$18.52^{1.39}$	$9.57^{1.07}$	$12.25^{1.62}$	$9.91^{1.78}$	$13.62^{0.75}$	$19.25^{3.84}$
HVOTES	$4.64^{0.55}$	$5.38^{0.63}$	$7.22^{3.54}$	$10.12^{1.21}$	$13.89^{1.12}$	$6.39^{1.24}$

Légende : Taux d'erreur Moyen $^{Ecart\ Type\ pour\ le\ taux\ d'erreur}$

TAB. 3.7 –: *Taux d'Erreur Moyen en Validation pour cinq 2-Cross-Validation*

qu'il est possible d'accroître la qualité des modèles ainsi que de créer des modèles performants dans des situations où l'on dispose de nombre d'objets non étiquetés.

3.3 Conclusion

L'évaluation de notre méthode de cns a permis de mettre en avant la qualité des cns produites, la très bonne stabilité et le coût calculatoire relativement faible de la méthode. De plus, les différentes comparaisons avec la méthode de référence que constitue les K-Modes montrent tout l'intérêt de KEROUAC [8].

Nous utilisons maintenant "la grille de lecture" utilisée plus tôt pour présenter les méthodes de cns classiques afin de résumer les caractéristiques principales de notre méthode :

- **Complexité :** identique à celle des graphes d'induction (log-linéaire selon le nombre d'objets et linéaire selon le nombre de variables), scalabilté apparemment bonne,
- **Géométrie des Classes :** pas d'a priori particulier pour la forme des classes [9],

8. Cependant, bien que les efforts en terme d'expérimentations aient été importants, mener un ensemble plus vaste encore d'expériences pourrait permettre d'obtenir un panorama plus vaste de résultats pour comparer ces deux méthodes...

9. Aucune évaluation précise menée sur ce point, mais dans la mesure où l'on utilise une mesure proche de celle employée dans RDA/AREVOMS on peut penser que le comportement de notre méthode est identique sur ce point à celui de RDA/AREVOMS (et ce bien que la méthode d'optimisation sous-jacente à notre méthode soit différente à celle de RDA/AREVOMS).

- **Gestion des Outliers :** oui[10]
- **Paramètres :** facteur de granularité
- **Résultats :** composition des classes, mode de chacune des classes et graphe permettant d'associer une règle caractérisant chacune des classes,
- **Critère :** $NCC^{\star}$.

Enfin, nous évaluons maintenant notre méthode au regard des challenges actuels en cns :

- **Problèmes inhérents aux données traitées :**
  - **Très grand nombre d'objets.** Complexité algorithmique théorique relativement faible et exhibe une bonne scalabilité.
  - **Dimensionnalité élevée.** La complexité de notre méthode est linéaire selon le nombre de variables ce qui de prîme abord semble très correct. Cependant KEROUAC est également sensible aux nombres de modalités des variables catégorielles traitées, ainsi si les variables possèdent de nombreuses modalités le coût calculatoire de notre méthode peut s'accroître sensiblement. Cela constitue selon nous le principal problème lié à notre méthode.
  - **Autres éléments problématiques.** La présence de *données manquantes* n'est pas gênante car leur traitement est aisément et rapidement paramétrable comme cela a été montré précédemment. La présence d'*outliers* n'apparaît pas vraiment problématique dans la mesure où notre méthode est relativement insensible à leur présence puisqu'elle peut générer des classes possédant peu d'objets voire un unique objet.
- **Problèmes inhérents à des contraintes applicatives :**
  - **Nécessité d'intégrer des connaissances, des contraintes dans le processus.** L'intégration de connaissances ou de contraintes est aisée comme cela est montré dans la section précédente.
  - **Bonne utilisabilité.** L'utilisabilité de notre méthode semble très bonne : le paramétrage de l'algorithme est facile, intelligible et intuitif, la présentation des résultats et des connaissances extraites par le processus de cns est explicite et intelligible.
  - **Données distribuées.** En soi, KEROUAC ne permet pas vraiment un traitement direct de données distribuées, cependant KEROUAC peut constituer une méthode d'agrégation de cns distribuées ce qui fait l'objet du chapitre 6.

Enfin, notons que cette méthode peut être utilisée comme base pour la mise au point de méthode d'apprentissage semi-supervisé (comme le montre la section précédente) ou encore de méthode visant à la complétion de données manquantes, d'apprentissage généralisé, ou de "profilage" (voir [JN03f])...

4 Validité en Apprentissage Non Supervisé

"The popularized definition of KDD postulates it as "The non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". We are unsure if valid is listed among the first characteristics in proportion to its importance, but certainly, patterns in data will be far from useful if they were invalid. A more cynical view would say that trivial processes can certainly deliver invalid, understandable and novel patterns."

- Vladimir Estevill-Castro -
"Why so many clustering algorithms - A Position Paper", SIGKDD Explorations, vol.4, issue 1, pages 65-75 (2002)

L'ECD est définie dans [FPSS96] comme le processus non trivial d'identification de connaissances valides, nouvelles, potentiellement utiles et intelligibles au sein d'un ensemble de données[1]. Estevill-Castro remarque dans [EC02] que, si l'on ne peut pas forcément expliquer la présence du terme valide en tête de définition par une connotation à l'importance majeure de la validité, il est par contre certain que la découverte de connaissances non valides est d'un intérêt nul. L'évaluation de la validité de la connaissance extraite (ou plutôt de l'éventuelle connaissance extraite puisqu'une connaissance non valide ne peut être considérée comme de la connaissance) constitue donc une étape clé du processus ECD que nous abordons ici dans le cadre restreint de la cns.

L'aspect fondamental de la classification non supervisée dans le processus d'ECD a mené ces dernières années à de multiples efforts de recherche dans ce domaine et plus particulièrement au développement de nouveaux algorithmes au coût calculatoire faible ainsi qu'à la mise au point de critères adaptés au traitement de type de données spécifiques. Il résulte de ces études le besoin de posséder des outils pour l'évaluation et la comparaison de la validité de résultats du processus de classification non supervisée. En effet, outre leur utilité pour la confirmation de la validité de résultats, ces outils peuvent également

1. Il s'agit ici de la définition anglo-saxone, la définition francophone insistant également sur la nature itérative et le caractère interactif de ce processus.

assister les utilisateurs dans le choix d'une classification parmi un ensemble de classifications, et ce, indépendamment de la méthode utilisée, des paramètres de la méthode et du nombre de classes. De tels outils pourraient encore permettre la comparaison objective de méthodes et ainsi conduire finalement à la définition d'un cadre pour la comparaison des méthodes de cns. [2]

Bien que nous regrettions que cette problématique ait été, selon nous, et, eu égard à son importance, l'objet de trop peu de travaux, nous pouvons cependant citer plusieurs travaux de recherche plus ou moins récente [BEF84], [Dav96], [Dom01], [Hal00a], [HBV01], [HV01], [HBV02b], [HBV02a], [RLR98b], [Sha96], [TK99], [XB91]... Notons cependant que cette problématique connaît un regain d'intérêt fort comme le démontre notamment les publications récentes de Maria Halkidi et Michalis Vazirgiannis ([Hal00a], [HBV01], [HV01], [HBV02b], [HBV02a]) ainsi que le tutoriel sur l'estimation de la qualité en fouille de données [3] qu'ils ont animé lors des conférences ECML/PKDD02 : *"An Introduction to Quality Assessment in Data Mining"* et leur très récent livre consacré intégralement à cette problématique [VHD03].

Dans un premier temps, nous nous référons à ces travaux pour introduire les critères existants pour l'évaluation de la validité de cns, puis proposons et expérimentons deux nouveaux critères associés à une méthodologie qui leur est propre pour l'évaluation de la validité de cns.

4.1 Validité d'une Classification Non Supervisée : Définition et Evaluation

L'objectif principal de la cns est la découverte de l'organisation (structuration) d'un ensemble d'objets selon des classes naturelles afin d'identifier des similarités et différences entre classes ainsi qu'inférer des spécificités intéressantes pour chaque classe. Or, la nature non supervisée de ce processus n'autorise pas une définition claire et directe de ce que sont des structures/organisations valides. Ainsi, les multiples algorithmes de cns se caractérisent par l'ensemble d'hypothèses qu'ils emploient afin de définir les propriétés devant être satisfaites par une structure valide. L'ensemble d'hypothèses déterminant la validité d'une structure n'étant pas universel et différant selon les méthodes, les résultats varient donc selon la méthode utilisée. Conséquemment, il est essentiel de définir une méthode d'évaluation des structures résultant d'un processus de cns. Ce type d'évaluation est nommé évaluation de la validité d'une cns.

On peut considérer qu'il existe, de manière générale, trois modes d'évaluation de la validité de cns. Le premier est basé sur des critères dits externes, qui

2. Sans l'utilisation de ce type d'outils l'évaluation de résultats provenant de cns ainsi que la comparaison de tels résultats peut être difficile et hasardeux.

3. La validité constitue une des composantes les plus importantes pour l'estimation de la qualité en fouille de données.

impliquent l'évaluation de cns au moyen d'une structure définie a priori, cette structure traduisant les connaissances et intuitions de l'utilisateur sur la structure des données. Le second mode est lui fondé sur des critères dits internes, ces derniers impliquent quant à eux une évaluation sur l'unique base des données à traiter et ne font aucunement intervenir des informations exogènes. Le dernier mode est dit relatif ; l'idée clé est ici d'évaluer une cns en se référant à d'autres cns obtenues par l'intermédiaire de la même méthode mais avec des paramétrages différents. Notons enfin que, majoritairement, les critères évalués ont un rapport avec l'homogénéité des classes ou avec la séparation entre classes, plus rarement, ces deux notions sont intégrées simultanément au sein d'un critère.

4.1.1 Mode d'Evaluation par Critères Externes

Le mode d'évaluation impliquant un critère externe est utilisé implicitement dans la plupart des publications traitant de l'évaluation expérimentale de méthodes de cns. Ces critères de validité évaluent dans quelle mesure une cns correspond à des connaissances et intuitions établies a priori sur les données. On admet généralement que ces informations ne peuvent être directement calculées à partir des données initiales. La forme la plus classique d'informations externes est un ensemble de classes et d'étiquettes associées à chacun des objets[4].

L'idée clé est donc de tester si l'ensemble d'objets est structuré de manière aléatoire ou non, et ce, en se référant à une structure pré-définie. Cette analyse se base alors sur l'hypothèse nulle H_0 d'une structure aléatoire. Pour tester cette hypothèse, les tests statistiques peuvent être utilisés, cependant, ils peuvent mener à des procédures calculatoirement coûteuses. Aussi, la méthode de Monte Carlo est parfois utilisée pour résoudre ce type de problèmes.

4.1.1.1 Méthode de Monte Carlo

Cette méthode est utilisée afin de calculer la fonction de densité de probabilité d'un indice statistique par l'intermédiaire de la simulation [TK99] : on procède alors tout d'abord par génération aléatoire d'un grand nombre de partitions des objets du jeu de données considéré (ces partitions correspondent pour ce jeu de données à des cns potentielles) ; puis pour chacune de ces partitions, on calcule la valeur de l'indice dont on recherche la fonction de densité de probabilité ; puis, en utilisant les différentes valeurs obtenues pour l'indice on peut déterminer une approximation de la fonction de densité de probabilité de l'indice. Enfin, les tests statistiques classiques peuvent être employés.

4. Ce type d'informations peut éventuellement être obtenu par une classification manuelle

4.1.1.2 Mesures Statistiques

Considérons $O = \{o_i, i = 1..n\}$ un ensemble d'objets, $P_h = \{C_1,...,C_z\}$ une partition de O en z classes et $P_{spec} = \{C_{spec_1},...,C_{spec_x}\}$ une cns pré-spécifiée (une partition de O en x classes). Par la suite, afin de faire référence à la cardinalité de différents ensembles composés de paires d'objets (o_i,o_j) , nous utilisons les notations suivantes :

- **SS** la cardinalité de l'ensemble composé des paires d'objets de O telles que les deux objets appartiennent à la même classe dans chacune des partitions P_h et P_{spec}
- **SD** la cardinalité de l'ensemble composé des paires d'objets de O telles que les deux objets appartiennent à la même classe dans la partition P_h et à des classes différentes dans la partition P_{spec}.
- **DS** la cardinalité de l'ensemble composé des paires d'objets de O telles que les deux objets appartiennent à la même classe dans la partition P_{spec} et à des classes différentes dans la partition P_h
- **DD** la cardinalité de l'ensemble composé des paires d'objets de O telles que les deux objets appartiennent à des classes différentes dans chacune des partitions P_h et P_{spec}.

REMARQUES :

- $M = SS + SD + DS + DD = \frac{n \times (n-1)}{2}$
- *Si l'on considère l'ensemble d'objets O, que l'on effectue une correspondance entre la partition P_{spec} (resp. P_h) et une variable catégorielle et que l'on considère que cette variable catégorielle est l'unique descripteur de chacun de ses objets alors il existe une relation entre les notations **SS**, **SD**, **DS**, **DD** et la valeur du nouveau critère de Condorcet pour la partition P_h (resp. P_{spec}) : $NCC(P_h) =$ **SS** + **DD** (resp. $NCC(P_{spec}) =$ **SS** + **DD**).*

Des mesures statistiques classiques sont alors définies, en employant ces notations, de la manière suivante :

- *Statistique de Rand :* $R = \frac{SS+DD}{M}$
- *Coefficient de Jaccard :* $J = \frac{SS}{SS+SD+DD}$
- *Indice de Folkes et Mallows :* $FM = \sqrt{\frac{SD}{SS+SD} \frac{SS}{DS+DD}}$
- *Statistique* Γ *de Hubert* : $\Gamma = \frac{SS}{M}$
- *Statistique* Γ *Normalisée :* $\overline{\Gamma}$

REMARQUES :

- *Les deux premières statistiques (R et J) prennent des valeurs entre 0 et 1 et sont maximales pour $z = x$ (i.e. lorsque les partitions P_h et P_{spec} ont le même nombre de classes).*

- *Il a été prouvé que, pour les 3 premières mesures (R, J et FM), de fortes valeurs indiques une grande similarité entre P_h et P_{spec} (i.e. plus fortes sont les valeurs, plus similaires sont les deux partitions)*
- *Pour les indices Γ et $\overline{\Gamma}$, on considère les matrices d'adjacence $n \times n$ MC_{P_h} et $MC_{P_{spec}}$ décrivant les relations d'équivalence associées respectivement aux partitions P_h et P_{spec}. Ces matrices sont définies comme suit :*

  $(MC_{P_{spec}})_{i,j} = \begin{cases} 1 \text{ si } \exists k \in \{1,..,x\} \text{ tel que } o_i \in C_{spec_k} \text{ et } o_j \in C_{spec_k} \\ 0 \text{ sinon} \end{cases}$

  $(MC_{P_h})_{i,j} = \begin{cases} 1 \text{ si } \exists k \in \{1,..,z\} \text{ tel que } o_i \in C_k \text{ et } o_j \in C_k \\ 0 \text{ sinon} \end{cases}$

  avec ces notations on a :

$$\Gamma = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} MC_{P_{h_{i,j}}} MC_{P_{spec_{i,j}}}$$

$$\overline{\Gamma} = \frac{\left(\frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (MC_{P_{h_{i,j}}} - \mu_{MC_{P_h}})(MC_{P_{spec_{i,j}}} - \mu_{MC_{P_{spec}}}) \right)}{(\sigma_{MC_{P_h}} \sigma_{MC_{P_{spec}}})}$$

- *Quel que soit l'indice que nous venons de présenter, on doit connaître sa fonction de densité de probabilité sous l'hypothèse H_0 de structuration aléatoire afin de procéder à des tests statistiques visant à évaluer la validité de la cns. Or comme nous l'avons annoncé plus tôt, déterminer cette fonction peut impliquer un coût calculatoire trop important et nécessiter l'adoption de techniques de Monte Carlo.*

Nous pouvons enfin noter l'existence de critères basés sur la théorie de l'information qui ne nécessitent pas l'utilisation de la méthode de Monte-Carlo tels celui proposé par Dom [Dom01], l'information mutuelle normalisée, ou encore la pureté moyenne ou des mesures basées sur l'entropie [BGG+99].

4.1.2 Mode d'Evaluation par Critères Internes

L'idée est ici d'évaluer une cns résultant de l'application d'un algorithme particulier par utilisation d'une mesure ne considérant que l'information comprise dans les données et aucune information additionnelle. L'utilisation de ce type de mesure conduit à la question suivante : "La mesure que j'utilise permet-elle réellement de capturer l'adéquation entre la classification obtenue et ce qui est particulier dans les données et que je souhaite découvrir ? ".

Les critères internes, tels que la somme du carré des erreurs (SSE : sum of squared errors), ont été utilisés de manière extensive car, la cns peut être vue comme un problème d'optimisation de la valeur d'une mesure interne de validité donnée, (par exemple, les k-means optimisent de manière gloutonne le

critère SSE). Nous listons maintenant quelques unes des mesures internes les plus courantes :

- *SSE (inertie intra-classe, homogénéité intra-classe)* il s'agit certainement de la mesure la plus populaire. Elle est définie de la manière suivante : soit une partition (une cns) $P_h = \{C_1,...,C_z\}$, nous notons n_{C_i} le nombre d'objets de la classe C_i et définissons $c_i = \{c_{i_1},...,c_{i_p}\}$ le centroïde de C_i par $c_{i_j} = \frac{1}{n_{C_i}} \sum_{o_a \in C_i} o_{a_j}$. Ainsi, $SSE(P_h) = \sum_{k=1..z} \sum_{o_a \in C_k} ||o_a - c_k||_2^2$.
  On peut étendre cette définition à d'autres mesure de dissimilarité s entre les objets et leurs centroïdes respectifs : $SSE(P_h) = \sum_{k=1..z} \sum_{o_a \in C_k} s(o_a,c_k)$.
- *Nombre d'arêtes coupées* : lorsque la cns est posée comme un problème de partitionnement de graphe, l'objectif est alors de minimiser le nombre d'arêtes coupées.
- *CU (Category Utility)* [GC85][Fis87], ce critère est une fonction de la prédictabilité des valeurs des attributs impliquées dans une cns. La mesure *CU* est définie comme la différence entre le nombre de valeurs d'attributs pouvant être correctement prédits grâce à l'établissement d'une partition des objets d'un jeu de données et le nombre de valeurs d'attributs pouvant être correctement prédits sans une telle connaissance. (Récemment, il a été montré que ce critère est lié aux critères de type SSE pour un type de codage spécifique [Mir01].) La mesure *CU* est donc définie pour maximiser la prédictabilité des attributs pour une cns, ce qui limite son champ d'utilisation à des problèmes de cns possédant une dimensionalité faible (et touchant de préférence des attributs catégoriels). En effet, pour des problèmes à forte dimension, tels que ceux posés en cns sur textes, l'objectif n'est évidemment pas d'être capable de prédire la présence d'un mot dans un document associé à une classe particulière
- *Coefficient de Corrélation CoPhénétique (CPCC)*: dans le cadre de cns hiérarchiques, une matrice, appelée matrice cophénétique, représente le diagramme hiérarchique (dendrogramme) produit par l'algorithme : chaque élément $M_{coph_{i,j}}$ de la matrice cophénétique représente le niveau pour lequel les objets o_i et o_j se retrouvent pour la première fois dans la même classe. Un indice de proximité entre une matrice cophénétique ($M_{coph_{i,j}}$) et la matrice d'adjacence MC_{P_h} d'une partition P en z classes a été établi et appelé coefficient de corrélation cophénétique (CPCC):

$$CPCC = \frac{\frac{1}{M}\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} M_{coph_{i,j}} MC_{P_{h_{i,j}}} - \mu_{M_{coph}}\mu_{MC_{P_h}}\right)}{\sqrt{\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} MC^2_{P_{h_{i,j}}} - \mu^2_{M_{coph}}\right)\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} M^2_{coph_{i,j}} - \mu^2_{MC_{P_h}}\right)}}$$

$$-1 \leq CPCC \leq 1$$

Avec, n le nombre d'objets, $M = \frac{n(n-1)}{2}$ le nombre de paires d'objets différents,

$$\mu_{M_{coph}} = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} M_{coph_{i,j}} \qquad \mu_{MC_{P_h}} = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} MC_{P_{h_{i,j}}}$$

Une procédure de type Monte Carlo peut être employée afin d'évaluer la fonction densité de probabilité de l'indice CPCC et procéder ultérieurement aux tests statistiques.

Dans la plupart des cas, l'utilisation de critères internes permet des comparaisons entre cns résultant de processus de classification relativement similaires et notamment en ce qui concerne la mesure de distance/similarité sous jacente à ce processus. Ainsi, dans de nombreuses situations (comme, par exemple, la comparaison de la validité de cns résultant de deux algorithmes employant des mesures de similarité très largement différents) un consensus sur la mesure de validité interne à utiliser ne peut pas être trouvé. Cela explique alors d'une part que, lors d'évaluations expérimentales d'un algorithme de cns, les mesures externes soient privilégiées si l'on dispose des informations nécessaires à leur mise en œuvre, et d'autre part que lorsqu'on est confronté à un problème d'évaluation de validité de cns sans posséder ces informations on privilégie souvent les modes d'évaluation relatifs qui impliquent un effort calculatoire moindre.

4.1.3 Modes d'Evaluation Relatifs

Les deux approches précédentes sont souvent basées sur des tests statistiques pouvant nécessiter un effort calculatoire important. L'approche évaluation de la validité par critères relatifs est différente et s'appuie sur le principe "choisir la meilleure cns parmi un ensemble de cns selon un critère prédéfini".

Plus précisément, le problème peut être posé de la manière suivante : "Soit P_{alg} l'ensemble des paramètres associés à un algorithme particulier (le nombre final de classes par exemple); parmi un ensemble de cns $\{P_i, i = 1..k\}$ obtenues par l'intermédiaire d'un même algorithme mais avec différents paramètres de P_{alg}, choisir celui traduisant le mieux la structure des données."

Ce problème possède diverses déclinaisons selon la constitution de l'ensemble P_{alg}, chacune de ces déclinaisons possédant une solution pratique propre :

- **Cas 1** : Le nombre final de classes, nc, n'est pas contenu dans P_{alg}
- **Cas 2** : Le nombre final de classes, nc, est contenu dans P_{alg}.

4.1.3.1 Cas 1 : Le nombre final de classes, nc, n'est pas contenu dans P_{alg}.

Dans ce cas, la détermination des valeurs optimales pour les paramètres se déroule comme suit :

- l'utilisateur lance l'algorithme pour une large gamme de valeurs de paramètres
- la plage de valeurs la plus large pour laquelle le nombre final de classes reste constant est ensuite sélectionnée.
- les valeurs correspondant au centre de cette plage sont alors choisies comme paramètres valides.

Cette procédure permet donc de déterminer les paramètres de l'algorithme de cns et le nombre de classes de la cns correspondant au mieux à la structure interne des données.

4.1.3.2 Cas 2 : Le nombre final de classes, nc, est contenu dans P_{alg}.

Dans ce cas, la procédure d'identification de la meilleure cns implique l'utilisation d'un indice de validité ; la détermination des valeurs optimales pour les paramètres se déroule quant à elle comme suit :

- l'utilisateur spécifie dans un premier temps un intervalle de valeurs $[nc_{min}; nc_{max}]$ qui doit comprendre le bon nombre final de classes nc
- pour chaque valeur de $nc \in [nc_{min}; nc_{max}]$l'algorithme est alors lancé x fois avec des valeurs distinctes pour les autres paramètres de P_{alg} (par exemple, des conditions initiales différentes dans le cas de méthodes de type K-means). A chaque jeu de paramètres correspond alors une cns et une valeur pour l'indice de validité employé.
- les meilleures valeurs de l'indice de validité obtenues pour chaque nombre final de classes sont alors représentées graphiquement.

L'étude de ce graphique permet de choisir la meilleure cns. Dans la mesure où certains indices ne sont pas indépendants du nombre de classes de la cns (certains indices, tels ceux basés uniquement sur l'inertie interne des classes par exemple, ont tendance à croître ou décroître pour des nombres de classes croissants), on ne se contente pas de rechercher la cns possédant la plus faible (ou plus forte) valeur pour l'indice :

- Si l'indice ne montre pas de tendance croissante (ou décroissante) on se contente de rechercher la valeur la plus forte (la plus faible) pour l'indice qui correspond dans ce cas à la cns la plus en adéquation avec la structure sous jacente aux données.
- Si, par contre, l'indice exhibe une tendance croissante (ou décroissante) on recherche alors la valeur correspondant au changement local le plus marquant : il s'agit ici de rechercher un "coude" dans le graphique. Cette valeur correspond alors à la cns la plus en adéquation avec la structure sous jacente aux données. L'absence d'un tel changement peut ici être considérée comme le signe d'une absence de structure interne dans les données.

4.1.3.3 Indices

Les indices que l'on peut utiliser dans le cadre de cns non floue[5] sont :

- *Statistique* Γ *de Hubert Modifiée et* Γ *de Hubert normalisée* $\overline{\Gamma}$

5. nous n'introduisons pas les indices employer dans le cadre de la cns floue, et invitons le lecteur intéresser à ce référer au références données au début de ce papier pour un approfondissement ou encore à [XB91], [BEF84], ...

– *Les Mesures Dunn et Dunn apparentées* [Dun74] L'indice de Dunn [Dun74] tente de permettre l'identification de classes homogènes et bien séparées. Cet indice se définit pour un nombre fixé de classe comme suit

$$D_{nc} = min_{i=1..nc}\left(min_{j=i+1..nc}\left(\frac{d(c_i,c_j)}{max_{k=1..nc}diam(c_k)}\right)\right)$$

avec $d(c_i,c_j)$ la distance entre les classes c_i et c_j : $d(c_i,c_j) = min_{x\in c_i, y\in c_j} d(x,y)$ et $diam(c_k)$ le diamètre de la classe c_k : $diam(c_k) = max_{x,y\in c_k} d(x,y)$. $d(c_i,c_j)$ peut être considérée comme une mesure de la séparation de deux classes, et $diam(c_k)$ une mesure de l'hétérogénéité d'une classe. Ainsi, s'il existe des classes homogènes et bien séparées l'indice doit présenter une forte valeur. Notons tout d'abord que cet indice présente l'avantage de ne pas présenter de tendance (croissante ou décroissante) en rapport avec nc, il peut donc être utilisé pour déterminer le nombre de classes de la cns la plus en adéquation avec les données. Par contre, le coût calculatoire qui lui est associé est relativement important et il présente une forte sensibilité au bruit (qui peut impliquer un accroissement des valeurs de $diam(c_k)$). Trois indices proposés dans [PB97] constituent des adaptations plus robustes de cet indice. Ces trois indices utilisent des graphes de voisinage : l'arbre recouvrant minimal (arm), le graphe des voisins relatifs (gvr), le graphe de Gabriel. L'introduction des graphes de voisinage permet de redéfinir $diam(c_k)$. Si l'on considère par exemple l'adaptation de l'indice de Dunn impliquant l'arm, on associe à chaque classe c_k l'arm lui correspondant et $diam(c_k)$ est alors défini comme le poids de l'arête la plus fortement valuée et que l'on note $diam^{arm}(c_k)$.

$$D_{nc} = min_{i=1..nc}\left(min_{j=i+1..nc}\left(\frac{d(c_i,c_j)}{max_{k=1..nc}diam^{arm}(c_k)}\right)\right)$$

Ainsi, la robustesse de l'indice est accrue, toutefois le coût calculatoire associé constitue là encore un point faible de ce type d'indice. (Les adaptations de l'indice de Dunn utilisant les graphes des voisins relatifs ou de Gabriel sont basées sur une adaptation similaire de celle présentée pour l'arm).

– *Indice de Davies Bouldin*[DB79] [PB97] Afin d'introduire cet indice, on définit une mesure de similarité R_{ij} entre deux classes c_i et c_j basée sur une mesure s_i d'hétérogénéité interne d'une classe c_i et une mesure de dissimilarité d_{ij} entre deux classes c_i et c_j. Cette mesure est définie de manière telle que les propriétés suivantes soient respectées :
  - $R_{ij} \geq 0$
  - $R_{ij} = R_{ji}$
  - si $s_i = 0$ et $s_j = 0$ alors $R_{ij} = 0$
  - si $s_j > s_k$ et $dij = d_{ik}$ alors $R_{ij} > R_{ik}$
  - si $s_j = s_k$ et $dij < d_{ik}$ alors $R_{ij} > R_{ik}$

Davies et Bouldin ont proposé la mesure R_{ij} suivante : $R_{ij} = \frac{s_i+s_j}{d_{ij}}$. L'indice DB est alors défini comme suit :

$$DB_{nc} = \frac{1}{nc} \sum_{i=1..nc} R_i, \qquad R_i = max_{j=1..nc, i \neq j} R_{ij}$$

Selon cette définition l'indice DB_{nc} correspond donc à la similarité moyenne entre chaque classe et sa classe la plus similaire. Dans la mesure où l'on cherche des classes telles qu'elles soient le moins similaires possible des autres classes, on cherche donc à minimiser DB_{nc}, de plus cet indice ne présente pas de tendance en relation avec le nombre de classes. Des définitions pour la dissimilarité entre classes ainsi que pour l'hétérogénéité interne d'une classe sont proposées dans [DB79]. Enfin, [PB97] ont introduit 3 variantes de cet indice utilisant les graphes de voisinage de manière analogue à l'indice de Dunn.

- *Indice de validité SD*, dans [Hal00a] Halkidi et al. proposent un indice basé sur les concepts de dispersion moyenne des classes et de séparation totale entre classes.
- *SDbw*, récemment proposé par Halkidi et al. [HV01], cet indice exploite les caractéristiques inhérentes aux classes d'une cns pour en estimer la validité et permettre la détermination du partitionnement optimal des données. Tout comme l'indice SD, cet indice est basé sur les concepts de dispersion moyenne des classes et de séparation totale entre classes et introduit la notion supplémentaire de densité. Plus récemment encore, [KL03] proposent $SDbw^{\star}$ une évolution de cet indice.
- *RMSSTD, SPR, RS, CD*[Sha96] Ces 4 indices nécessitent une utilisation simultanée, et doivent être appliqués à chaque étape d'un processus de cns hiérarchique. Ils sont définis de la manière suivante :
  - *Root Mean Square STandard Deviation (RMSSTD)* (racine carrée de la moyenne des carrés des écarts types) de la cns : cet indice mesure l'homogénéité de la cns associée à une étape de la cns hiérarchique par l'intermédiaire de la moyenne de la variance de chaque variable au sein de chaque classe de la cns. Cette mesure se doit donc d'être la plus faible possible afin de montrer une forte homogénéité des classes de la cns. Si l'on observe, lors d'un passage d'une cns vers la cns suivante possédant plus de classes, un accroissement de la valeur de la mesure cela signifie alors un problème d'homogénéité dans cette dernière cns.
  - *Semi-Partial R squared (SPR)* : cet indice mesure la différence d'homogénéité locale sur deux cns successives de la cns hiérarchique : il s'agit de mesurer la différence d'homogénéité entre une classe d'une cns (ca) et celle des deux classes de la cns précédente qui ont été fusionnée afin de "créer" ca. Ainsi, une valeur faible indiquera qu'il y a eu fusion de deux classes relativement homogènes

alors qu'une valeur élevée signifie que les deux classes fusionnées ne sont pas homogènes.

- *R Squared (RS)* mesure l'hétérogénéité entre classes, ses valeurs sont comprises entre 0 et 1 : une valeur proche de 0 indiquant une faible hétérogénéité entre classes alors qu'une valeur proche de 1 signifie une hétérogénéité significative entre classes.
- *Distance between two Clusters (CD)* (Distance entre deux classes), cet indice mesure la distance entre les classes qui sont fusionnées à un niveau donnée de la cns hiérarchique.

L'utilisation simultanée de ces 4 indices permet de déterminer le nombre de classes le plus approprié pour une cns d'un jeu de données particulier. Pour cela on s'appuie sur une étude graphique des valeurs de ces différents indices et l'on recherche le "coude" le plus important pour les courbes associées à ces indices. Ces indices, et plus particulièrement les indices *RMSSTD* et *RS*, peuvent être utilisés pour des cns non hiérarchiques. L'idée étant ici de lancer l'algorithme pour des nombres de classes différents puis de procéder à une étude graphique similaire.

L'évaluation de la validité de cns par mode d'évaluation relatif correspond ainsi à l'utilisation d'indices de type mesure interne associée à une analyse graphique des valeurs de l'indice utilisé pour différentes cns.

4.1.4 Autres Modes d'Evaluation

D'autres modes d'évaluation sont proposés dans la littérature tels celui proposé par Smyth [Smy96] et les approches d'évaluation de la stabilité des algorithmes de cns :

- Smyth [Smy96] introduit en effet un algorithme de cns basé sur la cross-validation ainsi que la méthode de Monte-Carlo. Plus précisément, cet algorithme consiste en γ cross validations sur γ échantillons d'apprentissage/test du jeu de données. Pour chaque échantillon du jeu de données ech, l'algorithme EM est utilisé pour déterminer une cns en nc classes de la partie d'apprentissage de l'échantillon, cette étape est réitérée nc_{max} fois pour des valeurs de nc allant de 1 à nc_{max}. La log-vraisemblance $L^u_{nc}(D)$ est ensuite calculée pour chaque cns à nc classes, elle est formellement définie en utilisant la fonction de densité de probabilité des données : $Lk(D) = \sum_{i=1..n} log(f_k(x_i|\varphi_k))$; avec f_k la fonction de densité de probabilité des données et φ_k l'ensemble des paramètres estimés à partir des données. Cela est ainsi répété γ fois, puis on calcule la moyenne des γ estimations réalisées par cross-validation pour chaque valeur de nc. Sur la base de ces estimations, il est alors possible de déterminer les probabilités associées à chaque valeur nc : $P(nc|D)$, l'étude de ces probabilités permettant de mettre en évidence le nombre de classes correspondant au mieux au jeu de données (s'il existe). Cette approche est basée sur des concepts probabilistes afin d'estimer le nombre de classes correspondant

au mieux aux données mais n'utilise pas des concepts plus directement liés aux données tels que la séparation entre classes ou l'homogénéité interne des classes.

– Les approches utilisées pour l'évaluation de la stabilité des algorithmes de cns peuvent également être utilisées pour évaluer dans quelle mesure une modification de l'ensemble des objets sur lequel est réalisée la cns implique une modification de la partition résultat. Des méthodes d'échantillonnage et de comparaison des partitions obtenues sont disponibles [LD01] pour ce type d'évaluation. L'idée est donc ici d'évaluer la stabilité de l'algorithme de cns pour des nombres de classes différents et de choisir le nombre de classes impliquant la plus forte stabilité.

Notons que ces approches permettent essentiellement la détermination du nombre de classes le plus valide pour une cns d'un jeu de données spécifique et qu'elles ne permettent donc pas vraiment la comparaison de validité de deux partitions. De plus, il faut également noter la dépendance de ces approches avec les méthodes de cns employées.

4.2 Nouveaux Indices et Nouvelle Méthodologie pour l'Evaluation et la Comparaison de la Validité de Classifications Non Supervisées

Nous proposons maintenant deux nouveaux indices de type critère interne (qui ne nécessitent donc aucune connaissance ou intuition a priori sur les données) utilisés au sein d'une méthodologie particulière de type évaluation relative pour l'évaluation et la comparaison de la validité de cns. Cette méthodologie bien qu'utilisant des caractérisations statistiques n'implique pas l'utilisation de la méthode de Monte Carlo et son coût calculatoire est relativement faible. Elle ne nécessite qu'une seule passe sur le jeu de données, et, son coût calculatoire est linéaire selon le nombre de variables du jeu de données et soit linéaire selon le nombre d'objets dans le cas de données catégorielles, soit quadratique selon le nombre d'objets pour des données quantitatives. Notons enfin que, l'encombrement mémoire associé à cette méthodologie est très faible quel que soit le type de données traité (nécessite le stockage d'un ensemble de tables de contingences). Ces indices peuvent être utilisés pour tous types de données mais sont toutefois spécialement adaptés au traitement de données catégorielles. Enfin, l'utilisation de cette méthodologie permet une représentation graphique de la validité des cns qui est particulièrement intéressante dans la mesure où par un simple artifice de visualisation, il est possible de dériver des connaissances additionnelles concernant la structure des données.

4.2.1 Concepts et Formalismes Introductifs

Notation 1
Nous rappelons ici les notations utilisées :
$O = \{o_i, i = 1..n\}$ *un ensemble d'objets*
$EV = \{V_1,..,V_p\}$ *l'espace, constitué de* p *variables décrivant les objets de* O.
$o_i = \{o_{i_1},..,o_{i_p}\}$ *un objet de* O, o_{i_j} *correspond à la valeur de* o_i *pour la variable* V_j *(cette valeur peut être numérique, catégorielle...)*
C_k *un ensemble d'objets de* O ($C_k \subseteq O$),
$P_h = \{C_1,...,C_z\}$ *une partition de* O *en* z *classes*

De manière classique, l'évaluation de la validité d'une cns s'appuie sur l'étude d'un critère traduisant l'homogénéité interne de ses classes ou la séparation de ses classes. Certains indices tirent toutefois partie d'une évaluation combinée de ces deux notions au sein d'un unique indice (les indices SD, $SDbw$, $SDbw^\star$ par exemple), d'autres indices s'utilisent quant à eux au travers d'évaluations graphiques simultanées mais séparées (les indices *RMSSTD, SPR, RS, CD* par exemple). La méthodologie que nous proposons prend, elle aussi, en compte ces deux notions, mais plutôt que d'introduire un critère combinant ces notions, nous proposons d'utiliser simultanément deux critères rendant compte pour l'un de l'hétérogénéité interne des classes et de la séparation entre classes pour l'autre et ce au travers d'une unique représentation graphique et non plusieurs.

Nous utilisons en fait, deux mesures traduisant respectivement l'homogénéité interne des classes et la séparation entre classes. Ces mesures sont en relation directe avec le Nouveau Critère de Condorcet (*NCC*)[6].

Dans le cadre des données catégorielles[7] la notion de similarité entre objets est utilisée ; afin de proposer une méthodologie utilisable pour des données quantitatives, nous lui substituons une extension de cette notion. Cette extension, nommée lien (selon une variable), se définit comme suit :

Définition 9 ***Lien entre 2 objets***
A chaque variable V_i *est associée une fonction* $Lien_i$ *qui définit un lien (une sorte de similarité) ou un non-lien (une sorte de dissimilarité) selon* V_i *entre deux objets de* O *(*o_a *et* o_b*) :*

$$Lien_i(o_{a_i},o_{b_i}) = \begin{cases} 1 \textit{ si une condition particulière déterminant un lien} \\ \textit{(selon } V_i\textit{) entre les objets } o_a \textit{ et } o_b \textit{ est vérifiée} \\ 0 \textit{ sinon (non-lien)} \end{cases} \quad (4.1)$$

6. mesure de qualité d'une cns proposée par Michaud [Mic97] et utilisée pour la cns pour données catégorielle [Mic97], [JN03c], voir chapitre 2 pour des développements plus complets

7. qui constitue le cadre naturel pour la définition du NCC

EXEMPLES :

- *Pour une variable catégorielle V_i, on peut définir naturellement $Lien_i$ comme suit :*
  $Lien_i(o_{a_i},o_{b_i}) = \delta_{sim}(o_{a_i},o_{b_i}) = \begin{cases} 1 \text{ si } o_{a_i} = o_{b_i} \\ 0 \text{ sinon} \end{cases}$
- *Pour une variable quantitative V_i, on peut par exemple définir $Lien_i$ comme suit :*
  $Lien_i(o_{a_i},o_{b_i}) = \begin{cases} 1 \text{ si } |o_{a_i} - o_{b_i}| \leq \delta, \text{ avec } \delta \text{ un seuil fixé par l'utilisateur} \\ 0 \text{ sinon} \end{cases}$
- *Pour une variable quantitative V_i, on peut également utiliser la discrétisation puis utiliser une fonction $Lien_i$ définie pour les variables catégorielles.*

Nous illustrons le cas des variables quantitatives sur un jeu de données composé de 4 objets décrits par deux variables quantitatives V_1 et V_2 :
$o_1 = \{1; 2\}$, $o_2 = \{1.5; 1\}$, $o_3 = \{2; 1\}$, $o_4 = \{1; 3\}$.

On considère 2 cas :
*- **le cas 1** caractérisé par une discrétisation des 2 variables selon les intervalles :* $]-\infty; 1.25[; [1.25; +\infty[$ *pour V_1 et* $[-\infty; 1.5[; [1.5; 2.5[; [2.5; +\infty[$ *pour V_2*
*- **le cas 2** se caractérise quant à lui par l'utilisation d'une fonction de type seuil pour les 2 variables : un seuil de 0.5 pour V_1 et de 1 pour V_2.*

Les tableaux 4.1 et 4.2 ainsi que la figure 4.1 illustrent les différentes valeurs pour les fonctions $Lien_1$ et $Lien_2$ pour ces deux cas.

	o_1	o_2	o_3	o_4
o_1	x	0	0	1
o_2	0	x	1	0
o_3	0	1	x	0
o_4	1	0	0	x

	o_1	o_2	o_3	o_4
o_1	x	0	0	1
o_2	0	x	1	0
o_3	0	1	x	1
o_4	1	0	1	x

TAB. 4.1 –: *Fonctions $Lien_1$ (à gauche) et $Lien_2$ (à droite) pour le cas 1*

	o_1	o_2	o_3	o_4
o_1	x	0	0	0
o_2	0	x	1	0
o_3	0	1	x	0
o_4	0	0	0	x

	o_1	o_2	o_3	o_4
o_1	x	0	0	1
o_2	0	x	1	1
o_3	0	1	x	1
o_4	1	1	1	x

TAB. 4.2 –: *Fonctions $Lien_1$ (à gauche) et $Lien_2$ (à droite) pour le cas 2*

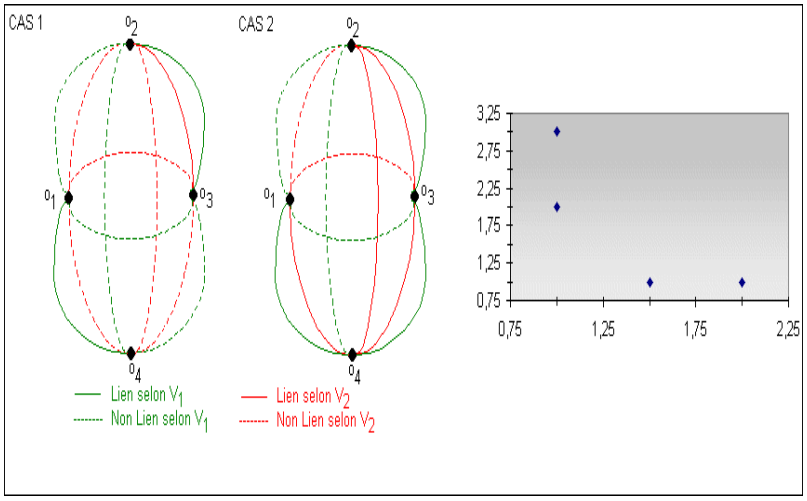


FIG. 4.1 –: *Représentation graphique des 4 objets dans l'espace 2D (à droite) et Représentation des liens/non-liens unissant les objets dans les 2 cas illustratifs (à gauche)*

4.2.1.1 Evaluation de l'homogénéité interne des classes d'une cns

Pour évaluer l'homogénéité interne d'une cns (une partition P_h de O), on peut utiliser la mesure LM (resp. NLM) qui dénombre le nombre de liens (resp. non-liens) entre objets de même classe de la cns. Ces mesures sont définies de la manière suivante :

$$LM(P_h) = \sum_{g=1..z} \left(\sum_{\substack{o_k \in C_g, o_l \in C_g, \\ k<l}} \left(\sum_{i=1..p} (lien_i\,(o_{k_i}, o_{l_i})) \right) \right) \tag{4.2}$$

$$0 \leq LM(P_h) \leq p \times \sum_{g=1..z} \frac{card(C_g)(card(C_g)-1)}{2} \tag{4.3}$$

$$NLM(P_h) = \sum_{g=1..z} \left(\sum_{\substack{o_k \in C_g, o_l \in C_g, \\ k<l}} \left(\sum_{i=1..p} (1 - lien_i\,(o_{k_i}, o_{l_i})) \right) \right) \tag{4.4}$$

$$0 \leq NLM(P_h) \leq p \times \sum_{g=1..z} \frac{card(C_g)(card(C_g)-1)}{2} \tag{4.5}$$

$$LM(P_h) + NLM(P_h) = p \times \sum_{g=1..z} \frac{card(C_g)(card(C_g)-1)}{2} \tag{4.6}$$

Ainsi, l'homogénéité interne d'une cns P_h est d'autant plus forte que $LM(P_h)$ (resp. $NLM(P_h)$) est élevée (resp. faible).

EXEMPLE : *Pour les 2 cas introduits précédemment et pour une partition* $P_h = \{\{o_1,o_4\},\{o_2,o_3\}\}$ *nous avons :* $LM(P_h) = 3$, $NLM(P_h) = 1$ *dans le cas 1 ; et* $LM(P_h) = 4$, $NLM(P_h) = 0$ *dans le cas 2.*

4.2.1.2 Evaluation de la séparation entre classes d'une cns (ou hétérogénéité entre classes)

Pour évaluer la séparation entre classes d'une cns (une partition P_h de O), on peut utiliser la mesure LD (resp. NLD) qui dénombre le nombre de liens (resp. non-liens) entre objets de classes différentes de la cns. Ces mesures sont définies de la manière suivante :

$$LD(P_h) = \sum_{\substack{f=1..z, g=1..z \\ f<g}} \left(\sum_{o_k \in C_f, o_l \in C_g,} \left(\sum_{i=1..p} (lien_i\,(o_{k_i}, o_{l_i})) \right) \right) \tag{4.7}$$

$$0 \leq LD(P_h) \leq p \times \sum_{\substack{f=1..z, g=1..z \\ f<g}} \frac{card(C_g)(card(C_f)}{2} \tag{4.8}$$

$$NLD(P_h) = \sum_{\substack{f=1..z, g=1..z \\ f<g}} \left(\sum_{o_k \in C_f, o_l \in C_g,} \left(\sum_{i=1..p} (1 - lien_i\,(o_{k_i}, o_{l_i})) \right) \right) \tag{4.9}$$

$$0 \leq NLD(P_h) \leq p \times \sum_{\substack{f=1..z, g=1..z \\ f<g}} \frac{card(C_g)(card(C_f)}{2} \tag{4.10}$$

$$LD(P_h) + NLD(P_h) = p \times p \times \sum_{\substack{f=1..z, g=1..z \\ f<g}} \frac{card(C_g)(card(C_f)}{2} \tag{4.11}$$

Ainsi, la séparation des classes d'une cns P_h est d'autant plus forte que $NLD(P_h)$ (resp. $LD(P_h)$) est élevée (resp. faible).

EXEMPLE : *Pour les 2 cas introduits précédemment et pour une partition* $P_h = \{\{o_1,o_4\},\{o_2,o_3\}\}$ *nous avons :* $LD(P_h) = 0$, $NLD(P_h) = 8$ *dans le cas 1 ; et* $LD(P_h) = 4$, $NLD(P_h) = 4$ *dans le cas 2.*

4.2.1.3 Notions Additionnelles

Nous définissons deux mesures additionnelles $M(P_h)$ et $D(P_h)$ qui correspondent respectivement :

- au nombre total de liens et non liens entre objets de même classe de P_h : $M(P_h) = NLM(P_h) + LM(P_h)$
- au nombre total de liens et non liens entre objets de classes différentes de P_h: $D(P_h) = NLD(P_h) + LD(P_h)$.

Finalement, nous notons $L(O)$ (resp. $NL(O)$) le nombre total de liens (resp. de non-liens) entre objets de O: $L(O) = LM(P_h) + LD(P_h)$ (resp. $NL(O) = NLM(P_h) + NLD(P_h)$).

REMARQUE : Pour des données catégorielles, le critère *NCC* est défini comme la somme de $NLM(P_h)$ et $LD(P_h)$.

EXEMPLE : *Pour les 2 cas introduits précédemment et pour une partition* $P_h = \{\{o_1,o_4\},\{o_2,o_3\}\}$ *nous avons :*
$M(P_h) = 4$, $D(P_h) = 8$, $L(O) = 3$, $NL(O) = 9$ *dans le cas 1 ; et* $M(P_h) = 4$, $D(P_h) = 8$, $L(O) = 8$, $NL(O) = 4$ *dans le cas 2.*

RÉSUMÉ :

$L(O) + NL(O) = \frac{n\times(n-1)}{2} \times p$

$M(P_h) + D(P_h) = \frac{n\times(n-1)}{2} \times p$

$M(P_h) = NLM(P_h) + LM(P_h)$

$D(P_h) = NLD(P_h) + LD(P_h)$

$L(O) = LM(P_h) + LD(P_h)$

$NL(O) = NLM(P_h) + NLD(P_h)$

Ces relations peuvent être synthétisées au sein d'une sorte de table de contingence de type comparaison par paires :

	liens	non liens	$Total$
même classes	LM	NLM	$M(C)$
classes diff.	LD	NLD	$D(C)$
$Total$	$L(O)$	$NL(O)$	$\frac{n(n-1)}{2}p$

4.2.1.4 Remarques importantes concernant l'aspect calculatoire

Bâtir cette table de contingence ne nécessite qu'une seule passe sur le jeu de données. Dans le cas de données catégorielles, cela ne requiert que $O(np)$ comparaisons afin de bâtir p tables de contingence (croisant les p variables avec la variable catégorielle virtuelle impliquée par la partition P_h)(intuitivement, les définitions formelles de LM, NLM, LD et NLD semblent impliquer $O(n^2p)$ comparaisons mais des astuces de calcul permettent de réduire ce nombre de comparaisons, voir l'exemple illustratif suivant), ce nombre de comparaisons peut atteindre $O(n^2p)$ en cas de présence de variables quantitatives et d'utilisation de fonctions $lien_i$ telles que définies dans le cas 2 des exemples illustratifs précédents.

Du point de vue de l'utilisation mémoire, quelle que soit la nature des données (catégorielles ou numériques), le stockage de p tables de contingence est nécessaire, ce qui correspond à un encombrement mémoire faible.

EXEMPLE :
Considérons un jeu de données synthétique composé de 4 objets ($o_1 = [y,y,n,n]$, $o_2 = [y,y,n,n]$, $o_3 = [n,y,y,y]$, $o_4 = [n,n,y,y]$) décrits par 4 variables catégorielles ($EV = \{V_1,V_2,V_3,V_4\}$) (voir table 4.3).
Considérons également la partition P_h suivante : $P_h = \{C_1,C_2\} = \{\{o_1,o_2\},\{o_3,o_4\}\}$.

Nous exposons maintenant comment calculer les valeurs des divers indices présentés dans la section précédente. (Nous utilisons ici la fonction $Lien$ telle qu'elle est définie dans l'exemple sur les données catégorielle de la définition 9).

	V_1	V_2	V_3	V_4
1	y	y	n	n
2	y	y	n	n
3	n	y	y	y
4	n	n	y	y

TAB. 4.3 –: *Jeu de données synthétique*

– *En une unique passe sur le jeu de données on peut bâtir les tables de contingence croisant la variable catégorielle virtuelle V_A impliquée par la partition P_h (V_A possède deux modalités a et b qui correspondent respectivement aux classes $\{o_1,o_2\}$ et $\{o_3,o_4\}$) avec les p variables de EV (V_1, V_2, V_3, V_4.)* :

$V_A \setminus V_1$	y	n	$V_A \setminus V_2$	y	n	$V_A \setminus V_3$	y	n	$V_A \setminus V_4$	y	n
a	2	0	a	2	0	a	0	2	a	2	0
b	0	2	b	1	1	b	2	0	b	0	2

Tables de Contingence croisant la variable virtuelle V_A et les variables de EV

– *le calcul de la valeur de chaque indice est alors réalisé à partir de ces tables. Si la table de contingence pour une variable V_i est notée:*

$V_A \backslash V_i$	V_{i_1}	...	$V_{i_{m_i}}$	
V_{A_1}	α_{1i1}	...	α_{1im_i}	$\alpha_{1i.}$
...	...	...	...	...
V_{A_z}	α_{zi1}	...	α_{zim_i}	$\alpha_{zi.}$
	$\alpha_{.i_1}$	...	$\alpha_{.im_i}$	n

V_A la variable catégorielle virtuelle à z modalités (associée à P_h),
V_i une variable exogène à m_i modalités notées $V_{i_j}(j = 1..m_i)$.
α_{li_h} le nombre d'objets ayant la valeur V_{i_h} pour V_i et la valeur V_{A_l} pour V_A.
$\alpha_{.i_j} = \sum_{h=1..z} \alpha_{hi_j}$; $\alpha_{hi.} = \sum_{j=1..m_i} \alpha_{hi_j}$

– *on peut alors calculer :*

$$LM(P_h) = \sum_{\substack{i = 1..p \\ \text{tel que } V_i \in EV}} \sum_{j=1..m_i} \sum_{t=1..z} \frac{\alpha_{ti_j}(\alpha_{ti_j}-1)}{2}$$

$$M(P_h) = card(EV) \times \sum_{t=1..z} \frac{card(C_t)(card(C_t)-1)}{2}$$

$$L(O) = \sum_{\substack{i=1..p \text{ tel que} \\ V_i \in EV}} \sum_{j=1..m_i} \frac{\alpha_{.i_j}(\alpha_{.i_j}-1)}{2}$$

$$NLM(P_h) = M(P_h) - LM(P_h)\,;\, LD(P_h) = L(O) - LM(P_h)$$

$$D(P_h) = \frac{n(n-1)}{2} \times card(EV) - M(P_h)\,;\, NLD(P_h) = D(P_h) - LD(P_h)$$

– *Pour l'exemple cela donne :*
$LM(P_h) = \left(\frac{2\times1}{2} + \frac{0\times(-1)}{2} + \frac{0\times(-1)}{2} + \frac{2\times1}{2}\right) + \left(\frac{2\times1}{2} + \frac{1\times0}{2} + \frac{0\times(-1)}{2} + \frac{1\times0}{2}\right) +$
$\left(\frac{0\times(-1)}{2} + \frac{2\times1)}{2} + \frac{2\times1}{2} + \frac{0\times-1}{2}\right) + \left(\frac{2\times1}{2} + \frac{0\times(-1)}{2} + \frac{0\times(-1)}{2} + \frac{2\times1}{2}\right) = 7$

$$M(P_h) = 4 \times \left(\frac{2\times1}{2} + \frac{2\times1}{2}\right) = 8$$

$$L(O) = \left(\frac{2\times1}{2} + \frac{2\times1}{2}\right) + \left(\frac{3\times2}{2} + \frac{1\times0}{2}\right) + \left(\frac{2\times1}{2} + \frac{2\times1}{2}\right) + \left(\frac{2\times1}{2} + \frac{2\times1}{2}\right) = 9$$

$$NLM(P_h) = 8 - 7 = 1 \quad ; \quad LD(P_h) = 9 - 7 = 2$$

$$D(P_h) = \frac{4\times3}{2} \times 4 - 8 = 16 \quad ; \quad NLD(P_h) = 16 - 2 = 14$$

4.2.2 La nouvelle méthodologie pour l'évaluation et la comparaison de validité de cns

Il apparaît intuitivement, qu'une cns valide (P_h) doit être telle que les objets de même classe sont majoritairement reliés par des liens et que les objets de classes différentes sont majoritairement reliés par des non-liens, ainsi une cns valide doit présenter de fortes valeurs pour $LM(P_h)$ et $NLD(P_h)$ ce qui implique de faibles valeurs pour $NLM(P_h)$ et $LD(P_h)$ (cela signifie alors une forte homogénéité interne des classes ainsi qu'une forte séparation des classes de la cns).

Cependant, la signification de fortes et faibles valeurs n'est elle pas totalement intuitive. Nous devons de surcroît remarquer que ces valeurs ne sont pas indépendantes et que si elles sont fortement corrélées entre elles, elles sont aussi corrélées avec le nombre de classes de la cns. En outre, très peu de jeux de données possèdent une structure interne telle que $LM(P_h)$ et $NLD(P_h)$ sont simultanément maximisés pour une partition P_h. Nous décrivons donc maintenant une méthodologie pour l'évaluation et la comparaison de la validité de cns (d'un même jeu de données) par analyse de leurs valeurs respectives pour LM et NLD dans le but de déterminer l'unique ou le sous ensemble de cns les plus valides. Cette méthodologie constitue un outil utile pour les utilisateurs qui recherchent la meilleure, ou tout au moins, une bonne cns pour un

jeu de données. Cette méthodologie est indépendante des méthodes utilisées, des paramètres des méthodes, ainsi que du nombre de classes. Pour atteindre cet objectif, nous utilisons une approche statistique de manière à déterminer dans quelle mesure des valeurs LM et NLD exhibées par une cns peuvent être considérées comme significativement élevées.

4.2.2.1 Caractérisation statistique des valeurs de : *LM* et *NLD*

Faisons l'hypothèse H_0 d'une organisation aléatoire de l'ensemble d'objets O selon une partition P_h en z classes. Nous pouvons déterminer la loi statistique suivie par LM et NLD sous cette hypothèse :

- $LM(P_h)$ suit une loi binomiale de paramètres $M(P_h)$ et $\frac{L(O)}{L(O)+NL(O)}$, ce que nous notons :
  $LM(P_h) \hookrightarrow B(M(P_h), \frac{L(O)}{L(O)+NL(O)})$;
- $NLD(P_h)$ suit une loi binomiale de paramètres $D(P_h)$ et $\frac{NL(O)}{L(O)+NL(O)}$, ce que nous notons :
  $NLD(P_h) \hookrightarrow B(D(P_h), \frac{NL(O)}{L(O)+NL(O)})$.

Par approximation avec la loi normale, on obtient :

- $LM(P_h)$ suit une loi normale
  de moyenne : $E_1 = M(P_h) \times \frac{L(O)}{L(O)+NL(O)}$,
  et d'écart type : $SD_1 = \sqrt{M(P_h) \times \frac{L(O)}{L(O)+NL(O)} \times (1 - \frac{L(O)}{L(O)+NL(O)})}$,
  Nous notons : $LM(P_h) \hookrightarrow N(E_1, SD_1)$;

- $NLD(P_h)$ suit une loi normale
  de moyenne : $E_2 = D(P_h) \times \frac{NL(O)}{L(O)+NL(O)}$,
  et d'écart type : $SD_2 = \sqrt{D(P_h) \times \frac{NL(O)}{L(O)+NL(O)} \times (1 - \frac{NL(O)}{L(O)+NL(O)})}$,
  Nous notons : $NLD(P_h) \hookrightarrow N(E_2, SD_2)$.

Dès lors, par centrage-réduction on obtient deux indices $xv_1(P_h)$ et $xv_2(P_h)$ qui suivent une loi normale centrée réduite :

- $xv_1(P_h) = \frac{LM(P_h) - E_1}{SD1} = \frac{LM(P_h) - M(P_h) \times \frac{L(O)}{L(O)+NL(O)}}{\sqrt{M(P_h) \times \frac{L(O)}{L(O)+NL(O)} \times (1 - \frac{L(O)}{L(O)+NL(O)})}}$
  $xv_1(P_h) \hookrightarrow N(0,1)$
- $xv_2(P_h) = \frac{NLD(P_h) - E_2}{SD2} = \frac{NLD(P_h) - D(P_h) \times \frac{NL(O)}{L(O)+NL(O)}}{\sqrt{D(P_h) \times \frac{NL(O)}{L(O)+NL(O)} \times (1 - \frac{NL(O)}{L(O)+NL(O)})}}$
  $xv_2(P_h) \hookrightarrow N(0,1)$

Si on considère H_1 l'hypothèse alternative à H_0 définie ainsi : "L'ensemble d'objets O est organisé de manière non aléatoire selon une partition P_h en z

classes telles que cette partition représente une cns valide", alors $LM(P_h)$ et $NLD(P_h)$ doivent simultanément exhiber des valeurs exceptionnellement élevées.

Nous pouvons donc maintenant bâtir un test statistique unilatéral droite pour $xv_1(P_h)$ et $xv_2(P_h)$ (nous notons ces tests T_1 et T_2). Nous considérons par la suite qu'une partition constitue une cns valide si et seulement si pour chaque test (T_1 et T_2) l'hypothèse H_1 est acceptée.

Définition 10 ***Classification non supervisée valide***
Une partition P_h est considérée comme valide avec un couple de risques du premier type (α_1,α_2) ssi :

$$xv_1(P_h) = \frac{LM(P_h) - M(P_h) \times \frac{L(O)}{L(O)+NL(O)}}{\sqrt{M(P_h) \times \frac{L(O)}{L(O)+NL(O)} \times (1 - \frac{L(O)}{L(O)+NL(O)})}}$$

$$pv_1(P_h) = 1 - \int_{-\infty}^{xv_1(P_h)} \frac{1}{\sqrt{2\Pi}} e^{\frac{-t^2}{2}} dt \quad (4.12)$$

$$pv_1(P_h) \leq \alpha_1 \Leftrightarrow xv_1(P_h) \geq F^{-1}(1-\alpha_1) \quad (4.13)$$

ET

$$xv_2(P_h) = \frac{NLD(P_h) - D(P_h) \times \frac{NL(O)}{L(O)+NL(O)}}{\sqrt{D(P_h) \times \frac{NL(O)}{L(O)+NL(O)} \times (1 - \frac{NL(O)}{L(O)+NL(O)})}}$$

$$pv_2(P_h) = 1 - \int_{-\infty}^{xv_2(P_h)} \frac{1}{\sqrt{2\Pi}} e^{\frac{-t^2}{2}} dt \quad (4.14)$$

$$pv_2(P_h) \leq \alpha_2 \Leftrightarrow xv_2(P_h) \geq F^{-1}(1-\alpha_2) \quad (4.15)$$

avec F la fonction de distribution de probabilité cumulée inverse de la loi normale centrée réduite

4.2.2.2 Méthodologie

Nous venons de présenter deux tests permettant de décider si oui ou non une partition constitue une cns valide. Cependant, il est clair que pour de nombreux jeux de données plusieurs partitions de l'ensemble des objets (O) de ces jeux de données peuvent être considérées comme des cns valides. Admettons donc que l'on dispose d'un ensemble de cns valides noté ECV. Le problème est alors de déterminer laquelle de ces cns est la cns la plus valide.

La méthodologie que nous proposons permet de résoudre ce problème dans certains cas, et permet toujours de déterminer un sous ensemble ECM de ECV ($ECM \subseteq ECV$) qui inclue toutes les cns candidates au "titre de cns la plus valide". Cette dernière situation correspond au cas pour lequel nous ne sommes pas capables de décider quelle est la cns de ECM la plus valide (si

toutefois il existe une cns plus valide que les autres) mais cependant capables de déterminer un sous ensemble de cns les plus valides. Notre méthodologie permet de plus la visualisation d'un ensemble d'informations concernant la structure de l'ensemble objets, ce qui constitue un outil utile pour l'utilisateur qui doit choisir une cns de ECM (si ECM inclue plus d'une cns).

Nous exposons cette méthodologie sous forme algorithmique (voir algorithme 3 en page 79).

REMARQUE :
Le point **4.** de l'algorithme de la méthodologie (l'extraction de ECM) peut également être réalisé par comparaison directe des couples de valeurs ($xv_1(P_i)$,$xv_2(P_i)$) (cela étant du à la relation de monotonie unissant à la fois $xv_1(P_i)$ et $pv_1(P_i)$ ainsi que $xv_2(P_i)$ et $pv_2(P_i)$). Cette comparaison mène aussi à 4 situations différentes :

- P_j est considérée comme plus valide que P_i ssi
  $(xv_1(P_i) < xv_1(P_j)$ et $xv_2(P_i) < xv_2(P_j))$ ou $(xv_1(P_i) \leq xv_1(P_j)$ et $xv_2(P_i) < xv_2(P_j))$ ou $(xv_1(P_i) < xv_1(P_j)$ et $xv_2(P_i) \leq xv_2(P_j))$
  nous notons cette relation : $P_j < b > P_i$
- P_i est considérée comme plus valide que P_j ssi
  $(xv_1(P_j) < xv_1(P_i)$ et $xv_2(P_j) < xv_2(P_i))$ ou $(xv_1(P_j) \leq xv_1(P_i)$ et $xv_2(P_j) < xv_2(P_i))$ ou $(xv_1(P_j) < xv_1(P_i)$ et $xv_2(P_j) \leq xv_2(P_i))$
  nous notons cette relation : $P_i < b > P_j$
- P_j et P_i sont considérées comme équivalentes du point de vue de la validité ssi $(xv_1(P_j) = xv_1(P_i)$ et $xv_2(P_j) = xv_2(P_i))$
  nous notons cette relation : $P_i < s > P_j$
- P_j et P_i sont considérées comme incomparables du point de vue de la validité ssi $(xv_1(P_j) < xv_1(P_i)$ et $xv_2(P_j) > xv_2(P_i))$ ou $(xv_1(P_j) > xv_1(P_i)$ et $xv_2(P_j) < xv_2(P_i))$
  nous notons cette relation : $P_i <? > P_j$

ECM est alors défini par : $ECM = \bigcup\{P_i \; telle \; que \; \nexists j \in 1..q, P_j < b > P_i\}$

Algorithme 3 Méthodologie pour la détermination de la cns la plus valide (ou de l'ensemble) des cns les plus valides

Données : $EP = \{P_1,..,P_q\}$ un ensemble de q partitions de O

1.Déterminer pour chaque partition P_i les valeurs de leurs statistiques $xv_1(P_i)$ et $xv_2(P_i)$ ainsi que les valeurs $pv_1(P_i)$ et $pv_2(P_i)$ obtenues respectivement pour les tests T_1 et T_2. Ainsi, chaque P_i ($i \in 1..q$) est caractérisée par deux couples de valeurs ($pv_1(P_i)$,$pv_2(P_i)$) et ($xv_1(P_i)$,$xv_2(P_i)$).

2. Fixer α_1 (resp. α_2) seuil sur le risque de rejeter à tort l'hypothèse H_0 (et d'accepter à tort H_1) pour les tests T_1 (resp. T_2) (ces valeurs sont fixées arbitrairement par l'utilisateur, les valeurs typiques sont 0.05, 0.025, 0.01...).

3. Créer l'ensemble de partitions ECV qui correspond à l'ensemble des cns valides (selon T_1 et T_2). ECV est formellement définit de la façon suivante :

$$ECV = \bigcup_{P_i \in EP} (P_i \; telle \; que \; pv_1(P_i) \leq \alpha_1 \; et \; pv_2(P_i) \leq \alpha_2)$$

4. Comparer la validité de deux cns P_i ($i \in 1..q$), P_j ($j \in 1..q, i \neq j$) revient à comparer leur couple de valeurs ($pv_1(P_i)$,$pv_2(P_i)$) and (pv_{1_j},$pv_2(P_j)$).
Cette comparaison mène à 4 situations differentes :

- P_i est considérée plus valide que P_j ssi
  ($pv_1(P_i) < pv_{1_j}$ et $pv_2(P_i) < pv_2(P_j)$) ou ($pv_1(P_i) \leq pv_{1_j}$ et $pv_2(P_i) < pv_2(P_j)$) ou ($pv_1(P_i) < pv_{1_j}$) et ($pv_2(P_i) \leq pv_2(P_j)$),
  nous notons cette relation : $P_i < b > P_j$
- P_j est considérée plus valide que P_i ssi
  ($pv_{1_j} < pv_1(P_i)$ et $pv_2(P_j) < pv_2(P_i)$) ou ($pv_{1_j} \leq pv_1(P_i)$ et $pv_2(P_j) < pv_2(P_i)$) ou ($pv_{1_j} < pv_1(P_i)$ et $pv_2(P_j) \leq pv_2(P_i)$),
  nous notons cette relation : $P_j < b > P_i$
- P_j et P_i sont considérées comme équivalentes du point de vue de la validité ssi $pv_{1_j} = pv_1(P_i)$ et $pv_2(P_j) = pv_2(P_i)$,
  nous notons cette relation: $P_i < s > P_j$
- P_j et P_i sont considérées comme incomparables du point de vue de la validité ssi ($pv_{1_j} < pv_1(P_i)$ et $pv_2(P_j) > pv_2(P_i)$) ou ($pv_{1_j} > pv_1(P_i)$ et $pv_2(P_j) < pv_2(P_i)$),
  nous notons cette relation : $P_i <? > P_j$

5. Extraire l'ensemble ECM des cns les plus valides, qui est formellement défini comme suit : $ECM = \bigcup\{P_i \; telle \; que \; \nexists j \in 1..q, P_j < b > P_i\}$

EXEMPLE : *Nous illustrons la méthodologie sur un exemple synthétique*[8]*. Considérons le jeu de données synthétique introduit préalablement en page 74 (voir tableau 4.3).*

Le tableau 4.4 présente les caractéristiques de chaque partition de l'ensemble des partitions possibles des objets du jeu de données (à l'exception de la partition grossière en une classe et de la partition la plus fine qui possède autant de classes que d'objets du jeu de données).

Si nous fixons $\alpha_1 = \alpha_2 = 0.15$ *nous obtenons alors* $ECV = \{P_6, P_{14}\}$*. La comparaison de la validité de ces 2 cns donne* $ECM = \{P_6\}$*. Remarquons que les partitions* P_6 *et* P_{14} *apparaissent clairement comme les seules cns valides et que nous pouvons aussi considérer que* P_6 *correspond à la cns la plus valide. Nous devons aussi noter que toutes les étapes de la méthodologie peuvent être résumées graphiquement comme le montre les figures 4.2 et 4.3 ; la figure 4.2 (resp. 4.3) correspond au couple* $(1 - pv_1(P_i), 1 - pv_2(P_i))$ *(resp.*$(xv_1(P_i), xv_2(P_i))$*), les lignes formées de tirets de la figure 4.2 correspondent à* $pv_1(P_i) = 1 - \alpha_1 = 0.85$ *et* $pv_2(P_i) = 1 - \alpha_2 = 0.85$*, elles délimitent la zone incluant des cns valides.*

	P_h	M	D	LM	NLD	LD	NLM	xv_1	xv_2	pv_1	pv_2
P_2	{2,3, 4},{1}	12	12	4	7	5	8	-0,298	-0,298	0,383	0,383
P_3	{1,3, 4},{2}	12	12	4	7	5	8	-0,298	-0,298	0,383	0,383
P_4	{1,2, 4},{3}	12	12	4	7	5	8	-0,298	-0,298	0,383	0,383
P_5	{1,2, 3},{4}	12	12	6	9	3	6	0,894	0,894	0,814	0,814
P_6	{1,2}, {3,4}	8	16	7	14	2	1	2,92	2,066	0,998	0,981
P_7	{1,3}, {2,4}	8	16	1	8	8	7	-1,461	-1,033	0,072	0,151
P_8	{1,4}, {2,3}	8	16	1	8	8	7	-1,461	-1,033	0,072	0,151
P_9	{1},{2},{3,4}	4	20	3	14	6	1	1,549	0,693	0,939	0,756
P_{10}	{1},{3},{2,4}	4	20	0	11	9	4	-1,549	-0,693	0,061	0,244
P_{11}	{1},{4},{2,3}	4	20	1	12	8	3	-0,516	-0,231	0,303	0,409
P_{12}	{2},{3},{1,4}	4	20	0	11	9	4	-1,549	-0,693	0,061	0,244
P_{13}	{2},{4},{1,3}	4	20	1	12	8	3	-0,516	-0,231	0,303	0,409
P_{14}	{3},{4},{1,2}	4	20	4	15	5	0	2,582	1,155	0,995	0,876

Légende : $M : M(P_h)$, $D : D(P_h)$, $LM : LM(P_h)$, $NLD : NLD(P_h)$, $NLM : NLM(P_h)$, $LD : LD(P_h)$

TAB. 4.4 –: *Partitions du jeu de données synthétique*

REMARQUES :

- La présentation graphique des résultats les résume parfaitement, et permet une bonne et rapide comparaison de la validité des cns.
- La présentation graphique des résultats permet également une visualisation du type de structure du jeu de données : cela indique quels nombres de classes doivent être considérés comme trop faibles et quels nombres

8. Attention, cet exemple vise essentiellement à illustrer les différentes étapes de la méthodologie, en effet, étant donné le faible nombre d'individus du jeu de données l'approximation normale est ici douteuse...

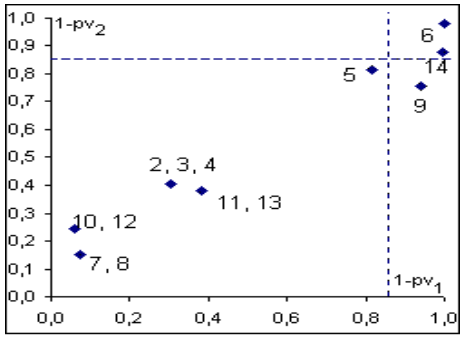


FIG. 4.2 –: *Couples* $(1 - pv_1(P_i)$,$1 - pv_2(P_i))$ *pour chaque partition*

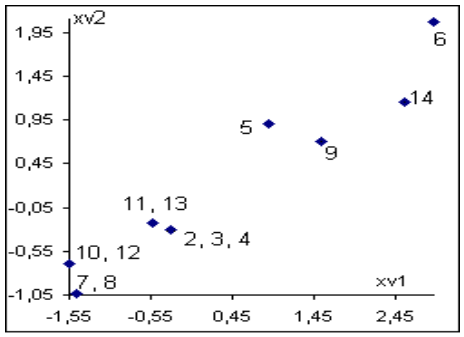


FIG. 4.3 –: *Couples* $(xv_1(P_i)$,$xv_2(P_i))$ *pour chaque partition*

de classes doivent être considérés comme trop élevés. (Cela apporte également d'autres informations que nous aborderons plus tard.)

– Nous utiliserons par la suite la représentation graphique employant les couples $(xv_1(P_i)$,$xv_2(P_i))$ (car ces deux graphiques représentent la même information et nous trouvons ce dernier type de graphique plus clair).

Avant toute autre expérimentation, nous pouvons lister certains avantages de cette méthodologie :

– présentation intuitive et graphique des résultats ;
– évaluation graphique simultanée de l'homogénéité interne des classes et de l'hétérogénéité entre classes de la cns, contrairement à la plupart des méthodologies classiques qui considèrent soit
  – une seule de ces deux notions,
  – ou, ces deux notions combinées au sein d'un unique indice,
  – ou encore ces deux notions de manière séparée.

- permet la comparaison de cns indépendamment de :
  - leurs nombres de classes (pas de tendance associée au nombre de classes),
  - l'algorithme (et les paramètres de l'algorithme) mis en œuvre pour les mettre à jour...
- une seule passe sur les données et coût calculatoire relativement faible (particulièrement dans le cas des données catégorielles) ;
- permet une caractérisation de la structure des données (ce point est complété ultérieurement).

4.2.2.3 Expérimentations

Nous avons présenté notre méthodologie sur un exemple synthétique, nous poursuivons maintenant les expérimentations et décrivons les résultats de ces dernières.

4.2.2.4 Expérimentations sur le jeu de données Small Soybean Disease

Nous utilisons tout d'abord le jeu de données "small soybean diseases" (provenant de la collection de l'Université de Californie à Irvine (UCI) [MM96]) classiquement adopté dans la littérature traitant de la cns. Ce jeu de données possèdent 47 objets (des graines de soja) décrits par 35 variables catégorielles ; de plus, à chaque objet est associé une des 4 étiquettes suivantes (qui correspondent à des pathologies) : diaporthe-stem-canker(notée D1), charcoal-rot (notée D2), rhizoctonia-root-rot(notée D3), phytophthora-rot(notée D4). Exceptée D4 qui est associée à 17 objets, toutes les autres pathologies sont associées à 10 objets chacune (voir page 217 pour de plus amples informations). Les expérimentations menées visent à montrer la puissance de la méthodologie présentée pour déterminer et comparer la validité de cns ainsi que pour déterminer la structure du jeu de données.

Description Nous avons mené deux expérimentations différentes :

- **Experience #1.** Nous avons utilisé KEROUAC (méthode de cns pour données catégorielles présentée dans [JN03c] [JN03a] et au chapitre 3) avec des paramètrages différents (des valeurs différentes pour le facteur de granularité) de manière à générer des cns possédant des nombres de classes différents (les paramètres ont été fixés de manière à obtenir des cns en 2, 3, 4, 5, 6, 7, 8, 9, 10 classes). (Cette méthode utilise une variation du Nouveau Critère de Condorcet [Mic97] et essaie donc de trouver une cns minimisant simultanément le nombre de dissimilarités entre objets de même classe et le nombre de similarités entre objets appartenant à des classes différentes). Les résultats sont exposés dans le tableau 4.5 (page 83). Ce tableau donne les informations suivantes :
  - le nombre de classes (#Cl.) de chaque partition (cns),

- la valeur à laquelle a été fixée le facteur de granularité (α) afin d'obtenir la partition et ce pour chaque partition,
- les valeurs de xv_1 et xv_2 pour chaque partition,
- la valeur du critère à minimiser sous-jacent à la méthode K-Modes (QKM) pour chaque partition,
- la valeur du critère à minimiser sous-jacent à la méthode KEROUAC (NCC) pour chaque partition,
- le taux de correction (T.C.) de chaque partition par rapport au concept "pathologie".

n°	#Cl.	α	xv_1	xv_2	QKM	NCC	T.C.
1a	2	1.5	18.02	24.98	310	13991	57.45%
2a	3	2	28.39	23.57	236	17009	78.72%
3a	4	3	31.41	18.17	199	19739	100%
4a	5	3.5	31.24	17.36	188	20055	100%
5a	6	4	29.86	13.74	173	21429	100%
6a	7	5	28.78	12.04	154	22055	100%
7a	8	5.1	27.43	10.41	141	22640	100%
8a	9	5.15	26.04	8.85	132	23206	100%
9a	10	5.3	25.86	8.66	128	23277	100%

TAB. 4.5 –: *Résultats de l'Expérience #1*

- **Experience #2.** Nous avons utilisé la méthode des K-Modes [Hua97] (qui est une extension des K-Means pour les données catégorielles) afin de réaliser 9 séries de cns. Chaque série correspond à 10 cns possédant un même nombre fixé de classes. (Nous avons réalisé des séries de cns avec les mêmes valeurs de paramètres car cette méthode est fortement sensible à l'initialisation contrairement à KEROUAC qui fournit toujours le même résultat pour une valeur donnée du facteur de granularité). Le nombre de classes (#Cl.) a été fixé respectivement à 2, 3, 4, 5, 6, 7, 8, 9, 10. Le tableau 4.6 (page 84) récapitule les résultats pour la "meilleure" partition de chaque série. (On désigne par "meilleure" partition, la partition possédant la valeur la plus faible pour QKM, c'est à dire la meilleure partition au sens du critère sous-jacent à la méthode des K-Modes). Le tableau 4.7 récapitule l'ensemble des résultats pour cette expérience en se contentant toutefois de ne décrire qu'une seule fois une même partition même si plusieurs processus de cns ont mené à une même partition. Ces tableaux donnent les informations suivantes pour chaque cns :
  - le nombre de classes (#Cl.) de chaque partition (cns),
  - les valeurs de xv_1 et xv_2 pour chaque partition,
  - la valeur du critère à minimiser sous-jacent à la méthode K-Modes (QKM) pour chaque partition,

- la valeur du critère à minimiser sous-jacent à la méthode KEROUAC (NCC) pour chaque partition,
- le taux de correction (T.C.) de chaque partition par rapport au concept "pathologie".

n°	#Cl.	xv_1	xv_2	QKM	NCC	T.C.
1x	2	18.96	21.22	308	15563	57.45%
2x	3	28.39	23.57	236	17009	78.72%
3x	4	31.41	18.17	199	19739	100%
4x	5	29.47	14	177	21265	100%
5x	6	28.58	12.6	165	21776	100%
6x	7	26.17	10.42	149	22486	97.87%
7x	8	25.78	9.57	141	22843	100%
8x	9	25.94	9.42	130	22932	100%
9x	10	24.28	8.13	126	23367	97.87%

TAB. 4.6 –: *Meilleurs résultats de l'expérience #2*

Les résultats concernant les valeurs de xv_1, xv_2, T.C., #Cl., QKM, NCC pour chaque cns (partition) de la première expérimentation ainsi que pour chaque cns de la seconde expérimentation) sont graphiquement présentés sur les figures 4.4 (page 86) et 4.5 (page 87). Ces figures reprennent l'ensemble des informations données dans les tableaux 4.5, 4.6 et 4.7.

Analyse des Résultats La question qui se pose après avoir réalisé ces expérimentations est de déterminer laquelle de toutes les cns obtenues est la plus valide. Nous procédons ici à une analyse segmentée en 3 points :

- nous analysons tout d'abord les résultats de l'expérience 1, le problème est alors de déterminer laquelle des cns obtenues par l'intermédiaire de la méthode KEROUAC peut être considérée comme la plus valide ;
- puis, nous nous penchons sur le même problème pour l'expérience 2 (laquelle des cns obtenues par l'intermédiaire de la méthode K-Modes peut être considérée comme la plus valide ?) ;
- le troisième problème abordé est celui de la détermination de la meilleure des cns, que celles ci soient obtenues grâce aux K-Modes ou à KEROUAC.

Le choix de cette segmentation de l'analyse en 3 points est ici motivé par le désir d'exposer les divers intérêts et avantages de notre méthode dans différentes situations d'évaluation/comparaison de la validité de cns :

- lorsque l'ensemble des cns est obtenue par utilisation d'une unique méthode (points 1 et 2);
- la comparaison de nos critères et méthodologie avec l'utilisation de critères internes au sein d'un mode d'évaluation relatif, et ce, que le critère

n°	#Cl.	xv_1	xv_2	QKM	NCC	T.C.	n°	#Cl.	xv_1	xv_2	QKM	NCC	T.C.
1b	2	18,02	24,98	310	13991	57,45%	6c	7	26,95	11,07	155	22278	100%
1c	2	20,81	21,53	309	15983	57,45%	6d	7	29,51	15,68	180	20522	100%
1d	2	18,14	21,82	311	15083	57,45%	6e	7	26,6	10,92	157	22302	100%
1e	2	18,96	21,22	308	15563	57,45%	6f	7	25,35	10,14	156	22521	100%
1f	2	18,13	23,84	310	14375	57,45%	6g	7	25,92	10,4	157	22464	100%
2a	3	16,6	21,92	297	14563	57,45%	6h	7	27,54	11,69	158	22063	100%
2b	3	22,09	15,32	270	19221	76,6%	6i	7	28,2	14,05	168	21061	100%
2c	3	28,39	23,57	236	17009	78,72%	6j	7	27,49	11,88	161	21970	100%
2d	3	21,9	15,32	271	19173	78,72%	7a	8	25,83	10,96	155	22183	97,87%
3a	4	26,74	21,15	226	17624	78,72%	7b	8	26,6	11,25	158	22148	97,87%
3b	4	19,59	11,05	246	20969	78,72%	7c	8	26,12	10,32	152	22528	100%
3c	4	26,98	21,38	223	17581	78,72%	7d	8	25,74	9,6	144	22826	100%
3d	4	26,96	21,36	224	17583	78,72%	7e	8	25,78	9,57	141	22843	100%
3e	4	28,5	16,08	202	20183	89,36%	7f	8	25,66	10,22	150	22520	100%
3f	4	28,5	16,08	202	20183	89,36%	7g	8	27,63	13,57	162	21200	100%
3g	4	29,84	16,84	199	20065	95,74%	7h	8	26,72	11,3	154	22140	100%
3h	4	29,98	16,91	199	20053	95,74%	7i	8	26,16	10,46	149	22467	100%
3i	4	31,41	18,17	199	19739	100%	7j	8	26,83	10,43	144	22560	100%
4a	5	25,66	19,47	210	18101	78,72%	8a	9	24,98	9,44	145	22815	97,87%
4b	5	26,48	20,7	219	17756	78,72%	8b	9	26,23	9,53	131	22914	100%
4c	5	28,12	14,86	190	20676	95,74%	8c	9	25,46	11,49	156	21871	100%
4d	5	29,85	16,24	190	20325	100%	8d	9	27,43	10,93	145	22402	100%
4e	5	29,47	14	177	21265	100%	8e	9	27	10,8	146	22411	100%
4f	5	29,99	16,32	188	20313	100%	8f	9	25,94	9,42	130	22932	100%
4g	5	30,29	16,44	184	20306	100%	8g	9	23,99	8,44	143	23187	100%
4h	5	28,67	13,62	181	21327	100%	8h	9	25,03	9,57	142	22754	100%
5a	6	26,69	13,22	174	21219	95,74%	8i	9	25	9,8	150	22640	100%
5b	6	27,4	13,14	175	21368	95,74%	8j	9	26,19	9,98	137	22699	100%
5c	6	28,12	13,85	177	21143	97,87%	9a	10	24,28	8,13	126	23367	97,87%
5d	6	27,44	12,1	169	21858	97,87%	9b	10	24,53	8,3	132	23313	97,87%
5e	6	28,12	12,61	165	21713	97,87%	9c	10	24,6	8,32	129	23309	100%
5f	6	28,54	12,93	175	21624	97,87%	9d	10	23,74	8,7	139	23028	100%
5g	6	28,58	12,6	165	21776	100%	9e	10	24,32	8,47	132	23207	100%
5h	6	30,04	16,05	177	20440	100%	9f	10	24,5	8,37	127	23275	100%
5i	6	29,81	15,54	177	20629	100%	9g	10	24,35	8,16	127	23363	100%
5j	6	29,57	15,8	182	20480	100%	9h	10	25,06	8,65	127	23203	100%
6a	7	26,17	10,42	149	22486	97,87%	9i	10	25,57	9,74	135	22739	100%
6b	7	27,13	10,97	152	22344	100%	9j	10	22,77	7,35	133	23590	100%

TAB. 4.7 –: *Résultats de l'expérience #2*

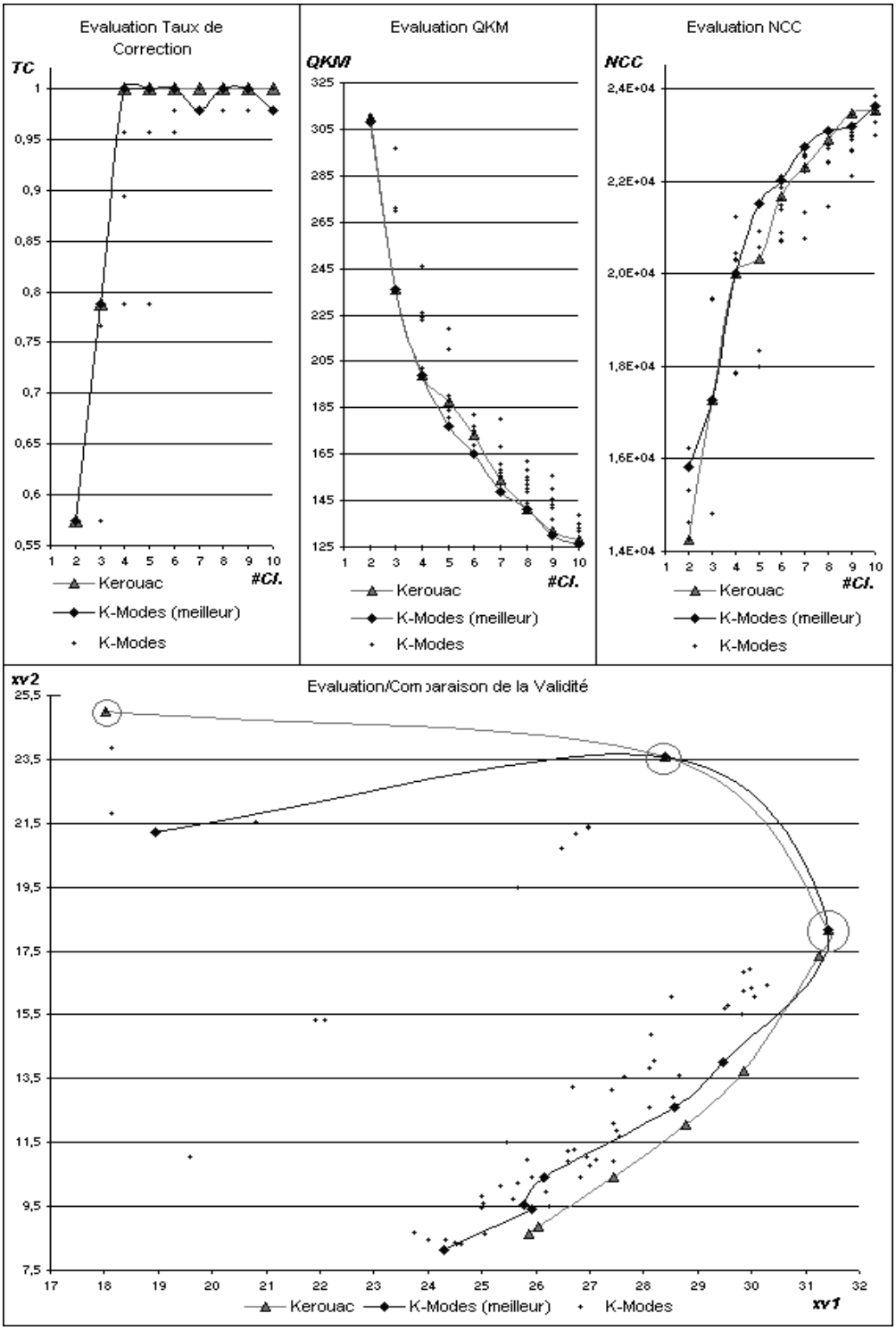


FIG. 4.4 –: *Eléments pour l'évaluation de la validité des cns sur le jeu de données Soybean Disease*

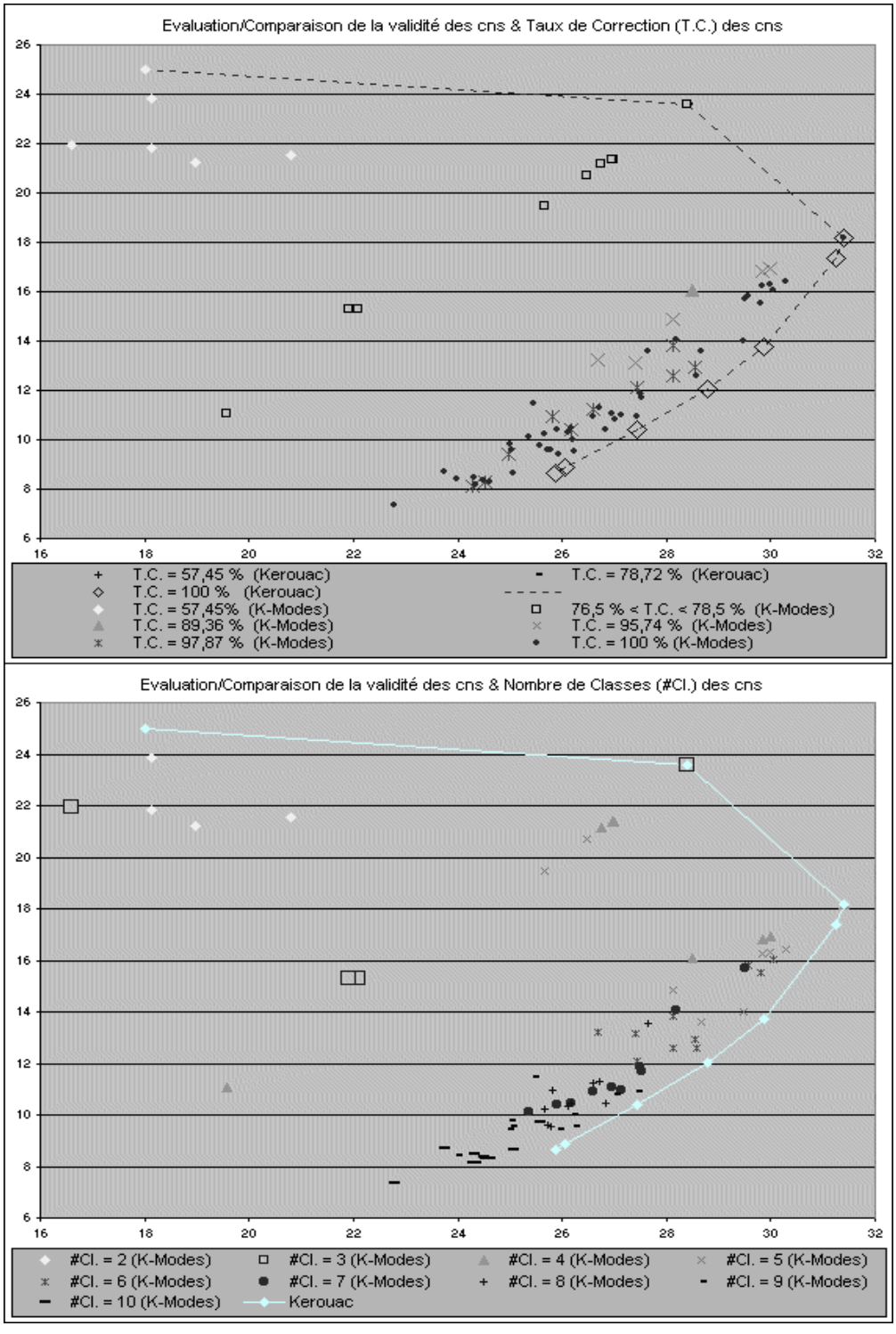


FIG. 4.5 –: *Eléments pour l'évaluation de la validité des cns sur le jeu de données Soybean Disease*

possède (QKM) ou non (NCC) une tendance selon le nombre de classes (points 1 et 2) ;
- la comparaison de nos critères et méthodologie avec l'utilisation d'un critère externe (T.C.) (points 1, 2 et 3) ;
- l'évaluation/comparaison de la validité de cns obtenues par des méthodes différentes (point 3).

Analyse de l'expérience #1 Pour déterminer quelle cns peut être considérée comme la plus valide nous pouvons utiliser plusieurs modes d'évaluation :

Mode d'évaluation relatif avec utilisation d'un critère interne : Le premier problème est le choix du critère à employer (le mode d'évaluation relatif a été choisi car son coût calculatoire est relativement faible et proche de celui de notre méthode). Nous choisissons ici les critères NCC et QKM car ils correspondent aux critères des méthodes de cns employées dans les expériences 1 et 2. Si l'on considère le critère :

- **NCC** : nous pouvons observer que pour cette expérience la valeur de ce critère croît avec le nombre de classes, or ce critère est indépendant du nombre de classes et doit être minimisé, on peut donc en conclure que *la cns 1a (en deux classes) apparaît comme la plus valide* (en fait une cns en une classe mènerait à une valeur du NCC plus faible encore, ce qui semble vouloir signifier qu'il n'existe pas de structure dans les données)
- **QKM** : ce critère doit être minimisé et est dépendant du nombre de classes de la cns (il tend à décroître avec un nombre croissant de classes). On observe ici que cette tendance à la décroissance est bien respectée. La méthodologie d'évaluation veut que l'on cherche alors un changement local marquant dans la courbe, or ce changement n'apparaît pas vraiment, *on ne peut donc rien conclure* de satisfaisant avec cette méthodologie quant à la cns qui doit être la plus valide.

Mode d'évaluation relatif avec utilisation d'un critère externe : Nous utiliserons ici une version ultra simpliste des critères externes basés sur la théorie de l'information : le taux de correction de la cns par rapport à une structure prédéfinie, cette structure étant la partition des objets impliquée par le concept comestibilité, le critère est donc le critère de correction par rapport au concept comestibilité. On observe que la correction croît simultanément avec le nombre de classes (entre 2 et 4 classes) puis se stabilise à 1 (100%) pour un nombre de classes supérieur ou égal à 4. On peut en conclure que *la cns la plus valide est donc celle en 4 classes* puisqu'elle traduit parfaitement le concept comestibilité avec un nombre de classes le plus restreint possible.

Notre méthodologie d'évaluation : Remarquons tout d'abord, que toutes les partitions présentées correspondent à des cns valides selon la définition 10 avec $\alpha_1 = \alpha_2 = 0.001$ (conséquemment aucune ligne n'est tracée sur le

graphique pour délimiter la zone incluant les cns valides). L'ensemble des meilleures cns est ici constitué par les cns 1a, 2a, 3a (i.e. en 2, 3 et 4 classes, voir tableau 4.5).
Les analyses des figures 4.4, 4.5 révèlent également :

- La structure des données par rapport à l'hétérogénéité entre classes : les valeurs de xv_2 sont organisées selon le nombre de classes de la cns. En fait, il apparaît que plus une cns possède de classes, moins significative est sa valeur pour xv_2.

  Une analyse plus précise de ces valeurs montre que :

  - il n'y a pas de forte différence entre les cns en 1a et 2a (i.e en 2 et 3 classes) de ce point de vue (par opposition au point de vue de l'homogénéité des classes (i.e. des valeurs de xv_1))
  - la décroissance en significativité des valeurs xv_2 ralentit lorsque le nombre de classes est élevé.

  Nous pouvons, à partir de ces observations, conclure que l'hétérogénéité entre classes caractérise évidemment la structure du jeu de données, mais que cet aspect n'est pas très fort puisque la relation entre le nombre de classes et xv_2 est monotone (la monotonicité ne traduit pas une caractéristique structurelle spéciale pour les données car on peut considérer normal que les cns possédant un nombre de classe élevé possèdent également une hétérogénéité entre classes moins significative). De surcroît, dans la mesure où la valeur pour xv_1 est beaucoup plus significative pour la cns en 3 classes que pour la cns en 2 classes, (et que par ailleurs leurs valeurs pour xv_2 sont quasi équivalentes) nous pouvons conclure que la cns en 3 classes est plus valide que la cns en 2 classes.
- La structure des données par rapport à l'homogénéité des classes : les valeurs de xv_1 sont organisées de manière non monotone par rapport au nombre de classes des cns. En effet, les valeurs de xv_1 croissent dans un premier temps pour des nombres de classes croissants (jusqu'à #Cl.=4) puis décroissent. Nous pouvons ainsi conclure de ces observations que l'homogénéité entre classes caractérise évidemment la structure des données, et que cet aspect est assez fort puisque la relation entre le nombre de classes et xv_1 est non-monotone (cette non-monotonicité est effectivement intéressante car elle s'oppose à la corrélation naturelle entre nombre de classes et significativité de l'homogénéité entre classes).
- Ces deux premiers points indiquent que la cns la plus valide est soit la cns 2a ou 3a (i.e. la cns en 3 classes ou en 4 classes). Etant donné que nous avons observé que l'aspect homogénéité des classes caractérise plus fortement la structure des données, nous pouvons conclure que *la cns 2a est la plus valide*. Enfin, si la cns 4a (en 5 classes) n'a pas été sélectionnée parmi l'ensemble des cns les plus valides, nous pourrions toutefois réviser ce choix car ses valeurs pour xv_1 et xv_2 sont proches de celles de la cns en 4 classes. (En définitive, si l'on recherchait une cns valide avec le

souhait qu'elle inclue plus de 4 classes, choisir cette cns ne constituerait pas une mauvaise idée.)

Analyse de l'Expérience #2 Nous analysons maintenant les résultats obtenus lors de l'expérience #2 (cns obtenues avec la méthode des k-modes). Afin de déterminer laquelle des cns peut être considérée comme la plus valide, nous pouvons utiliser les modes d'évaluation considérés pour l'analyse des résultats de l'expérience #1 :

Mode d'évaluation relatif avec utilisation d'un critère interne :
- **NCC** : les observations et donc les conclusions sont les mêmes que pour l'expérience #1, on conclut donc que *la cns 1b (en 2 classes) apparaît comme la plus valide*
- **QKM** : mêmes observations et donc conclusions que pour l'expérience #1, *on ne peut donc rien conclure* de satisfaisant avec cette méthodologie.

Mode d'évaluation relatif avec utilisation d'un critère externe : observations quasi-similaires à celles de l'expérience #1 et donc mêmes conclusions, *on conclut que la cns la plus valide est donc la cns 3i (en 4 classes)*.

Notre méthodologie d'évaluation : Tout d'abord, toutes les partitions obtenues correspondent à des cns valides selon la définition 10 avec $\alpha_1 = \alpha_2 = 0.001$. L'ensemble des meilleures cns est ici constitué par les cns 1b, 2c, 3i (en 2, 3 et 4 classes ; entourées sur la figure 4.4).
Si nous considérions chaque série de cns (correspondant à un même nombre de classes), sélectionnions parmi ses cns la cns la plus valide, et tracions la courbe joignant les cns sélectionnées selon l'ordre sur le nombre de classes des cns sélectionnées, nous obtiendrions alors une courbe ressemblant à celle de l'expérience #1. (Notons bien qu'il s'agit d'une courbe non tracée et qu'il ne s'agit pas de la courbe nommée K-Modes (meilleur) qui joint les meilleures cns de chaque série au sens du critère QKM.) De plus, les cns 1b, 2c, 3i correspondent respectivement aux cns 1a, 2a, 3a de l'expérience #1. Cela permet les mêmes conclusions que pour l'expérience #1 :
- l'hétérogénéité entre classes caractérise évidemment la structure du jeu de données, mais cet aspect n'est pas très fort puisque la relation entre le nombre de classes et xv_2 est monotone, alors que pour l'homogénénité interne des classes, la relation de monotonie est brisée ce qui montre qu'elle caractérise fortement la structure des données.
- l'ensemble de ces points mènerait à la sélection de la cns en *4 classes comme cns la plus valide*. (notons que cette cns est identique à la cns en 4 classes obtenue par KEROUAC).

Analyse combinée des Expériences #1 et #2 L'analyse simultanée des résultats des 2 expériences peut permettre de choisir la cns la plus valide parmi

un ensemble de cns résultant de 2 méthodologies différentes. Afin de déterminer laquelle des cns peut être considérée comme la plus valide nous pouvons utiliser les modes d'évaluation considérés pour les expériences #1 et #2 :

Mode d'évaluation relatif avec utilisation d'un critère interne :

- **NCC** : la cns possédant la plus faible valeur pour le critère NCC est la cns en 2 classes obtenue en utilisant KEROUAC, on conclut donc que *la cns 1a (en 2 classes) obtenue avec* KEROUAC *ou encore la cns 1b (qui est la même mais obtenue cette fois avec les K-Modes) apparaît comme la plus valide*
- **QKM** : mêmes observations et donc conclusions que pour les expériences #1 et #2, *on ne peut donc rien conclure* de satisfaisant avec cette méthodologie.

Mode d'évaluation relatif avec utilisation d'un critère externe : observations similaires à celles des expériences #1 et #2, *on conclue que la cns la plus valide est donc celle en 4 classes obtenue avec* KEROUAC *(cns 3a) qui est également l'une des cns obtenues avec les K-Modes (cns 3i).*

Notre méthodologie d'évaluation : On peut reprendre les observations des deux expériences précédentes, celles ci mènent à la conclusion que *la cns la plus valide est celle en 4 classes obtenue avec* KEROUAC *(cns 3a) qui est également l'une des cns obtenues avec les K-Modes (cns 3i).*

Récapitulatif, Confrontation des Analyses des Résultats Les résultats des diverses analyses sont regroupés dans le tableau 4.8. Il apparaît clairement que les conclusions apportées par notre méthode sont en adéquation totale avec celles obtenus par l'utilisation d'un critère externe, cela semble donc signifier que les résultats de notre méthode sont excellents, contrairement à ceux obtenus par les critères internes. En effet, pour le critère QKM (qui correspond à une version spéciale du critère SSE), l'utilisation du mode d'évaluation relatif ne mène à aucun résultat car l'interprétation graphique des résultats est impossible. Les seules informations que l'on peut tirer de ce critère concerneraient la recherche de la cns la plus valide pour un nombre de classes fixé. (Notons d'ailleurs que l'utilisation de notre méthodologie pour ce type de recherche mènerait dans la majorité des cas à l'obtention de résultats similaires à ceux obtenus avec ce critère). Concernant, l'utilisation du critère NCC les résultats fournis ne semblent pas vraiment valables puisqu'ils contredisent les connaissances que l'on a sur les données (les différentes pathologies associées aux graines), et de plus, ils tendent à signifier l'absence de toute structure dans les données. Enfin, les figures 4.4 et 4.5 permettent de conclure, que la cns la plus valide peut être obtenue aussi bien avec les K-Modes qu'avec KEROUAC, mais que par contre, à nombre de classes égales, les cns obtenues par KEROUAC sont le plus souvent plus valides que celles obtenues par les K-Modes, ce dernier élément est intéressant pour la comparaison globale de la

validité des structures fournies par deux méthodes différentes pour un même jeu de données.

Expérience	Mode d'Evaluation Comparaison de la validité	cns la plus valide	#Cl.	Méthode de cns utilisée
Expérience #1	Mode Relatif + NCC	1a	2	KEROUAC
Expérience #1	Mode Relatif + QKM	aucune		
Expérience #1	Critère Externe (T.C.)	4a	4	KEROUAC
Expérience #1	Notre Méthodologie	4a	4	KEROUAC
Expérience #2	Mode Relatif + NCC	1b	2	K-Modes
Expérience #2	Mode Relatif + QKM	aucune		
Expérience #2	Critère Externe (T.C.)	3i	4	K-Modes
Expérience #1	Notre Méthodologie	3i	4	K-Modes
Expérience #1 & #2	Mode Relatif + NCC	1a ou 1b	2	KEROUAC ou K-Modes
Expérience #1 & #2	Mode Relatif + QKM	aucune		
Expérience #1 & #2	Critère Externe (T.C.)	3a ou 3i	4	KEROUAC ou K-Modes
Expérience #1 & #2	Notre Méthodologie	3a ou 3i	4	KEROUAC ou K-Modes

TAB. 4.8 –: *Récapitulatif des Analyses des Résultats*

4.2.3 Expériences sur le jeu de données Mushrooms

Nous avons ensuite utilisé les données "Mushrooms" (provenant également de la collection de l'UCI [MM96]) un autre jeu de données classiquement utilisé. Ce jeu de données inclut les descriptions d'échantillons de 23 espèces de champignons des familles Agaricus et Lepiota. Il inclut 8124 champignons différents décrits par 22 variables catégorielles. Chaque champignon est de plus identifié comme comestible ou vénéneux. (voir page 217 pour de plus amples informations sur ce jeu de données)

4.2.3.1 Description

Nous avons mené les mêmes expériences que celles précédemment décrites pour le jeu de données Soybean Disease, mais en employant des paramètres différents pour KEROUAC et pour les K-modes de manière à obtenir des cns en 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, et 25 classes. Les résultats sont présentés sur les figures 4.6 et 4.7 (toutes les partitions obtenues correspondant à des cns valides selon la définition 10 avec $\alpha_1 = \alpha_2 = 0.001$; conséquemment, aucune ligne n'est tracée pour délimiter la zone incluant les cns valides).

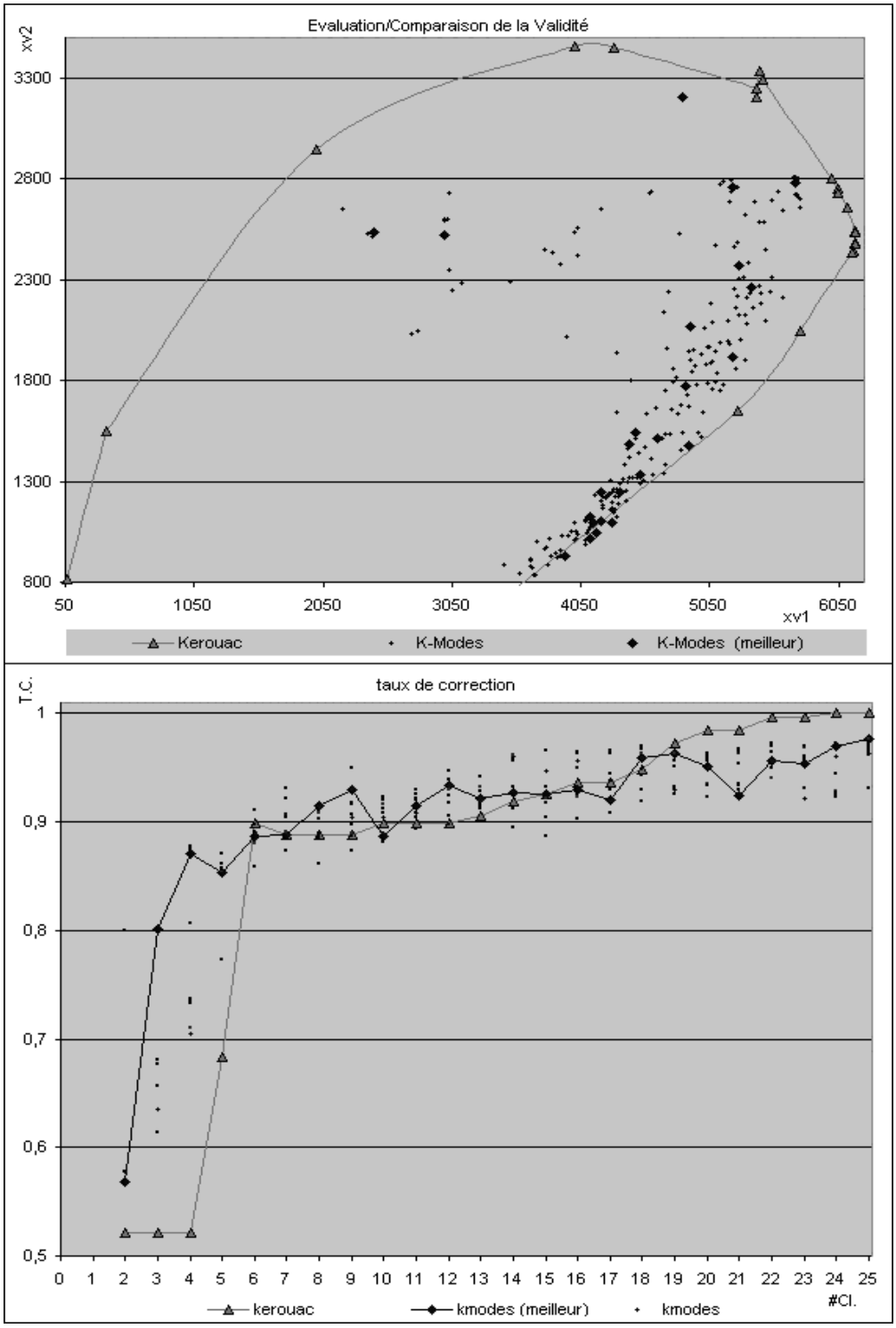


FIG. 4.6 –: *Divers éléments pour l'évaluation de la validité des cns sur le jeu de données Mushrooms*

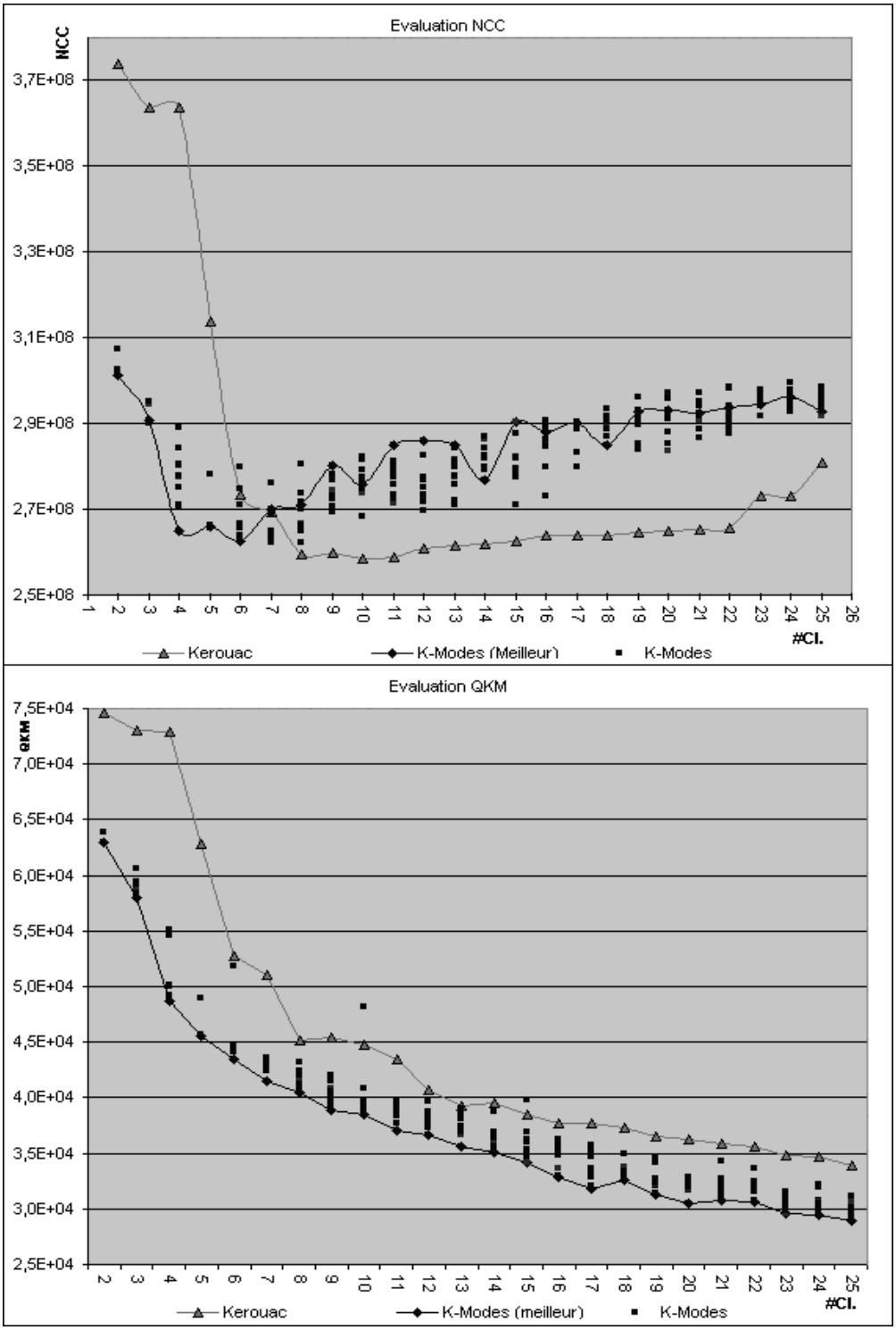


FIG. 4.7 –: *Divers éléments pour l'évaluation de la validité des cns sur le jeu de données Mushrooms*

4.2.3.2 Analyse des Résultats

Nous ne fournissons pas ici une analyse complète des résultats. Les cns les plus valides sont celles possédant 6, 10, 15 et 20 classes avec KEROUAC qui correspondent également à des cns obtenues en utilisant les k-modes. Une analyse complète mènerait à sélectionner la cns en 20 classes qui possède un taux de correction par rapport au concept comestibilité de 98.42 %. Pour cela nous considérons que, comme pour le jeu de données Small Soybean Diseases, les données sont fortement caractérisées par l'aspect homogénéité interne des classes et plus faiblement par l'aspect hétérogénéité entre classes. Cela n'apparaît pas de manière totalement évidente sur le graphique car pour ces deux aspects la relation de monotonie est brisée mais il nous semble cependant que cela est plus marquant pour l'aspect homogénéité interne que pour l'aspect hétérogénéité entre classes dans la mesure où, pour ce dernier aspect, la cassure intervient pour un nombre de classes égal à 5 qui nous semble très faible.

Notons également que quelle que soit la méthode utilisée (K-Modes ou KEROUAC) notre méthodologie mène au choix de la même cns. Concernant les cns parfaitement correcte du point de vue de la correction par rapport au concept comestibilité, la cns en 24 classes obtenue par KEROUAC est celle possédant le moins de classes, et nous pouvons remarquer que son niveau de validité est très proche de celui de la cns la plus valide. Enfin une comparaison des deux méthodes par l'intermédiaire de notre méthodologie mènerait à la conclusion que sur ce jeu de données, KEROUAC semble fournir des résultats plus valides.

L'utilisation du critère NCC mènerait au choix de la cns en 10 classes obtenue par KEROUAC (si l'on ne considérait que les cns obtenues par les K-Modes, on choisirait alors la "meilleure" cns en 6 classes). Ainsi, l'utilisation du critère NCC peut permettre de déterminer une cns plus valide et ce quelle que soit la méthode utilisée (il est toutefois clair que la méthode KEROUAC verra ses résultats privilégiés), cependant ce choix de cns semble peu conforme aux connaissances que l'on possède (concept comestibilité).

L'utilisation du critère QKM ne permet pas le choix d'une cns apparaissant plus valide que les autres si l'on ne sépare pas les résultats des 2 méthodes employées. Ainsi, pour les cns obtenues par KEROUAC, on observe un "coude" dans la représentation graphique pour un nombre de classes valant 8, on choisirait alors la cns en 8 classes. Par contre pour les cns obtenues grâce aux K-Modes repérer le coude semble très hasardeux, la prise de décision s'avère ici très difficile... Comme les cns obtenues par les K-Modes sont les plus valides au sens du critère QKM on ne pourra déterminer la cns la plus valide parmi l'ensemble de toutes les cns (la seule indication est que cette dernière a été obtenue grâce aux K-Modes, ce qui peut apparaître normal vu la relation unissant le critère QKM et la méthode des K-Modes).

Ainsi, l'utilisation du critère QKM peut permettre de déterminer une cns plus valide dans le cas des cns obtenues par la méthode KEROUAC mais ne semble ni permettre la comparaison des cns obtenues par des méthodes dif-

férentes ni la détermination de la cns la plus valide pour les résultats obtenus pour les K-Modes... De plus le choix opéré pour les cns issues de l'utilisation de KEROUAC ne semble pas des plus judicieux.

De manière globale il nous semble donc que l'application de notre méthodologie soit le meilleur choix que l'on puisse faire pour ce jeu de données.

Les expérimentations menées sur les jeux de données Mushrooms et Soybean Diseases indiquent clairement que la méthodologie que nous proposons permet toujours de déterminer une cns apparaissant comme la plus valide, et que les résultats fournis sont en presque parfaite adéquation avec les connaissances dont on dispose sur les données. De plus, aucun surcoût calculatoire n'est impliqué...

4.2.4 Résumé et Informations Supplémentaires

Nous venons de présenter deux critères et une méthodologie pour l'évaluation et la comparaison de la validité de cns. Ses principaux attraits sont :

- une évaluation simultanée et séparée de l'homogénéité des classes et de l'hétérogénéité entre classes contrairement aux approaches classiques qui considèrent soient une seule de ces deux notions, soit une évaluation de ces deux notions au sein d'un critère les combinant ;
- sa capacité à traiter des cns obtenues par différentes méthodes ;
- sa capacité à traiter des cns ayant des nombres de classes différents ;
- la visualisation relativement clair de ses résultats ;
- elle permet la caractérisation visuelle de la structure sous-jacente aux données contrairement aux méthodes existantes (point développé plus tard) ;
- son coût calculatoire associé relativement faible et la non-utilisation de la méthode de Monte-Carlo ;
- sa capacité à traiter des types de données différents et hétérogènes (en utilisant différents types de fonctions $Lien$).

La caractérisation visuelle de la structure sous-jacente aux données est en effet possible par l'intermédiaire de la visualisation de la validité de différentes cns possédant des nombres de classes différents. Par exemple, si nous utilisons la méthode KEROUAC, nous pouvons tracer la courbe joignant les cns selon l'ordre croissant sur le nombre de classes (voir les figures 4.4, 4.5, 4.6), et nous pouvons ensuite associer cette courbe à l'un des 3 types de structures suivants (voir figure 4.8[9]) :

- Type A : Structures impliquant une cns plus valide que toute les autres, i.e. il existe une cns possédant l'homogénéité des classes la plus significative et l'hétérogénéité entre classes la plus significative. Ce type de

9. sur ces figures les portions de courbes incluant les cns les plus valides sont comprises entre les flèches $\downarrow$ et $\leftarrow$

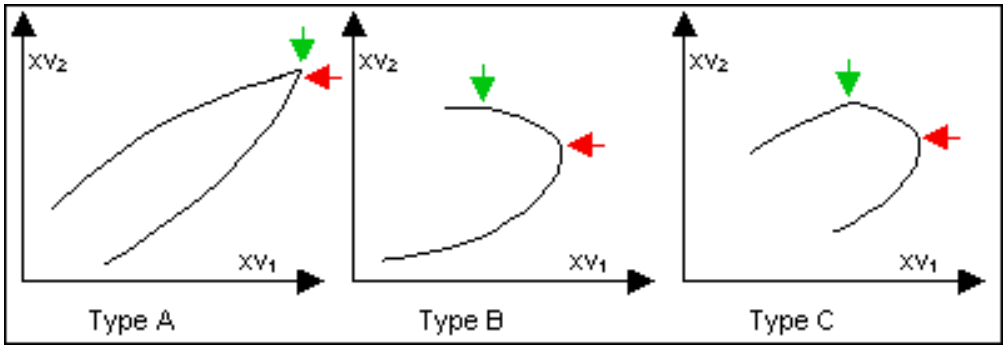


FIG. 4.8 –: *Différents Types de Structures*

structure est à la fois fortement caractérisé par l'homogénéité des classes et l'hétérogénéité entre classes.

- Type B : Structures impliquant un ensemble de cns considérées comme les plus valides et caractérisées fortement par l'homogénéité des classes, i.e. il existe plusieurs cns différentes pouvant être considérées comme les plus valides et caractérisées essentiellement par l'homogénéité des classes. (Dans ce cas, nous pensons que la cns la plus valide est celle exhibant l'homogénéité des classes la plus significative)
- Type C : Structures impliquant un ensemble de cns considérées comme les plus valides et caractérisées fortement à la fois par l'homogénéité des classes et l'hétérogénéité entre classes. (Dans ce cas, nous pensons que la cns pouvant être considérée comme la cns la plus valide est soit celle exhibant l'hétérogénéité entre classes la plus significative, soit celle exhibant l'homogénéité des classes la plus significative.).

Un ensemble de tests réalisé sur 15 jeux de données issus de la collection de l'université de Californie à Irvine [MM96]) [10] illustre ces assertions.

Les résultats de ces tests sont présentés sur les figures 4.9, 4.10, 4.11, 4.12, 4.13 (la forme générale de la courbe permettant la caractérisation de la structure du jeux de données est dessinée en rouge sur ces graphiques).

On peut ainsi remarquer que :

- le jeu de données BREAST possède une structure de type A ;
- les jeux de données CANCER (figure 4.9), GERMAN (figure 4.13), Mushrooms (figure 4.6) possèdent une structure de type C ;
- les jeux de données restant (figures 4.9, 4.10, 4.11, 4.12, 4.13), et le jeu de données Soybean Disease (figure 4.4) possèdent une structure de type B.

10. Ces jeux de données sont les jeux : CANCER, HOUSE-VOTES84 (noté HVOTES), CONTRACEPTION (noté CONTRA.), SPAM, MONKS 3, CAR, NURSERY, FLAGS, ION, WINE, PIMA, BREAST CANCER (noté BREAST), SICK, GERMAN, VEHICLE. Ils sont présentés en annexe (voir page 217). De plus, toutes les variables numériques ont subi un processus de discrétisation supervisée suivant la méthode FUSINTER [ZRR98].

Pour conclure, nous désirons en premier lieu insister sur la simplicité d'utilisation de cette méthode et sur les informations intéressantes qu'elle peut fournir à l'utilisateur grâce à la représentation graphique.

Nous envisageons de poursuivre ces travaux par une étude expérimentales extensive de cette méthodologie (incluant d'autres méthodes de cns, des jeux de données hétérogènes, et la comparaison avec d'autres critères d'évaluation de la validité), nous montrons dans le chapitre suivant que ces résultats nous ont permis de développer deux méthodes de sélection de variables efficaces et rapides dédiées respectivement à l'apprentissage supervisé et à l'apprentissage non supervisé.

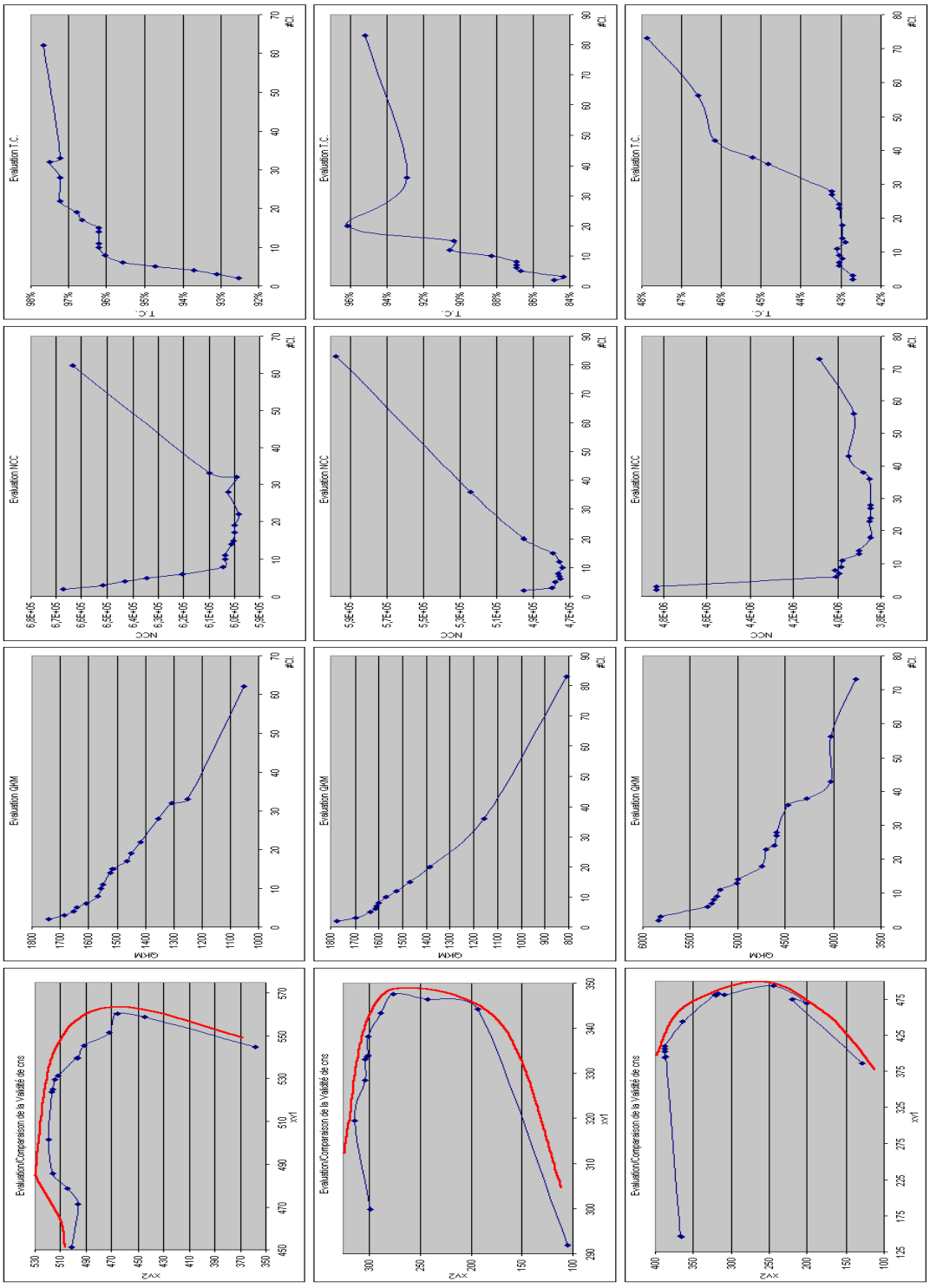


FIG. 4.9 –: *Représentations graphiques pour la détermination des structures des jeux de données : CANCER, HVOTES, CONTRA.*

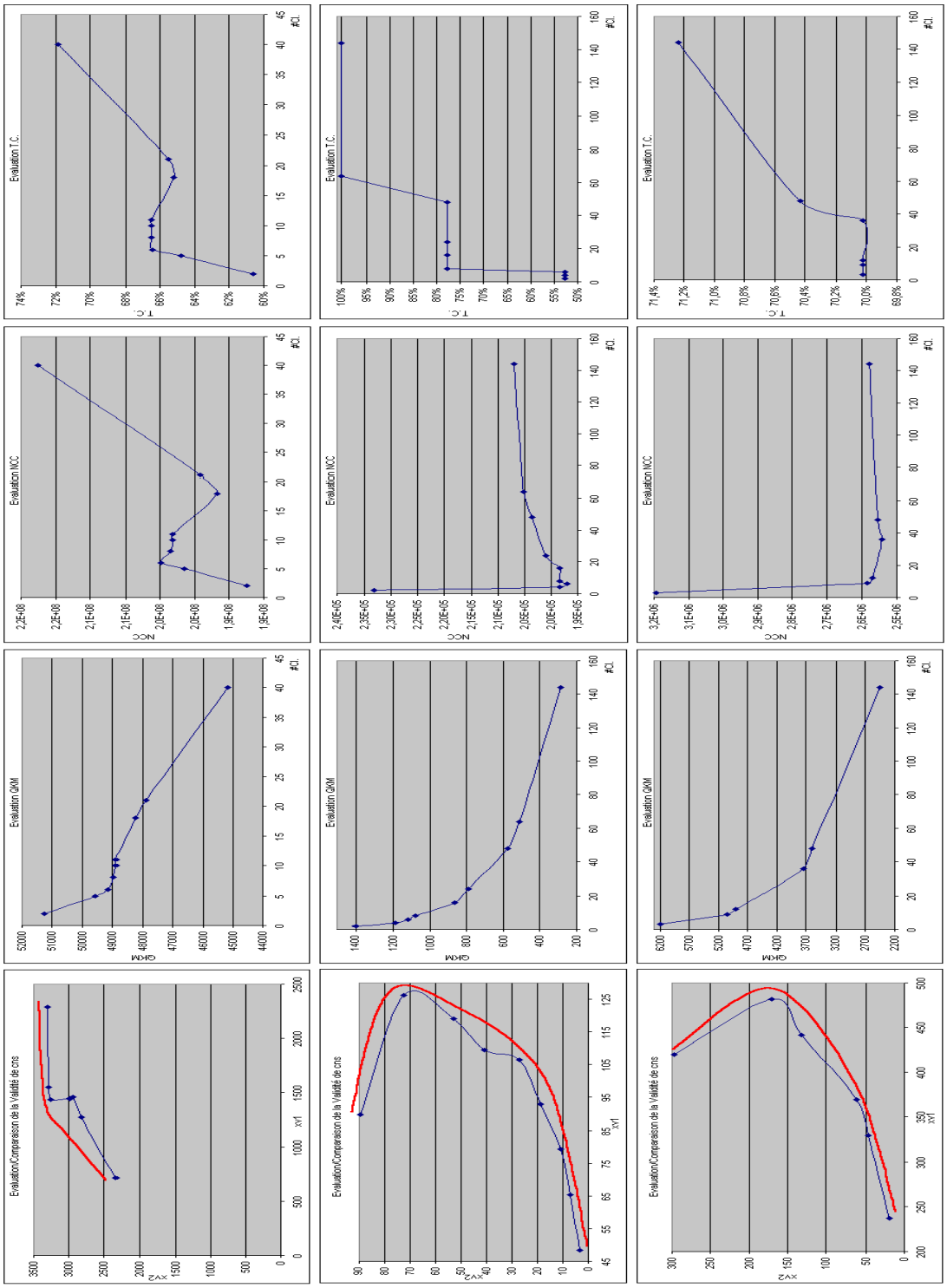


FIG. 4.10 –: *Représentations graphiques pour la détermination des structures des jeux de données : SPAM, MONKS 3, CAR*

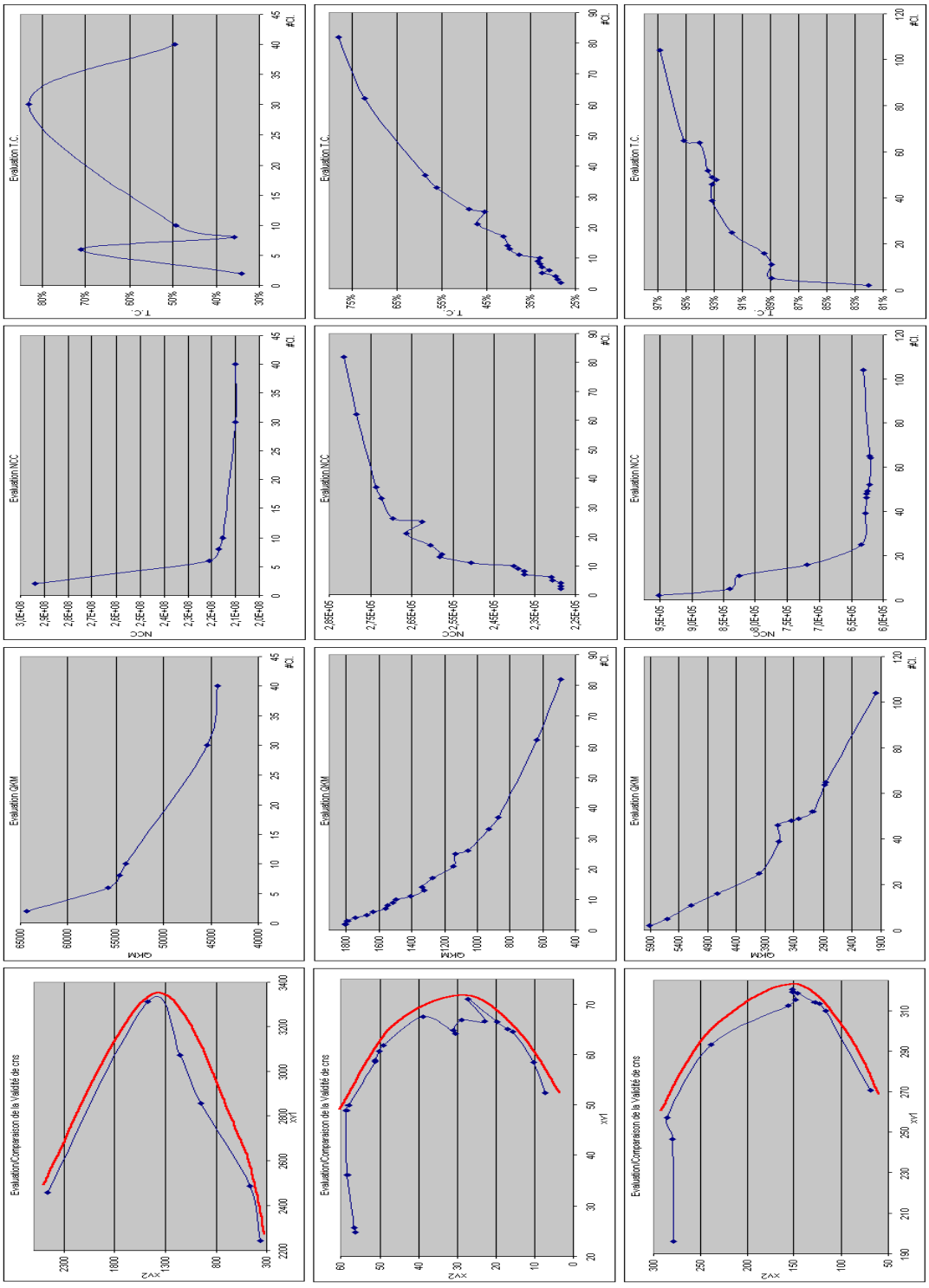


FIG. 4.11 –: *Représentations graphiques pour la détermination des structures des jeux de données : NURSERY, FLAGS, ION*

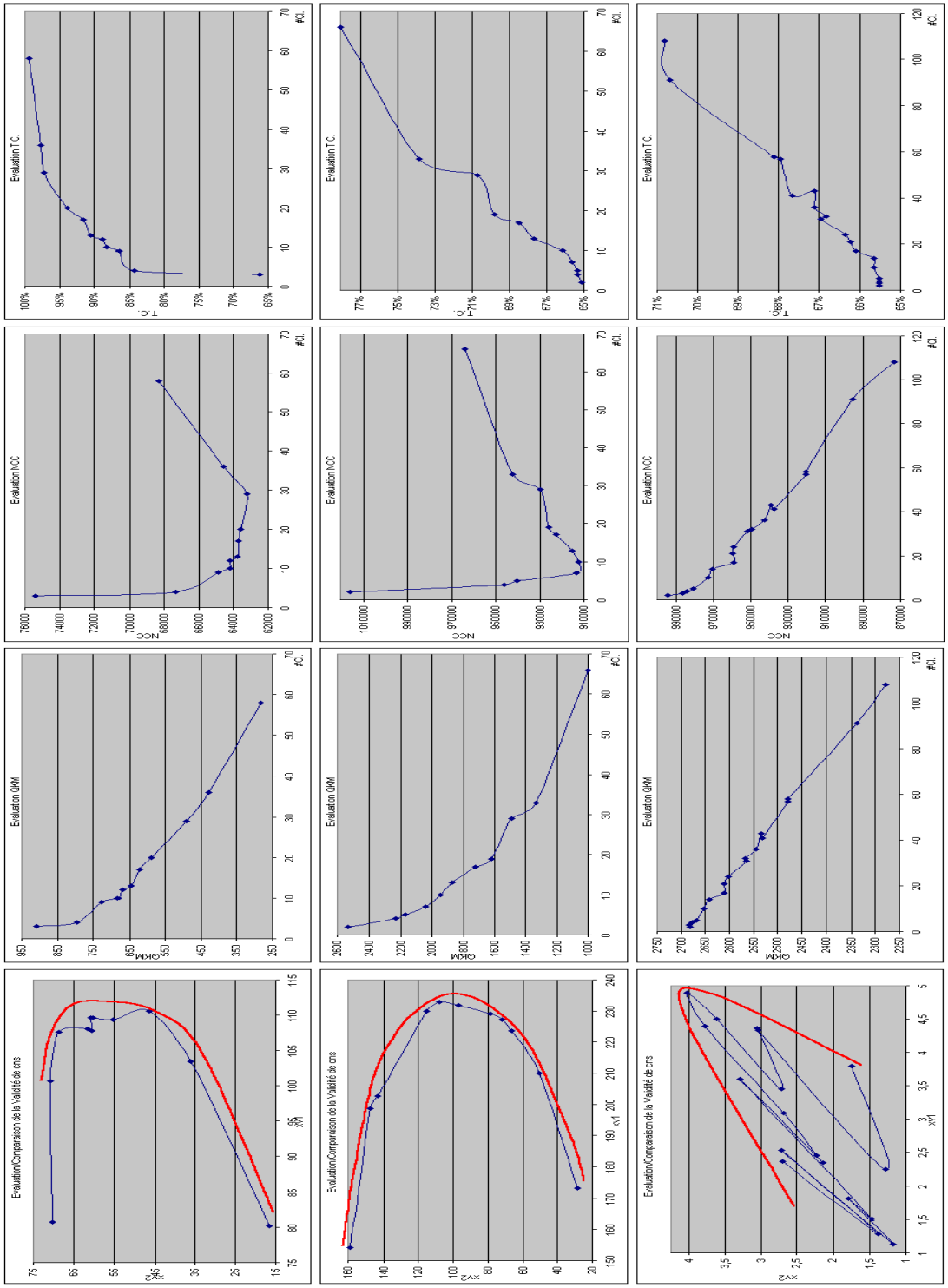


FIG. 4.12 –: *Représentations graphiques pour la détermination des structures des jeux de données : WINE, PIMA, BREAST*

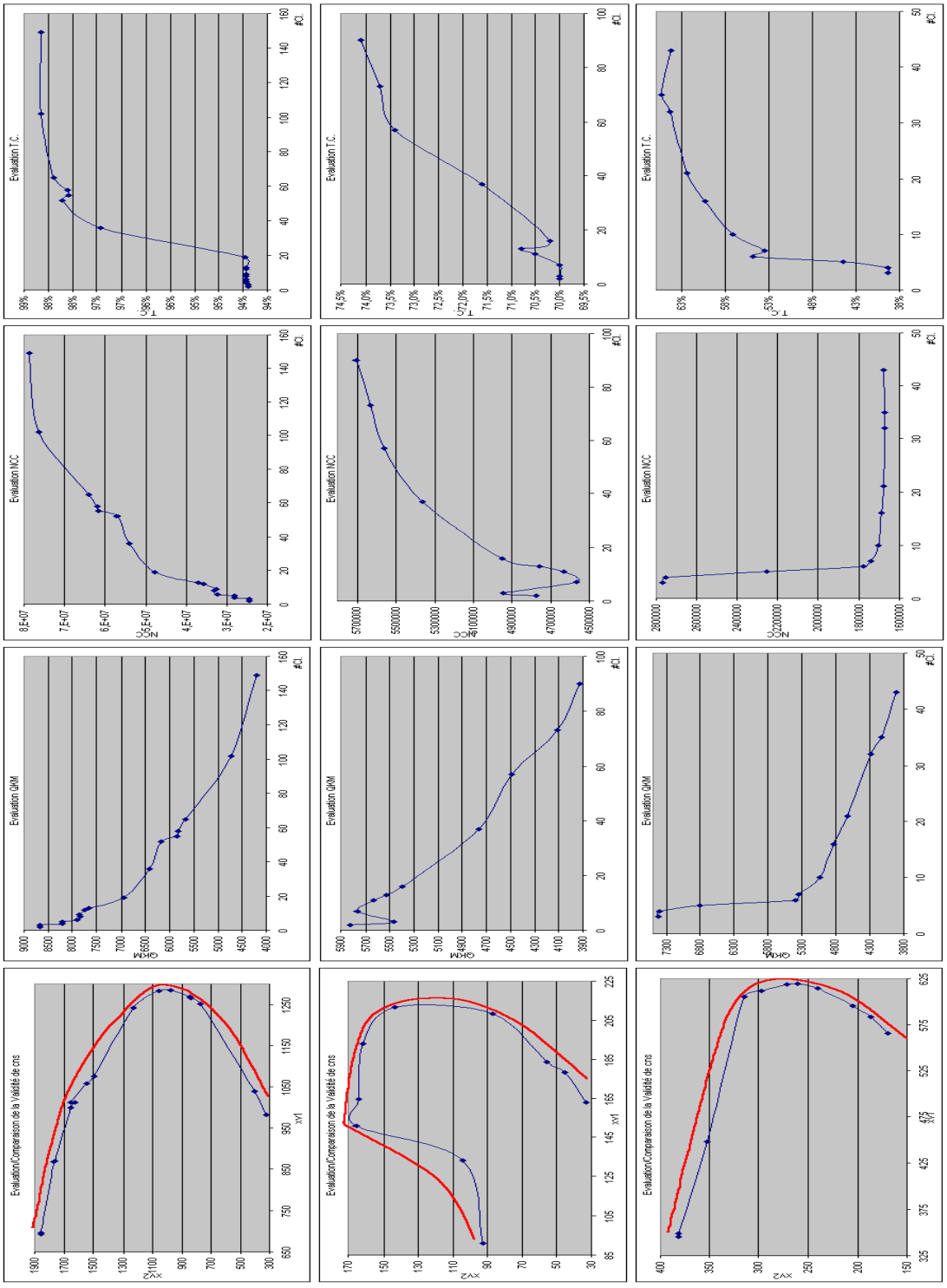


FIG. 4.13 –: *Représentations graphiques pour la détermination des structures des jeux de données : SICK, GERMAN, VEHICLE*

5 Sélection de Variables, Contributions pour l'apprentissage supervisé et non supervisé

"Less is more..."

- Huan Liu & Hiroshi Motoda -
"Feature Extraction, Construction, and Selection : A Data Mining Perspective", Kluwer Academic, Boston MA (1998)

La tendance actuelle d'un accroissement toujours plus fort de la taille des bases de données rend la problématique de l'amélioration de l'espace de représentation des données (ERD) de plus en plus critique en ECD. Une des difficultés majeures associées à la problématique de l'amélioration de la qualité de l'ERD est celle de la dimension de cet espace[1]. Ce problème se traduit par le nombre de variables (descripteurs) caractérisant chaque objet (par exemple, le nombre de variables exogènes dans le cadre de l'apprentissage supervisé)[2]. Un nombre élevé de descripteurs peut en effet s'avérer pénalisant pour un traitement pertinent et efficace des données, d'une part par les problèmes algorithmiques que cela peut entraîner (liés au coût calculatoire et à la capacité de stockage nécessaire), et d'autre part car parmi les descripteurs certains peuvent être non-pertinents, inutiles et/ou redondants perturbant ainsi le bon traitement des données. Or, il est très souvent difficile voire impossible de distinguer les descripteurs pertinents des descripteurs non-pertinents.

Le problème de la dimension des données peut ainsi être résumé par l'aphorisme de Liu et Motoda "Less is more" [LM98] qui met en exergue la nécessité de supprimer l'ensemble des portions non pertinentes des données de manière préalable à tout traitement si on désire en extraire des informations utiles et compréhensibles.

La sélection de variables (SdV) constitue une solution à ce problème. Ce processus vise en effet à la détermination d'un sous ensemble optimal de descripteurs[3] et donc à la réduction du nombre de variables par élimination des variables non pertinentes. Cela implique alors une accélération des traitements

1. l'autre difficulté majeure étant la quantité de données
2. le problème de la quantité de données se caractérise par le nombre d'objets à traiter

postérieurs et peut engendrer une amélioration de la qualité de ces mêmes traitements (la précision prédictive de modèles d'apprentissage supervisé peut, par exemple, être accrue). Enfin, le bruit généré par certaines variables peut être réduit.

La SdV suit généralement un processus itératif (cf. figure 5.1) menant au choix d'un sous-ensemble optimal de variables, ce processus se décompose en trois étapes (Génération de sous-ensembles de variables → Evaluation des sous-ensembles → Test sur le critère d'arrêt) qui s'enchaînent séquentiellement et constituent la partie itérative du mécanisme de SdV. Cette partie itérative s'achève lorsqu'un critère d'arrêt est satisfait, elle est alors suivie par une phase de validation du sous ensemble optimal par l'intermédiaire d'un algorithme d'apprentissage. Plus rarement, la SdV est constituée par un unique processus séquentiel permettant d'établir un classement des différentes variables de l'ERD selon leur intérêt pour le processus ultérieur d'apprentissage.

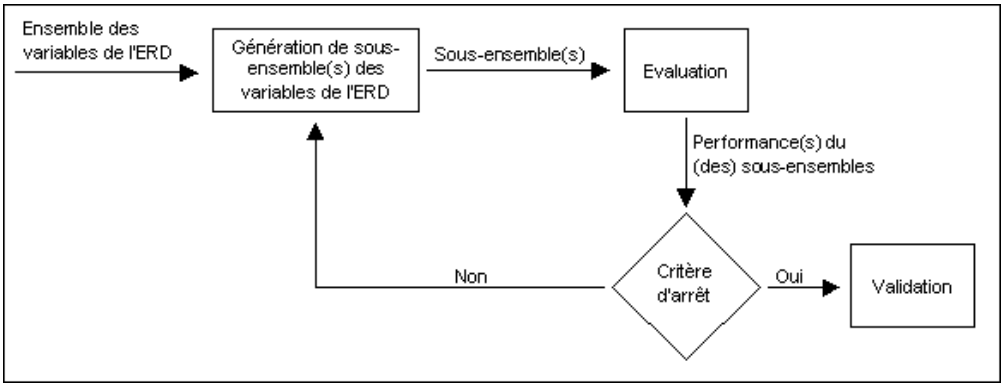


FIG. 5.1 –: *schéma du processus de sélection de variables*

L'évaluation de la qualité du processus de SdV s'effectue en prenant en compte :

- le différentiel de qualité des processus d'apprentissage réalisés respectivement sur l'ERD complet et sur le sous espace de l'ERD constitué des variables sélectionnées (par exemple, le différentiel de précision prédictive en apprentissage supervisé entre un modèle bâti à partir de l'ERD complet et un modèle bâti à partir d'un sous-espace de l'ERD) ;
- le différentiel de dimension de l'ERD complet et du sous espace de l'ERD constitué des variables sélectionnées et conséquemment l'accélération du processus d'apprentissage impliquée par la SdV.

3. l'optimalité du sous ensemble s'entend ici comme l'optimalité du sous ensemble par rapport à un critère particulier

5.1 Sélection de Variables pour l'Apprentissage Supervisé

Pour l'apprentissage supervisé, l'objectif de la SdV est de déterminer quelles sont les variables exogènes de l'ERD pertinentes pour la prédiction de la variable endogène. La réduction de la dimension de l'ERD doit alors permettre une accélération de la phase d'apprentissage et/ou de généralisation, ainsi que l'obtention d'une capacité prédictive du modèle équivalente ou supérieure à celle du modèle bâti à partir l'ensemble complet des variables de l'ERD.

5.1.1 Caractéristiques de la Sélection de Variables

- **Intérêts** :
  - permettre l'élimination de variables inutiles et redondantes,
  - accélérer le processus d'apprentissage
  - améliorer la précision prédictive des algorithmes d'induction.
  - permettre d'effectuer des recherches sur un sous-ensemble optimal de variables (il est alors possible de prendre en compte les interactions qui existent entre les variables).

- **Forme des résultats**, deux formes possibles :
  - une liste ordonnée de variables classées selon un critère d'évaluation. (Ce type de résultats ne fournit des renseignements que sur la pertinence d'une variable par rapport aux autres et le nombre de variables constituant l'ensemble final doit être connu/déterminé.)
  - un sous-ensemble optimum de variables au sein duquel aucune différence ne peut être faîte quant à la pertinence des variables.

- **Caractéristiques des Méthodes** :
  - Un type d'approche : filtre ou enveloppe,
  - Une direction de recherche dans l'espace des sous-ensemble de variables de l'ERD : par élimination ou par ajout ou par élimination/ajout de variable(s),
  - Une stratégie de recherche : complète ou heuristique ou aléatoire,
  - Une fonction d'évaluation des sous ensembles de variables (l'algorithme de recherche doit maximiser ou minimiser cette fonction),
  - Un critère d'arrêt.

5.1.2 Les Types de Méthodes

Il existe deux familles d'algorithmes visant à sélectionner un sous-ensemble optimal de variables : les méthodes "enveloppe" (Wrapper Approach) et les

méthodes "filtre" (Filter Approach). Ces deux familles d'approches s'opposent de part l'utilisation ou la non-utilisation de l'algorithme d'induction : les méthodes enveloppe utilisent l'algorithme d'induction contrairement aux méthodes filtre. Ainsi, les méthodes enveloppe évaluent les différents sous-ensembles générés par l'intermédiaire de l'algorithme d'apprentissage (cf. figure 5.2 à droite) ; les méthodes filtre, quant à elles, n'utilisent absolument pas l'algorithme d'apprentissage dans leur processus de recherche du sous-ensemble optimal (cf. figure 5.2 à gauche).

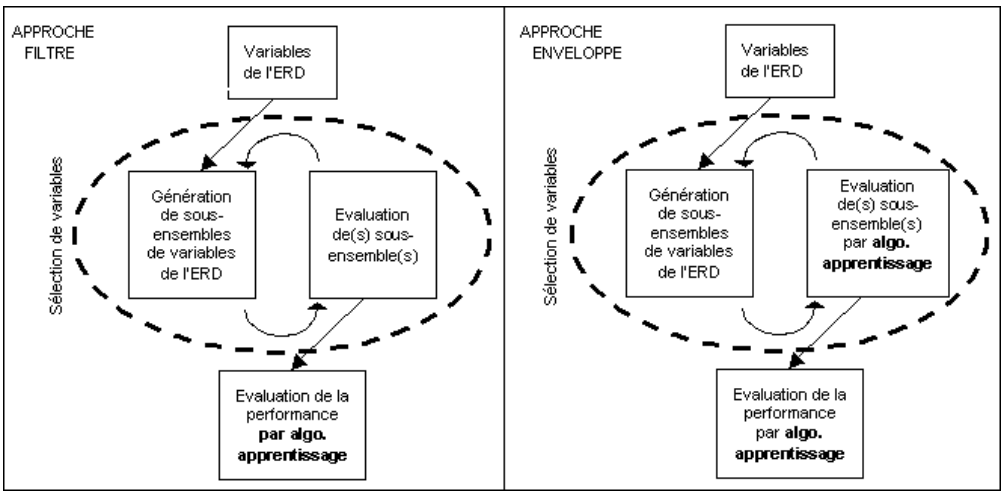


FIG. 5.2 –: *Approches Filtre et Enveloppe pour la Sélection de Variables*

5.1.3 Directions de Recherche

La sélection de variables est un problème de recherche où chaque état de l'espace de recherche spécifie un des 2^p sous-ensemble de variables de l'ERD (p le nombre de variables de l'ERD). Le passage de l'état initial à l'état final peut être schématisé par un graphe partiellement ordonné où chaque état enfant possède un ensemble de variables différents de ses parents. Les méthodes de sélection de variables utilisent donc l'ordre partiel des variables pour organiser leur recherche d'un sous-ensemble optimal de variables. Cet ordre partiel correspond à l'agencement des variables dans le temps, c'est à dire à leur utilisation lors du processus de sélection. Les directions de recherche peuvent être de trois types : Ajout de variables, Suppression de variables, Ajout/Suppression de variables.

5.1.3.1 Forward Selection (FS) (Ajout de variables)

Cette stratégie débute avec l'ensemble vide, puis, à chaque itération, la variable optimale suivant un certain critère est ajoutée. Le processus s'arrête

quand il n'y a plus de variable à ajouter, ou quand un certain critère est satisfait.

5.1.3.2 Backward Elimination (BE) (Suppression de variables)

Cette stratégie débute avec l'ensemble de toutes les variables, puis, à chaque itération, une variable est enlevée de l'ensemble. Cette variable est telle que sa suppression donne le meilleur sous-ensemble selon un critère particulier. Le processus s'arrête quand il n'y a plus de variable à supprimer, ou quand un certain critère est satisfait.

5.1.3.3 Méthodes Bidirectionnelles

Il est également possible d'utiliser une variation de l'ordre partiel des variables : Devijver et Kittler [DK82] définissent un opérateur qui ajoute k variables et en enlève une. La première décision à prendre est donc le point de départ de la recherche :

- Avec un ensemble vide : il s'agit de la Forward Stepwise Selection
- Avec un ensemble complet : Backward Stepwise Elimination
- Avec un ensemble d'attributs choisis aléatoirement.

REMARQUE : Les méthodes bidirectionelles permettent de pallier au problème de l'irrévocabilité de la suppression ou de l'ajout d'une variable. En effet, l'importance d'une variable peut se voir modifiée au cours des différentes itérations du processus de SdV. Ces méthodes autorisent l'ajout et la suppression d'une variable de l'ensemble des variables à n'importe quelle étape de la recherche (autre que la première) contrairement à la FS (resp. BE) pour laquelle une fois qu'une variable a été ajoutée (resp. supprimée) il est impossible de la retirer (resp. réintégrer).

5.1.4 Stratégie de Recherche

La stratégie de recherche dépend de la taille de l'espace de recherche. Si la taille de l'ensemble initial de variables est p, alors le nombre de sous-ensembles candidats est 2^p. Une recherche exhaustive n'est donc que rarement envisageable, ainsi, pour atteindre les objectifs de la SdV trois catégories de méthodes sont applicables :

- **Les stratégies de recherche complète** : une recherche complète des sous-ensembles optimaux est effectuée en tenant compte de la fonction d'évaluation utilisée. Cette méthode n'est pas forcément exhaustive [SJL90] : différentes fonctions heuristiques peuvent être utilisées afin de réduire l'espace de recherche sans compromettre les chances de trouver le sous-ensemble optimal.

- **Les stratégies de recherche heuristique** : on ajoute (ou on ôte) pas à pas des variables à l'ensemble des variables sélectionnées (ou restantes) jusqu'à ce que le sous-ensemble ne puisse plus être amélioré. Cette méthode revient à parcourir le chemin reliant l'état initial à l'état final du graphe des états. A chaque itération, toutes les variables restantes jusque là peuvent être sélectionnées. Il y a plusieurs variantes de ce simple processus mais la génération des sous-ensembles est fondamentalement incrémentale (soit croissante soit décroissante). La taille de l'espace de recherche est p^2 (il existe cependant des exceptions : [KR92a], [Car93]). Ces procédures sont simples à implémenter, peu coûteuses et donnent un résultat très rapidement parce que l'espace de recherche est seulement quadratique en terme de nombre de variables. Mais elles ne permettent pas d'obtenir un sous-ensemble optimal.
- **Les stratégies de recherche aléatoire** : la recherche d'un bon sous-ensemble se fait sur l'espace réduit aux sous-ensembles possibles. La taille de cet espace (inférieure à 2^p) est définie en fixant le nombre d'itérations. L'optimalité de la solution dépend à la fois des ressources disponibles et des valeurs assignées aux paramètres liés à la procédure de génération. Pour obtenir une solution, il n'est pas nécessaire d'attendre la fin de la recherche. Il n'est cependant pas possible de savoir si le sous-ensemble obtenu à l'instant t est optimal mais seulement si il est meilleur que les précédents.

5.1.5 Fonction d'Evaluation

L'objectif associé à la fonction d'évaluation est de mesurer la capacité d'une variable, ou d'un ensemble de variables, à discriminer les classes de la partition impliquée par la variable endogène. L'optimalité d'un sous-ensemble est relative à la fonction d'évaluation utilisée. Dash et Liu [DL97] considèrent que ces fonctions peuvent être regroupées en cinq catégories qui sont les suivantes :

- **Information** : fonctions quantifiant l'information apportée par une variable sur la variable à prédire. La variable, ayant le gain d'information le plus élevé, est préférée aux autres variables. (Le gain d'information étant la différence entre l'incertitude a priori et l'incertitude a posteriori.)
- **Distance** : fonctions s'intéressant au pouvoir discriminant d'une variable. Elles évaluent la séparabilité des classes en se basant sur les distributions de probabilités des classes. Une variable est préférée à une autre si elle induit une plus grande séparabilité.
- **Dépendance** : fonctions mesurant la corrélation ou l'association. Elles permettent de calculer le degré avec lequel une variable exogène est associée à une variable endogène.
- **Consistance** : fonctions liées au biais des variables minimum (min-features bias [AD91]). Ces méthodes recherchent l'ensemble de variables le plus petit qui satisfait un pourcentage d'inconsistance minimum défini par

l'utilisateur. (Deux objets sont dits inconsistants si leurs modalités sont identiques et s'ils appartiennent à deux classes différentes.) Ces mesures peuvent permettre de détecter les variables redondantes.

– **Précision** : ces méthodes utilisent le classifieur comme fonction d'évaluation. Le classifieur choisit, parmi tous les sous-ensembles de variables, celui qui est à l'origine de la meilleure précision prédictive.

5.1.6 Critère d'Arrêt

Il existe deux types de critères d'arrêt selon que celui ci est associé à la stratégie de recherche ou à la fonction d'évaluation :

– Un critère d'arrêt associé à une stratégie de recherche se base soit :
  – sur un nombre pré-défini de variables à sélectionner,
  – sur un nombre d'itérations pré-fixé.
– Un critère d'arrêt associé à une fonction d'évaluation se base soit sur le fait que:
  – l'ajout ou la suppression d'une variable ne produit aucun sous-ensemble plus performant,
  – le sous-ensemble obtenu est, d'après certaines fonctions d'évaluation, le sous-ensemble optimal.

5.1.7 Approches Filtres

Le filtrage est un processus de pré-traitement des données par filtrage des variables non pertinentes avant que n'intervienne la phase d'induction. Il utilise les caractéristiques générales de l'ensemble d'apprentissage pour sélectionner certaines variables et en exclure d'autres. Le schéma le plus simple est d'évaluer individuellement chaque variable grâce à une fonction d'évaluation et de sélectionner les variables possédant les plus grandes valeurs. La fonction d'évaluation est, la plupart du temps, sous la forme d'un critère nommé critère de sélection.

1. ***Critères de Sélection***
   Il convient de distinguer les critères issus de l'approche statistique de ceux basés sur la comparaison par paires d'objets.

   Approche statistique :
   Il existe deux types de mesures : les mesures myopes et les mesures contextuelles.

   – **Les mesures myopes** sont des estimateurs de la qualité d'une variable hors du contexte des autres variables explicatives. Elles sont

inadaptées pour les algorithmes traitant des données contenant des variables corrélées. Il en existe 3 catégories :

- Les mesures liées à l'information : L'entropie de Shannon [Sha48], le gain d'information , le ratio du gain [Qui86], le gain normalisé [JKSK97], la distance de De Mantaras [DM91],
- les mesures liées au critère de distance (la distance existante entre les distributions de probabilités des classes est considérée et ainsi la séparabilité des classes est évaluée) : le critère de Gini [BFOS84], le critère ORT [FI92].
- Les mesures liées au critère de dépendance, ces mesures calculent l'écart à l'indépendance de deux variables d'un tableau de contingence, le critère du khi2 (Pearson 1904), le critère de Tschuprow [Har84][Min87], le coefficient de Cramer.

– **Les mesures contextuelles** estiment la qualité d'une variable descriptive dans le contexte des autres variables descriptives. Ces mesures sont plus coûteuses mais permettent de découvrir des dépendances indécelables par les mesures myopes, la mesure la plus connue est le critère heuristique-statistique τ de Zhou [ZD91].

Comparaisons par Paires
L'idée fondamentale des comparaisons par paires est à attribuer à Condorcet dès 1785 et consiste en la comparaison des partitions induites par deux variables catégorielles, paires d'objets à paires d'objets [Ken39]. Ces mesures sont moins nombreuses que les mesures statistiques. Elles se décomposent également en mesures myopes et mesures contextuelles.

- **Mesures myopes** : Critère de Condorcet [Mic82], critère de Zahn [Zha64], critère de l'écart à l'indépendance [Mar84a][Mar84b].
- **Mesures contextuelles** : le mérite contextuel [Hon94], Relief [KR92a] [KR92b] (ici, le critère à lui seul n'est pas contextuel car une variable est estimée en dehors du contexte des autres variables. Mais, l'algorithme dans lequel il s'intègre le rend contextuel.).

Critères Liés aux Paires de Concepts :
Ces critères, décrit dans [ZKV94], sont basés sur les paires de concepts. Ils travaillent sur les représentants respectifs des concepts et tentent de les discriminer.

2. ***Présentation des Méthodes Filtre***
Afin de répertorier les différents algorithmes de filtrage, deux axes ont été utilisés : la stratégie de recherche et le critère de sélection utilisés. Nous listons maintenant diverses méthodes de SdV, cette présentation est organisée selon les deux axes précités.

Approches Complètes

- **Critère de distance :** message de description de longueur minimale (MDML)[SJL90]
- **Critère de consistance :** PRESET [Mod93], FOCUS [AG92], FOCUS2 [AG94], les méthodes Branch and Bound [NF77], ces dernières utilisent un critère de sélection caractérisé par une propriété de monotonicité (tout sous ensembles de variables possède une valeur du critère de sélection moins bonne ou similaire à celle des sous ensembles l'incluant) qui permet l'élimination, a priori, de certains sous ensembles de variables par utilisation des méthodes Branch and Bound. ABB [LMD98], par exemple, utilise ce principe et un critère d'inconsistance, sa complexité est en $O(2^p n)$.

Approches Heuristiques:

- **Critère d'information :** GIM [AG94], algorithme basé sur la couverture de markov [KS96], Cardie [Car93], MIFS [Bat94]. Cette dernière méthode utilise l'information mutuelle pour évaluer l'information contenue dans chaque variable pour sélectionner le sous ensemble de variable le plus informatif par détermination du sous ensemble de variables qui possède la plus grande information mutuelle avec la variable endogène tout en minimisant l'information mutuelle existant entre les variables exogènes sélectionnées. La sélection des variables est ici effectuée séquentiellement (processus de forward sélection).
- **Critère de distance** : GS [AG94],
- **Critère d'indépendance :** algorithme du khi2 [LS95], CFS [Hal00b] qui utilise le coefficient de Kvalseth (incertitude symétrique) pour mesurer la liaison entre variables associé classiquement à une recherche du type Best First, G3 [LR00],
- **Critère de consistance :** GS pondéré [AG94], Relief [KR92a] qui est un algorithme itératif basé sur la pondération des variables et inspiré des algorithmes d'apprentissage par instances. Relief possèdent plusieurs évolutions (Relief A, Relief B, Relief C, Relief D, Relief E, Relief F), la plus intéressante étant certainement Relief F [Kon94], qui permet de traiter des problèmes multi-classes. La complexité de Relief (et de ces différentes versions) est en $O(Ipn)$, avec n le nombre d'objets et I le nombre d'itérations.

Approches Aléatoires :
LVF [LS96] est une version filtre des algorithmes « Las Vegas ». Ces derniers font des choix probabilistes qui les mènent plus rapidement vers

une solution correcte. Un certain type de ces algorithmes utilise la stratégie aléatoire pour guider leur recherche de telle manière qu'une solution correcte est garantie même si des choix non judicieux ont été réalisés. LVF utilise un critère de sélection basé sur l'inconsistance qui spécifie jusqu'à quel point la réduction de la dimension des données peut être acceptée, le seuil d'acceptation fixé par l'utilisateur, constitue le critère d'arrêt. Cet algorithme trouve rapidement une solution proche de l'optimum.

Méthodes à base d'algorithmes génétiques
Les algorithmes génétiques (AG) sont des stratégies de recherche basées sur le principe de sélection naturelle. Une population de solutions possibles nommées chromosomes est maintenue. Les chromosomes sont sélectionnés, croisés et mutés dans le but de faire évoluer une nouvelle population. Le processus est répété jusqu'à ce qu'une condition d'arrêt soit atteinte pour l'individu le plus adapté de la population ou quand un certain nombre de générations a été produit. La capacité à effectuer des recherches dans des espaces très grands et sans connaissance sur le domaine, ainsi que la relative insensibilité bruit des AGs sont connus. Ils apparaissent donc idéaux pour des usages où les connaissances du domaine et les théories sont difficiles voire impossibles à obtenir. Lors de l'utilisation des AGs, la chose la plus importante est de choisir une représentation bien appropriée et une fonction d'évaluation adéquate. Les AGs ont été utilisés, dans le cadre de la SdV pour l'apprentissage supervisé, par de nombreux auteurs tels que Guerra-Salcedo [GSCWS99], Freitas [Fre02], Whitley et Smith [GSCWS99], Vafaie et De Jong [VJ92], [VDJ93], [VDJ94], Yang [YPH97], [YH98]... Notons que, dans la majorité des cas d'utilisation d'AG, l'approche de SdV n'est pas de type filtre mais de type enveloppe.
Le schéma de la figure 5.4 (page 125) résume le processus de sélection de variables par les AG pour une approche filtre.

L'avantage des approches filtres est leur capacité à être utilisées en amont de n'importe quel algorithme d'induction due à leur indépendance vis à vis de celui ci. Cependant, elles ignorent totalement les effets du sous-ensemble de variables sélectionnées sur les performances de cet algorithme.

5.1.8 Approches Enveloppes

Ces approches ont été introduites par John, Kohavi et Plfeger [JKP94]. Pour ces auteurs, les algorithmes de filtrage ne sont pas toujours efficaces car ils ignorent totalement l'influence de l'ensemble de variables sélectionnées sur

les performances de l'algorithme d'induction. Pour résoudre ce problème, ils proposent une approche différente qui utilise le résultat de l'algorithme d'apprentissage comme fonction d'évaluation : « les méthodes enveloppes ». L'algorithme d'induction appliqué aux données pré-traitées est utilisé comme un sous-programme et considéré comme une boite noire par cet ensemble de méthodes. L'algorithme d'induction travaille sur l'ensemble de données avec différents sous-ensembles de variables et fournit pour chacun d'eux la précision estimée sur le classement de nouvelles instances (cf. figure 5.2 à droite). Le sous-ensemble induisant le classifieur le plus précis est ensuite retenu pour la tache d'induction. Les méthodes enveloppe consistent donc à estimer le taux de succès (par cross-validation) en utilisant uniquement les variables du sous-ensemble à évaluer. D'autres auteurs ont repris cette méthode en utilisant d'autres approches. Parmi ces méthodes, on peut citer : La méthode Oblivion de Langley et Sage [LS94], la méthode BEAM de Aha et Bankert [AB95], CAP de Caruana et Freitag [CF94], NLC [RRJ03], Doak [Doa92], Race [ML94].

Le désavantage majeur des méthodes enveloppe est le coût calculatoire important dû à l'appel de l'algorithme d'induction pour chaque sous-ensemble considéré.

5.1.9 Autres Approches

D'autres approches plus ou moins usitées peuvent être employées nous pouvons notamment citées les méthodes issues de l'économétrie et de l'analyse factorielle. Enfin, les méthodes basées sur les support vectors machines constituent une approche suscitant actuellement un intérêt très vif, le très récent numéro spécial de la revue *Journal of Machine Learning Research* [GE03] témoigne parfaitement de cet intérêt et introduit cette problématique tout en proposant plusieurs méthodes participant des approches filtre et/ou enveloppe. Méthodes de sélection de variables utilisant les SVM (Méthode de Stoppiglia [SD03] (classement des variables selon leur pertinence), l'algorithme SVM-RFE [Rak03], l'algorithme VS-SSVM [BBMES03] (algorithmes de Backward Sélection)...).

Méthodes	Auteur et Année	Types	Dir. de Recherche	Strat. de Recherche	Critère de Sélection	Critère d'Arrêt
Branch and Bound	Narandra et Fukunaga, 1977	Filtre	Backward	Complète	Consistance	Nb. Itérations
MDLM	Scheinvald, 1990	Filtre	Forward	Complète	Distance	Nb. Itérations
Focus	Almuallim et Dietterich, 1991	Filtre	Forward	Complète	Consistance	Sous-Ens. Optimal
Relief	Kira et Rendell, 1992	Filtre	Autre	Heuristique	Consistance	Seuil Seuil
Focus 2	Almuallim et Dietterich, 1992	Filtre	Forward	Complète	Consistance	Sous-Ens. Optimal
Preset	Modrzejewski, 1993	Filtre	Autre	Complète	Consistance	Sous-Ens. Optimal
Sélection et AG	Vafaie et De Jong, 1993	Filtre	Autre	Aléatoire	Consistance	Nb. Itérations
GIM	Almuallim, 1994	Filtre	Forward	Heuristique	Information	Sous-Ens. Optimal
MIFS	Battiti, 1994	Filtre	Forward	Heuristique	Information	Sous-Ens. Optimal
GS	Almuallim, 1995	Filtre	Forward	Heuristique	Distance	Pas d'amélioration
GP	Almuallim, 1996	Filtre	Forward	Heuristique	Consistance	Sous-Ens. Optimal
Relief-F	Kononenko, 1994	Filtre	Autre	Heuristique	Consistance	Seuil
Khi2	Liu, 1995	Filtre	Autre	Heuristique	Indépendance	Sous-Ens. Optimal
LVF	Liu, 1996	Filtre	Autre	Aléatoire	Consistance	Nb. Itérations
Sél. & couv. de Markov	Koller, 1996	Filtre	Backward	Heuristique	Information	Nb. Itérations
CFS	Hall, 2000 2000	Filtre	Forward	Heuristique	Indépendance	Sous-Ens. Optimal
G3	Rakotomalala et Lallich, 2000	Filtre	Forward	Heuristique	Indépendance	Sous-Ens. Optimal

TAB. 5.1 –: *Tableau récapitulatif (Partie 1) inspiré de l'exposé de l'article [LEB02]*

Méthodes	Auteur et Année	Types	Dir. de Recherche	Strat. de Recherche	Critère de Sélection	Critère d'Arrêt
Méthode de Doak	Doak, 1992	Env.	Autre	Heuristique	Précision	Sous-Ens. Optimal
CAP	Caruana et Freitag, 1994	Env.	Autre	Complète	Précision	Sous-Ens. Optimal
RACE	Moore et Lee	Env.	Autre	Complète	Précision	Sous-Ens. Optimal
Méthode de John	John, 1994	Env.	Forward	Complète	Précision	Pas d'amélioration
Oblivious	Langley et Sage, 1994	Env.	Backward	Complète	Précision	Sous-Ens. Optimal
Régressions	Johnston, 1988	Env.	Forward	Complète	Consistance	
Backward Elimination	Johnston, 1989	Env.	Backward	Heuristique	Consistance	Seuil
Forward Selection	Johnston, 1990	Env.	Forward	Heuristique	Consistance	Seuil Seuil
Stepwise Regression	Johnston, 1991	Env.	Forward	Heuristique	Consistance	Seuil Seuil
Stagewise Regression	Johnston, 1992	Env.	Forward	Heuristique	Consistance	Seuil Seuil
BEAM	Aha et Bankert, 1996	Env. .	Autre	Heuristique	Précision	Sous-Ens. Optimal
NLC	Ruiz, 2003	Env.	Forward	Heuristique	Précision	Sous-Ens. Optimal
SVM SVM	Stoppiglia, 2003	Filtre	Forward	Heuristique	Distance	Seuil

TAB. 5.2 –: *récapitulatif (Partie 2) inspiré de l'exposé de l'article [LEB02]*

5.2 Contribution à la Sélection de Variables pour l'Apprentissage Supervisé : Une Nouvelle Méthode Efficiente et Rapide

Nous proposons maintenant une nouvelle méthode efficiente et rapide pour la sélection de variables dans le cadre de l'apprentissage supervisé sur variables catégorielles[4]. Cette méthode de type filtre ne requiert qu'une unique passe sur le jeux de données (ce qui lui confère un avantage calculatoire important sur la plupart des méthodes). Elle utilise en fait un algorithme génétique (AG) et la méthodologie d'évaluation/comparaison de la validité de cns (cf. chapitre 4) au sein d'un processus itératif pour la sélection d'un sous-ensemble de variables .

Dans les sections suivantes nous considérons un problème d'apprentissage caractérisé par un ensemble $O = \{o_1,...,o_n\}$ de n objets décrits par :

- un espace de représentation des données EV comprenant p variables catégorielles $EV = \{V_1,...,V_p\}$. ($o_i = \{o_{i_1},...,o_{i_p}\}$)
- une variable catégorielle V_A qui représente le concept à apprendre (variable endogène) possédant k modalités.

Nous utilisons un problème d'apprentissage synthétique pour illustrer nos développements (voir table 5.3) : $O = \{o_1,o_2,o_3,o_4,o_5,o_6,o_7\}$, $EV = \{V_1,V_2,V_3,V_4\}$, V_A a 3 modalités a,b,c.

	V_1	V_2	V_3	V_4	V_A
o_1	o	o	o	o	a
o_2	o	o	n	o	a
o_3	o	n	o	o	a
o_4	n	o	n	o	b
o_5	n	o	o	o	b
o_6	n	o	n	n	c
o_7	n	n	o	n	c

TAB. 5.3 –: *Jeu de données synthétiques*

5.2.1 Hypothèses et Idées Fondamentales

Nous donnons maintenant les idées et hypothèses qui constituent les bases de la méthode que nous proposons :

1. ***Hypothèse** : Si l'ERD, EV, d'un problème d'apprentissage est tel que le concept à apprendre implique une structure naturelle de l'ensemble des objets O dans cet ERD, alors cela doit permettre un bon processus d'apprentissage.*

4. Cette méthode peut également être employée dans le cadre de données quantitatives mais son coût calculatoire s'accroît alors grandement la rendant moins attrayante...

2. ***Hypothèse*** *: Une cns valide de l'ensemble des objets O correspond à une structure naturelle de O.*
3. ***Hypothèse*** *: Sur la base de 1 et 2, on peut admettre que si l'ERD EV d'un problème d'apprentissage est tel que le concept à apprendre implique une organisation des objets de O selon une cns valide dans cet espace EV, alors l'ERD EV doit autoriser un bon processus d'apprentissage.*
4. ***Idée*** *: Dans le cadre de la sélection de variables, nous pouvons considérer que l'ERD $EV_\star \subseteq EV$, constitué des variables sélectionnées pour l'apprentissage, doit être tel que le concept à apprendre implique une organisation des objets de O selon une cns valide dans l'espace $EV_\star$.*
5. ***Hypothèse*** *: La cns de l'ensemble d'objets O impliquée par le concept à apprendre est composée d'autant de classes qu'il existe de modalités du concept à apprendre, et chaque classe est exclusivement composée d'objets correspondant à la même modalité du concept à apprendre.(Cette cns est notée par la suite P)*
   Pour le problème d'apprentissage servant d'exemple :
   $P = \{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o_6,o_7\}\}$.
6. ***Idée*** *: Dans le cadre de la sélection de variables, si nous considérons l'ensemble de tous les ERDs potentiels (ces espaces sont les sous espaces non vides de EV), l'ERD que l'on sélectionne finalement (i.e. l'ERD constitué des variables sélectionnées pour l'apprentissage) doit être tel que la cns P apparaît comme la plus valide au sein de ce sous espace de EV.*

5.2.2 Evaluation de la Validité d'une Partition dans un Sous-Espace de l'ERD

La notion de validité d'une partition $P = \{C_1,...,C_k\}$ dans un sous espace de l'ERD $EV_\star$ est définie de manière identique à la validité d'une partition dans l'ERD complet EV : on utilise la méthode présentée au chapitre précédent. Le point important est de ne prendre en compte non pas l'ensemble des variables de EV mais uniquement celles de $EV_\star$. Nous utilisons donc les indices $LM_{EV_\star}(P)$, $NLM_{EV_\star}(P)$, $LD_{EV_\star}(P)$, $NLD_{EV_\star}(P)$:

$$LM_{EV_\star}(P) = \sum_{g=1..k} \sum_{\substack{o_a \in C_g, o_b \in C_g, \\ a<b}} \sum_{\substack{i \text{ tel que} \\ V_i \in EV_\star}} (lien_i(o_{a_i}, o_{b_i}))$$

$$NLM_{EV_\star}(P) = \sum_{g=1..k} \sum_{\substack{o_a \in C_g, o_b \in C_g, \\ a<b}} \sum_{\substack{i \text{ tel que} \\ V_i \in EV_\star}} (1 - lien_i(o_{a_i}, o_{b_i}))$$

$$LD_{EV_\star}(P) = \sum_{\substack{f=1..k, g=1..k \\ f<g}} \sum_{o_a \in C_f, o_b \in C_g,} \sum_{\substack{i \text{ tel que} \\ V_i \in EV_\star}} (lien_i\,(o_{a_i}, o_{b_i}))$$

$$NLD_{EV_\star}(P) = \sum_{\substack{f=1..k, g=1..k \\ f<g}} \sum_{o_a \in C_f, o_b \in C_g,} \sum_{\substack{i \text{ tel que} \\ V_i \in EV_\star}} (1 - lien_i\,(o_{a_i}, o_{b_i}))$$

Ce qui nous permet de définir les indices supplémentaires :

- $L_{EV_\star}(O) + NL_{EV_\star}(O) = \frac{n\times(n-1)}{2} \times card(EV_\star)$
- $D_{EV_\star}(P) + M_{EV_\star}(P) = \frac{n\times(n-1)}{2} \times card(EV_\star)$
- $M_{EV_\star}(P) = NLM_{EV_\star}(P) + LM_{EV_\star}(P)$
- $D_{EV_\star}(P) = NLD_{EV_\star}(P) + LD_{EV_\star}(P)$
- $L_{EV_\star}(O) = LM_{EV_\star}(P) + LD_{EV_\star}(P)$
- $NL_{EV_\star}(O) = NLM_{EV_\star}(P) + NLD_{EV_\star}(P)$

Ainsi, nous utiliserons les indices $xv_1^{EV_\star}(P)$ et $xv_2^{EV_\star}(P)$ pour caractériser la validité de P au sein de $EV_\star$:

- $xv_1^{EV_\star}(P) = \frac{LM_{EV_\star}(P) - M_{EV_\star}(P) \times \frac{L_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)}}{\sqrt{M_{EV_\star}(P) \times \frac{L_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)} \times (1 - \frac{L_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)})}}$
  $xv_1^{EV_\star}(P) \hookrightarrow N\,(0,1)$
- $xv_2^{EV_\star}(P) = \frac{NLD_{EV_\star}(P) - D_{EV_\star}(P) \times \frac{NL_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)}}{\sqrt{D_{EV_\star}(P) \times \frac{NL_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)} \times (1 - \frac{NL_{EV_\star}(O)}{L_{EV_\star}(O)+NL_{EV_\star}(O)})}}$
  $xv_2^{EV_\star}(P) \hookrightarrow N\,(0,1)$

Ainsi, on peut comparer la validité de P dans différents sous espaces de EV, et ce en suivant un mode de comparaison identique à celui présenté dans le chapitre 4.

5.2.3 La Nouvelle Méthode de Sélections de Variables

Nous montrons maintenant comment utiliser la méthodologie de comparaison de validité de cns (présentée au chapitre 4) en l'associant à un AG pour

bâtir une méthode de sélection de variables pour l'apprentissage supervisé basée sur les idées et hypothèses émises précédemment.

5.2.3.1 La Méthode de Base : une Méthode Exhaustive

L'idée de base de cette méthode est de considérer la partition P et de tester la validité de cette cns dans chaque sous espace de EV. Étant données les hypothèses précédemment émises, l'ERD sélectionné (ou l'ensemble des sous-espaces sélectionnés) est le sous espace (ou l'ensemble des sous-espaces) impliquant la plus forte validité pour P. La méthode peut être traduite par l'algorithme 4.

Ce processus requiert une unique passe sur les données pour obtenir toutes les tables de contingence utiles (qui ne nécessitent qu'une faible capacité de stockage), $2^p - 1$ calculs pour tester chaque sous espace non vide de EV, et $2^p - 1$ comparaisons pour déterminer le meilleur sous espace (ou l'ensemble des meilleurs sous espaces). (voir le chapitre précédent et l'exemple suivant l'algorithme 4 pour des informations complémentaires sur le coût calculatoire.) Si le nombre de variables p est faible, l'utilisation de cette méthode est envisageable (car réalisable du point de vue calculatoire), mais pour des nombres de variables un peu plus élevés l'utilisation de cette méthode n'est pas envisageable du point de vue calculatoire. Nous devons alors adopter une heuristique pour déterminer le meilleur sous espace, ou au moins, un bon sous espace, sans pour autant utiliser une phase de test exhaustive et ainsi limiter le coût calculatoire de la méthode. Nous avons choisi d'adopter les algorithmes génétiques (AGs) qui sont connus comme une solution efficace pour la résolution de problèmes combinatoires.

Algorithme 4 SdV pour l'Apprentissage Supervisé : Méthode Exhaustive

1. Données : la partition P, l'ERD EV

2. En une unique passe sur les données bâtir les tables de contingence nécessaires aux calculs des mesures de validité nécessaires à la méthodologie d'évaluation/comparaison de la validité de cns présentée préalablement (i.e. les tables de contingence croisant la variable endogène et exogènes).

3. En utilisant les informations de ces tables de contingence, calculer les valeurs des 2 mesures de validité $xv_1^{EV_\star}$ et $xv_2^{EV_\star}$ pour la cns P dans chaque sous espace non vide $EV_\star$ de EV puis comparer la validité de la cns P dans chacun de ces sous espaces.

4. Cette comparaison permet la sélection du sous-espace dans lequel (ou de l'ensemble des sous-espaces) dans lequel (ou lesquels) P apparaît la plus valide. Ce sous espace (ou cet ensemble de sous espaces) constitue alors le sous espace sélectionné (ou l'ensemble des sous-espaces sélectionnés).

EXEMPLE : *Pour le jeu de données synthétique illustratif, appliquer cette méthode exhaustive revient tout d'abord à calculer les valeurs pour* $xv_1^{EV_\star}(P)$ *et* $xv_2^{EV_\star}(P)$ *dans chaque sous-espace de* $EV = \{V_1,V_2,V_3,V_4\}$*. Ces valeurs sont reportées dans la table 5.4 et la figure 5.3 présente graphiquement ces valeurs.*[5]

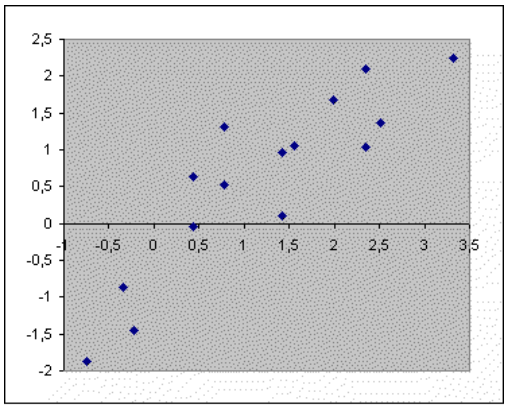


FIG. 5.3 –: *Valeurs* $xv_1^{EV_\star}(P)$ *et* $xv_2^{EV_\star}(P)$ *dans chaque sous-espace de* $EV = \{V_1,V_2,V_3,V_4\}$

Nous rappelons une nouvelle fois que le calcul de l'ensemble de ces valeurs n'a nécessité qu'une seule passe sur les données :

– *cette unique passe permet d'obtenir les tables de contingence croisant la variable endogène* V_A *avec les variables exogènes.*

$V_A \setminus V_1$	o	n	$V_A \setminus V_2$	o	n	$V_A \setminus V_3$	o	n	$V_A \setminus V_4$	o	n
a	3	0	a	2	1	a	2	1	a	3	0
b	0	2	b	2	0	b	1	1	b	2	0
c	0	2	c	1	1	c	1	1	c	0	2

Tables de Contingence croisant la variable endogène V_A *et les variables exogènes*

– *le calcul de chaque valeur est alors réalisé à partir de ces tables : si la table de contingence pour une variable* V_i *est notée:*

$V_A \backslash V_i$	V_{i_1}	...	$V_{i_{m_i}}$	
V_{A_1}	α_{1i1}	...	α_{1im_i}	$\alpha_{1i.}$
...	...	...	...	...
V_{A_k}	α_{ki1}	...	α_{kim_i}	$\alpha_{ki.}$
	$\alpha_{.i_1}$	...	$\alpha_{.im_i}$	n

V_A la variable endogène à k modalités,
V_i une variable exogène à m_i modalités notées $V_{i_j} (j = 1..m_i)$.
α_{lih} le nombre d'objets ayant la valeur V_{i_h} pour V_i et la valeur V_{A_l} pour V_A.
$\alpha_{.i_j} = \sum_{h=1..k} \alpha_{hi_j}$
$\alpha_{hi_.} = \sum_{j=1..m_i} \alpha_{hi_j}$

5. Attention, cet exemple vise essentiellement à illustrer les différentes étapes de la méthodologie, en effet, étant donné le faible nombre d'individus du jeu de données l'approximation normale est ici douteuse...

	$xv_1(P)^{EV_\star}$	$xv_2(P)^{EV_\star}$	fit_1	rang fit_1	fit_2	rang fit_2	f_1	f_2
$V_\star = \{V_1\}$	2,35	2,09	3,14	2	11	2	11	11
$V_\star = \{V_2\}$	-0,34	-0,87	0	12	14,14	12	0	14,14
$V_\star = \{V_3\}$	-0,22	-1,46	0	12	14,14	12	0	14,14
$V_\star = \{V_4\}$	2,35	1,03	2,57	5	11,79	5	11,58	11,79
$V_\star = \{V_1,V_2\}$	1,42	0,96	1,71	7	12,46	7	12,43	12,46
$V_\star = \{V_1,V_3\}$	0,78	1,32	1,53	8	12,66	8	12,61	12,66
$V_\star = \{V_1,V_4\}$	3,32	2,24	4	1	10,24	1	10,14	10,24
$V_\star = \{V_2,V_3\}$	-0,75	-1,87	0	12	14,14	12	0	14,14
$V_\star = \{V_2,V_4\}$	1,42	0,11	1,42	9	13,09	9	12,72	13,09
$V_\star = \{V_3,V_4\}$	0,78	0,53	0,94	10	13,22	10	13,2	13,22
$V_\star = \{V_1,V_2,V_3\}$	0,44	0,64	0,78	11	13,38	11	13,37	13,38
$V_\star = \{V_1,V_2,V_4\}$	2,51	1,37	2,86	3	11,43	3	11,28	11,43
$V_\star = \{V_1,V_3,V_4\}$	1,99	1,67	2,6	4	11,56	4	11,54	11,56
$V_\star = \{V_2,V_3,V_4\}$	0,44	-0,05	0	12	14,14	12	0	14,14
$V_\star = \{V_1,V_2,V_3,V_4\}$	1,56	1,05	1,88	6	12,3	6	12,26	12,3

TAB. 5.4:

on peut alors calculer :

$$LM(P) = \sum_{\substack{i=1..p \text{ tel que} \\ V_i \in EV_\star}} \sum_{j=1..m_i} \sum_{z=1..k} \frac{\alpha_{zi_j}(\alpha_{zi_j}-1)}{2}$$

$$M(P) = card(EV_\star) \times \sum_{z=1..k} \frac{card(C_z)(card(C_z)-1)}{2}$$

$$L(O) = \sum_{\substack{i=1..p \text{ tel que} \\ V_i \in EV_\star}} \sum_{j=1..m_i} \frac{\alpha_{.i_j}(\alpha_{.i_j}-1)}{2}$$

$$NLM(P) = M(P) - LM(P)\,;\, LD(P) = L(O) - LM(P)$$

$$D(P) = \frac{n(n-1)}{2} \times card(EV_\star) - M(P)\,;\, NLD(P) = NV(P) - LD(P)$$

– *Puis, on utilise la méthodologie du chapitre 4 pour déterminer le sous-espace de EV dans lequel P apparaît comme la plus naturelle, il s'agit ici de $V_\star = \{V_1,V_4\}$ (notons au passage que la conjonction de ces deux variables correspond effectivement au concept le plus simple permettant une discrimination parfaite des 3 modalités de la variable endogène AL).*

5.2.3.2 Réduction de la Complexité par Introduction d'un AG

Le problème auquel nous sommes confronté, la découverte d'un bon sous-espace sans pour autant pratiquer une recherche exhaustive, peut effectivement être résolu efficacement par utilisation d'un AG de la manière suivante :

- chaque chromosome de l'AG correspond à un sous espace de EV qui est caractérisé par la présence/absence de variables de EV ;
- chaque chromosome possède p gènes, chaque gène correspond à l'une des p variables de EV, un gène a une valeur binaire (un gène est codé sur un seul bit) qui code la présence/absence de la variable dans le sous espace de EV codé par le chromosome ;
- la fonction de fitness de l'AG est basée sur la méthodologie pour l'évaluation/comparaison de la validité de cns. Toutefois dans la mesure où cette fonction de fitness doit permettre de comparer tout couple de sous espaces (i.e. selon cette fonction il n'existe pas de couple de sous espaces tel que la fonction de fitness ne puisse comparer la validité de P dans ces 2 sous espaces), la fonction de fitness correspond à une adaptation de la méthodologie proposée préalablement.
- pour le reste, l'AG est utilisé et défini de manière classique.

L'algorithme 5 ainsi que la figure 5.4 illustrent le fonctionnement de la méthode de sélection de variables que nous proposons.

Algorithme 5 Sélection de Variables pour l'Apprentissage Supervisé : Utilisation d'un AG

---

1. Données : la cns P, l'ERD EV

2. En une unique passe sur les données bâtir les tables de contingence nécessaires aux calculs des mesures de validité nécessaires à la méthodologie d'évaluation/comparaison de la validité de cns présentée préalablement.

3. Fixer les paramètres de l'AG : *nombre de générations, taille de la population, Probabilité de Croisement, Probabilité de mutation*

4. Lancer l'AG utilisant la fonction de fitness spécifique définie par la suite.

5. Sélectionner le meilleur sous-espace déterminé par l'AG

Concernant la définition de la fonction de fitness de l'AG, un problème est effectivement soulevé : la méthodologie utilisée pour l'évaluation/comparaison de la validité de cns implique une optimisation multi-objectif (elle nécessite la comparaison de couples de valeurs et peut mener à des comparaisons impossibles de sous espaces) qui s'avèrent problématique pour l'utilisation d'AGs. L'utilisation d'AGs multi-objectif qui impliquent pour la plupart des mécanismes de sélection coûteux du point de vue calculatoire, constitue une solution éventuelle, nous lui préférons la solution consistant à dériver une fonc-

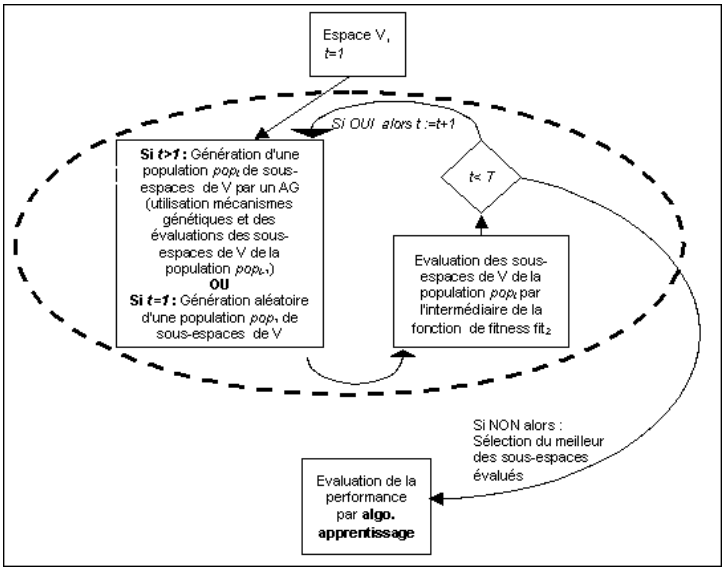


FIG. 5.4 –: *schéma fonctionnel de la méthode proposée*

tion de fitness telle qu'elle intègre en une unique fonction objectif les deux fonctions objectifs.

Nous proposons maintenant deux fonctions basées sur l'observation que P doit présenter de fortes valeurs pour $xv_1^{EV_\star}$ et $xv_2^{EV_\star}$ pour être considérée comme valide dans $EV_\star$. Ces fonctions $fit1(P,EV_\star)$ et $fit2(P,EV_\star)$ (avec $EV_\star$ le sous espace considéré) sont les suivantes :

1. $fit1(P,EV_\star) = \begin{cases} \sqrt{(xv_{1_P}^{EV_\star})^2 + (xv_{2_P}^{EV_\star})^2} \text{ si } xv_{1_P}^{EV_\star} > 0 \text{ et } xv_{2_P}^{EV_\star} > 0 \\ 0 \text{ sinon} \end{cases}$
   qui correspond en quelque sorte à une distance du point de vue de la validité entre une structure ne constituant pas une cns valide (ou encore la cns P dans un sous espace de EV tel que cette cns n'apparaisse pas comme valide) et la cns P dans le sous espace $EV_\star$. Cette fonction de fitness doit être maximisée.

2. $fit2(P,EV_\star) = \begin{cases} \sqrt{(\tilde{x_1} - xv_{1_P}^{EV_\star})^2 + (\tilde{x_2} - xv_{2_P}^{EV_\star})^2} \text{ si } xv_{1_P}^{EV_\star} > 0 \text{ et } xv_{2_P}^{EV_\star} > 0 \\ +\infty \text{ sinon} \end{cases}$
   qui correspond en quelque sorte à une distance du point de vue de la validité entre une cns virtuelle particulière (dont les valeurs xv_1 et xv_2 seraient respectivement $\tilde{x_1}$ et $\tilde{x_2}$) et la cns P. En fait, dans ce cas, nous fixons $\tilde{x_1} = \tilde{x_2} = très\ forte\ valeur$ de manière à conférer à la cns virtuelle particulière l'aspect d'une sorte de cns idéale du point de vue de la validité (ou encore la validité de P dans un espace tel qu'il confère à P une validité idéale). Ainsi, cette dernière fonction de fitness correspond en somme à une distance du point de vue de la validité entre une cns

virtuelle idéale du point de vue de la validité et la cns P. Cette fonction de fitness doit donc être minimisée.

Les tests réalisés ont montré que la seconde fonction de fitness est la plus intéressante car elle mène à des cns possédant des valeurs équilibrées pour xv_1 et xv_2 contrairement à la première fonction de fitness qui peut mener à des valeurs non équilibrées (i.e. une très forte valeur pour l'une des 2 valeurs et faible pour l'autre). Cela s'explique notamment par la forme de ces deux fonctions, la figure 5.5 présentant respectivement les surfaces impliquées par la fonction $f_1(x,y) = \left(max_{x\in[0;10],y\in[0;10]}(\sqrt{x^2+y^2})\right) - \sqrt{x^2+y^2}$ et par la fonction $f_2(x,y) = \sqrt{(10-x)^2+(10-y)^2}$ permet d'appréhender cela de manière intuitive.

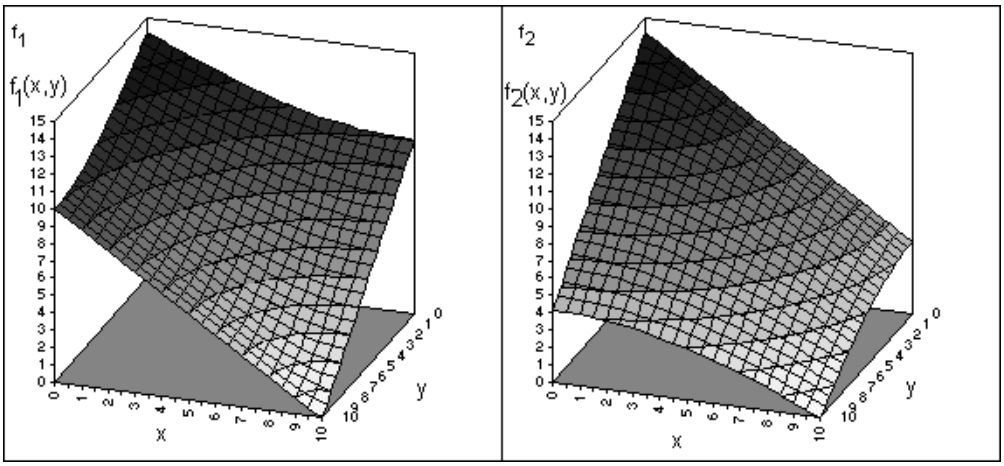


FIG. 5.5 –: *fonctions f_1 et f_2*

5.2.4 Evaluation Expérimentale

5.2.4.1 Présentation de l'Evaluation Expérimentale

L'évaluation expérimentale, réalisée sur 17 jeux de données de la collection de l'université de Californie à Irvine [MM96][6], a mis en jeu cinq méthodes d'apprentissage différentes : ID3, C4.5, Sipina, 1-plus proche voisins et bayésiens naïfs et utilisée des espaces de représentation respectivement issus de processus de SdV préalables réalisés par les algorithmes ReliefF, CFS, MIFS

6. Ces jeux de données sont les jeux : GERMAN, MUSHROOMS (noté MUSH.), SICK, VEHICLE, ADULT, MONKS 3, FLAGS, BREAST CANCER (noté BREAST), ZOO, WINE, CANCER, PIMA, WAVE, CONTRACEPTION (noté CONTRA.), ION, SPAM, HOUSE-VOTES84 (noté HVOTES). Ils sont présentés en annexe (voir page 217). De plus, toutes les variables numériques ont subi un processus de discrétisation supervisée par l'intermédiaire de la méthode FUSINTER [ZRR98].

(ces 3 méthodes constituant des méthodes de référence du domaine), par notre algorithme de SdV, ou encore sans sélection préalable. Divers apprentissages ont permis la réalisation d'une étude comparative concernant d'une part le taux d'erreur des divers apprentissages selon la méthode de SdV employée et d'autre part le nombre de variables sélectionnées par chaque méthode de SdV. Cette évaluation est menée pour une 10-cross-validation ainsi que pour cinq 2-cross-validations. Notons de plus que :

- La version de CFS utilisée est telle que le critère employé est bien le critère classique (voir [Hal00b]) et la stratégie de recherche est basée sur un AG et non sur une simple approche de type best first.
- La version de MIFS employée est la version classique (critère classique, voir [Bat94], et stratégie de recherche gloutonne classique).
- La version de ReliefF employée est telle que : le critère employé est bien le critère à la fois de consistance et contextuel classique ; la stratégie de recherche utilise quant à elle un échantillon d'objets de la taille de l'ensemble des objets du jeu de données.
- CFS, MIFS et notre méthode fournissent quant à elles le sous-ensemble optimal de variables (ou un sous-ensemble l'approchant).
- ReliefF fournit la liste des variables classées selon leur pertinence (nous avons ensuite étudié cette liste de valeur afin de déterminer le sous-ensemble de variables apparemment le plus intéressant).
- l'AG utilisé pour CFS et notre méthode est une version élitiste des AGs de base, il est paramètré de la manière suivante : *nb de générations =2000, taille de la population = 30, Proba. Croisement = 0.98, Proba. mutation = 0.3* .

5.2.4.2 Analyse de l'Evaluation Expérimentale

Les résultats des expériences sont regroupés au sein des tableaux 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17. Les résultats présentés dans les 11 derniers tableaux sont présentés de manière graphique dans les figures 5.6, 5.7, 5.8, 5.9, 5.10, 5.11.

Les tableaux 5.5, 5.6, 5.7 (voir page 130) ainsi que la figure 5.6 (voir page 131) présentent des résultats généraux :

- Les tableaux 5.5, 5.6 permettent d'évaluer le comportement général des diverses méthodes d'apprentissage utilisées lorsqu'elles sont associées aux méthodes de SdV. En effet, ils présentent la valeur moyenne du rapport "taux de succès avec sélection / taux de succès sans sélection" pour chaque méthode d'apprentissage associée à chacune des méthodes de SdV, et ce, soit dans le cadre d'une 10-cross-validation (tableau 5.5), soit dans le cadre de cinq 2-cross-validations (tableau 5.6) (la moyenne est calculée sur l'ensemble des 17 jeux de données). Les résultats permettent de conclure que, de manière générale, l'ensemble des méthodes de SdV impliquent l'obtention de taux de succès quasi-équivalent lorsqu'on utilise les variables fournies par ces méthodes ou l'ensemble complet des

variables. Ainsi, quelle que soit la méthode d'apprentissage utilisée et quelle que soit la méthode de SdV utilisée, les taux de succès sont corrects et quasiment similaires. On peut toutefois noter un très léger déficit de qualité d'apprentissage pour la méthode d'apprentissage Sipina lorsqu'elle est associée à la méthode de CFS. On peut ainsi conclure que de manière générale ces 4 méthodes de SdV sont presque équivalentes du point de vue de la qualité des apprentissages qu'elles impliquent.

- Le tableau 5.7 et la figure 5.6 permettent l'évaluation de la réduction de la taille de l'ERD impliquée par l'utilisation des méthodes de SdV. Ainsi, il apparaît clairement que l'ensemble de ces méthodes permettent une réduction significative de la taille de l'ERD. De plus, il existe ici des distinctions claires entre les méthodes de SdV :
  - CFS réduit de manière générale très significativement la taille de cet espace puisqu'en moyenne elle ne conserve que 41,4% des variables. Elle constitue la méthode la plus efficace pour la réduction de l'ERD : son apparente plus grande capacité à réduire cet espace n'est mise en défaut que sur quelques rares jeux de données.
  - Notre méthode et MIFS permettent également, en général, de réduire significativement la taille de cet espace puisqu'en moyenne elles ne conservent respectivement que 56,9% et 62,6% des variables. Elles constituent, derrière CFS les méthodes les plus efficaces pour la réduction de l'ERD. Leur proximité en moyenne sur leur capacité à réduire l'ERD ne reflète cependant pas leurs comportements largement différents : selon le jeu de données, il peut arriver que l'une surpasse fortement l'autre dans sa capacité à réduire l'ERD. On peut ainsi conclure que si notre méthode semble légèrement plus efficace que MIFS de ce point de vue, il est par contre clair que ponctuellement ce résultat peut être inversé.
  - La méthode ReliefF, même si elle permet de réduire l'ERD (74,4% des variables conservées en moyenne), semble cependant en retrait par rapport aux autres méthodes.
- Du point de vue du coût calculatoire, MIFS, CFS et notre méthode nécessitent un temps de calcul proche avec un avantage toutefois à MIFS qui utilise une stratégie de recherche gloutonne contrairement aux 2 autres méthodes (temps de calcul de l'ordre de quelques secondes à la minute selon les jeux de données). En effet, les AGs sont, en principe, plus lents que les méthodes d'optimisation gloutonnes telle que celle employée dans MIFS. En fait, CFS et notre méthode pourraient être plus rapides si nous remplacions l'AG par une telle méthode d'optimisation (bien que dans ce cas nous pourrions obtenir des résultats de moindre qualité du point de vue de la correction en prédiction, nous ne pensons pas que la réduction de qualité associée soit réellement significative, et envisageons actuellement de tester cette approche...). ReliefF, par contre, implique un

temps de calcul plus important (parfois plusieurs minutes) ce qui s'explique par les multiples passes sur le jeu de données que cette méthode implique contrairement aux 3 autres méthodes.

Les tableaux 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17, ainsi que les figures 5.7, 5.8, 5.9, 5.10, 5.11 permettent d'appréhender de manière plus précise (sur des cas particuliers) les résultats. Les tableaux présentent sur chaque cas isolé (réalisation d'une 10-cross-validation ou de cinq 2-cross-validations pour une méthode d'apprentissage particulière) le taux d'erreur moyen en validation ainsi que l'écart-type de ce taux d'erreur. Les figures se contentent de présenter le taux d'erreur moyen en validation.

Les points les plus intéressants que l'on peut extraire d'une analyse détaillée de ces résultats sont :

- que la tendance générale de taux de correction proche pour les apprentissages réalisés avec et sans SdV est vérifiée localement,
- que CFS semble impliquer parfois des déficits importants en terme de correction et notamment lorsqu'elle sélectionne un nombre faible de variables (le cas du jeu de données MONKS 3 par exemple),
- que, tout comme pour la réduction de l'ERD, la méthode MIFS et la notre sont en général proches mais il arrive ponctuellement que l'une surpasse plus fortement l'autre.
- que, la stabilité des apprentissages est quasiment similaire pour les apprentissages sur un même jeu de données que l'on ait utilisé ou non la SdV et quelle que soit la méthode de SdV employée.

En définitive, selon nous, cette étude expérimentale tend à privilégier l'utilisation de CFS par rapport à MIFS et notre méthode et que l'on peut rejeter l'idée d'employer ReliefF sans trop de soucis. Toutefois, le coût calculatoire faible de CFS, MIFS et notre méthode, associé à l'unique passe sur les données qu'elles nécessitent, ainsi que la variabilité "ponctuelle" des résultats (déficit en terme de correction parfois important pour CFS, différentiel en terme de correction et de nombre de variables sélectionnées parfois significatif entre MIFS et notre méthode), semblent plaider en faveur d'une utilisation simultanée de ces 3 méthodes.

	ID3	C4.5	Sipina	B. Naïfs	1-PPV
Notre Méthode	0.9987	1.0001	0.9842	0.9951	1.0121
MIFS	1.0044	1.0086	0.9961	1.0030	1.0070
CFS	0.9951	0.9935	0.9679	0.9957	0.9955
ReliefF	0.9966	0.9999	0.9936	1.0011	1.0055

TAB. 5.5 –: *Evaluation des Méthodes de SdV pour une 10-Cross-Validation*

	ID3	C4.5	Sipina	B. Naïfs	1-PPV
Notre Méthode	0.9960	1.0046	1.0074	1.0193	1.0042
MIFS	1.0030	1.0086	1.0024	1.0118	1.0078
CFS	0.9863	0.9988	0.9879	1.0351	1.0199
ReliefF	0.9928	1.0014	0.9997	1.0102	1.0107

TAB. 5.6 –: *Evaluation des Méthodes de SdV pour cinq 2-Cross-Validations*

	sans SdV	Notre méthode	MIFS	ReliefF	CFS
GERMAN	20	$6^{30\%}$	$3^{15\%}$	$14^{70\%}$	$5^{25\%}$
MUSH.	22	$8^{36.36\%}$	$1^{4.55\%}$	$17^{77.27\%}$	$3^{13.64\%}$
SICK	28	$6^{21.43\%}$	$9^{32.14\%}$	$12^{42.86\%}$	$1^{3.57\%}$
VEHICLE	18	$12^{66.67\%}$	$6^{33.33\%}$	$18^{100\%}$	$10^{55.56\%}$
ADULT	14	$7^{50\%}$	$5^{35.71\%}$	$6^{42.86\%}$	$5^{35.71\%}$
MONKS 3	6	$3^{50\%}$	$6^{100\%}$	$2^{33.33\%}$	$1^{16.67\%}$
FLAGS	28	$14^{50\%}$	$21^{75\%}$	$27^{96.43\%}$	$3^{10.71\%}$
BREAST	9	$8^{88.89\%}$	$9^{100\%}$	$4^{44.44\%}$	$9^{100\%}$
ZOO	16	$12^{75\%}$	$16^{100\%}$	$14^{87.5\%}$	$9^{56.25\%}$
WINE	13	$11^{84.62\%}$	$13^{100\%}$	$11^{84.62\%}$	$9^{69.23\%}$
CANCER	9	$8^{88.89\%}$	$9^{100\%}$	$9^{100\%}$	$9^{100\%}$
PIMA	8	$2^{25\%}$	$4^{50\%}$	$7^{87.5\%}$	$3^{37.5\%}$
WAVE	21	$15^{71.43\%}$	$21^{100\%}$	$19^{90.48\%}$	$15^{71.43\%}$
CONTRA.	9	$2^{22.22\%}$	$2^{22.22\%}$	$2^{22.22\%}$	$5^{55.56\%}$
ION	34	$25^{73.53\%}$	$13^{38.24\%}$	$33^{97.06\%}$	$9^{26.47\%}$
SPAM	57	$25^{43.86\%}$	$51^{89.47\%}$	$57^{100\%}$	$12^{21.05\%}$
HVOTES	16	$10^{62.5\%}$	$11^{68.75\%}$	$14^{87.5\%}$	$1^{6.25\%}$
moyenne	19.29	$10.24^{56.9\%}$	$11.76^{62.6\%}$	$15.65^{74.4\%}$	$6.41^{41.4\%}$

TAB. 5.7 –: *Evaluation des Méthodes de SdV sur 17 jeux de données de la collection de l'UCI : Nombre de variables sélectionnées*$^{\%\ de\ variables\ sélectionées}$

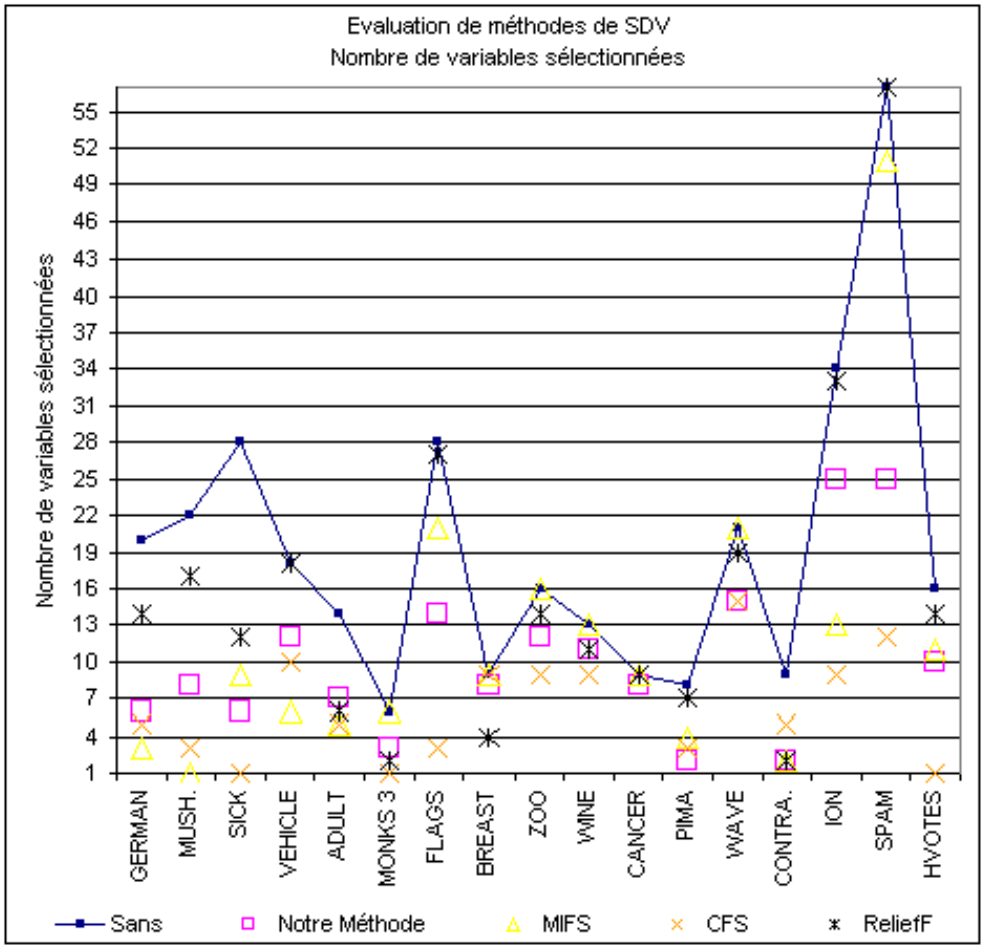


FIG. 5.6 –: *Evaluation Expérimentale de Méthodes de SdV*

5.2.5 Conclusion

En résumé, nous proposons, une méthode :

- basée sur l'hypothèse que l'espace de représentation des données doit être tel que le concept à apprendre doit impliquer qu'une cns représentant ce concept soit valide dans cet espace ;
- ne nécessitant qu'une unique passe sur le jeu de données, et une complexité algorithmique faible ce qui lui confère une rapidité très intéressante ;
- utilisant la méthodologie préalablement introduite pour la comparaison de la validité de cns ;
- utilisant un AG et une nouvelle fonction de fitness particulière afin de résoudre le problème combinatoire de la recherche du sous espace de EV

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$31.2^{3.37}$	$31.2^{3.37}$	$31.7^{3.63}$	$31.7^{4.56}$	$30.8^{3.66}$
MUSH.	$0.07^{0.13}$	$0.1^{0.09}$	$1.48^{0.32}$	$1.03^{0.29}$	0^{0}
SICK	$2.36^{1.11}$	$3.25^{1.06}$	$2.93^{1.1}$	$3.25^{0.85}$	$2.79^{0.94}$
VEHICLE	$33.7^{4.64}$	$32.51^{3.34}$	$31.92^{3.34}$	$32.74^{4.48}$	$33.7^{4.64}$
ADULT	$15.03^{0.89}$	$14.9^{0.61}$	$17.83^{2.84}$	$14.81^{0.08}$	$15.61^{0.63}$
MONKS 3	0^{0}	$2.76^{2.66}$	0^{0}	$19.44^{5.18}$	$2.78^{2.9}$
FLAGS	$49.53^{7.23}$	$48.37^{8.71}$	$45.26^{7.59}$	$42.18^{10.71}$	$51.13^{9.29}$
BREAST	$9.3^{2.23}$	$9.01^{2.22}$	$9.3^{2.23}$	$9.3^{2.23}$	$5.58^{3.35}$
ZOO	$26.64^{10.62}$	$26.76^{10.93}$	$26.64^{10.62}$	$26.73^{7.75}$	$26.82^{14.96}$
WINE	$8.46^{3.83}$	$8.43^{6.42}$	$8.46^{3.83}$	$8.4^{7.16}$	$9.51^{6.07}$
CANCER	$7.47^{2.73}$	$8.48^{2.48}$	$7.47^{2.73}$	$7.47^{2.73}$	$7.47^{2.73}$
PIMA	$25.26^{3.22}$	$25.27^{2.59}$	$25.26^{3.93}$	$25.28^{6.14}$	$25.26^{4.42}$
WAVE	$27.36^{1.37}$	$27.66^{2.19}$	$27.36^{1.37}$	$27.74^{2.17}$	$28.14^{1.55}$
CONTRA.	$52^{3.83}$	$51.94^{4.2}$	$51.8^{2.25}$	$53.7^{3.73}$	$52.01^{2.31}$
ION	$10.83^{6.98}$	$10.5^{5.96}$	$8.26^{5.92}$	$8.56^{3.14}$	$11.13^{6.58}$
SPAM	$11.52^{1.38}$	$13.17^{1.86}$	$12.19^{1.16}$	$12.45^{1.32}$	$11.52^{1.38}$
HVOTES	$4.37^{3.62}$	$4.38^{2.64}$	$4.39^{2.84}$	$4.35^{2.79}$	$5.05^{3.36}$

TAB. 5.8 –: *Evaluation des Méthodes de SdV avec ID3 pour une 10-Cross-Validation*
Légende : a^y a = moy. taux d'erreur pour une 10-cross-validation, y écart type taux d'erreur

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$30.54^{0.47}$	$30.52^{0.44}$	$30.3^{0.25}$	$30.5^{0.28}$	$30.8^{0.53}$
MUSH.	$0.26^{0.05}$	$0.41^{0.11}$	1.48^{0}	$1.04^{0.03}$	$0.23^{0.04}$
SICK	$3.19^{0.17}$	3.25^{0}	3.25^{0}	3.25^{0}	3.25^{0}
VEHICLE	$37.64^{0.62}$	$38.06^{1.54}$	$37.3^{1.99}$	$37.02^{1.63}$	$37.64^{0.62}$
ADULT	$15.34^{0.16}$	$15.23^{0.08}$	$16.92^{0.4}$	$15.06^{0.08}$	$16.36^{0.06}$
MONKS 3	$5^{0.06}$	$4.72^{0.24}$	$5^{0.06}$	19.44^{0}	$5.09^{0.33}$
FLAGS	$52.99^{1.71}$	$53.3^{3.52}$	$52.59^{2.55}$	$53.4^{4.41}$	$55.05^{4.25}$
BREAST	$8.56^{0.48}$	$8.38^{0.5}$	$8.56^{0.48}$	$8.56^{0.48}$	$8.61^{0.55}$
ZOO	$27.88^{0.95}$	$31.07^{5.62}$	$27.88^{0.95}$	$31.71^{7.02}$	$27.11^{0.81}$
WINE	$14.49^{1.15}$	$18.2^{0.98}$	$14.49^{1.15}$	$19.33^{1.04}$	$20.22^{3.52}$
CANCER	$8.37^{0.55}$	$7.82^{0.65}$	$8.37^{0.55}$	$8.37^{0.55}$	$8.37^{0.55}$
PIMA	$26.43^{1.44}$	$26.35^{1.34}$	25.26^{0}	$25.78^{1.04}$	25.26^{0}
WAVE	$29.84^{1.38}$	$30.14^{0.66}$	$29.84^{1.38}$	$30.04^{0.52}$	$29.65^{0.56}$
CONTRA.	$53.22^{0.39}$	$53.51^{0.84}$	$53.4^{0.93}$	$53.33^{1.09}$	$53.81^{0.59}$
ION	$18.52^{1.39}$	$20.06^{3.99}$	$17.67^{1.47}$	$19.43^{2.23}$	$18.98^{1.32}$
SPAM	$17.45^{0.2}$	$13.06^{1.06}$	$13.66^{0.34}$	$13.67^{0.35}$	$17.45^{0.2}$
HVOTES	$4.64^{0.55}$	$4.92^{0.69}$	$4.87^{0.63}$	$5.24^{0.72}$	$5.15^{1.08}$

TAB. 5.9 –: *Evaluation des Méthodes de SdV avec ID3 pour cinq 2-Cross-Validations*
Légende : a^y a = moy. taux d'erreur pour cinq 2-cross-validation, y écart type taux d'erreur

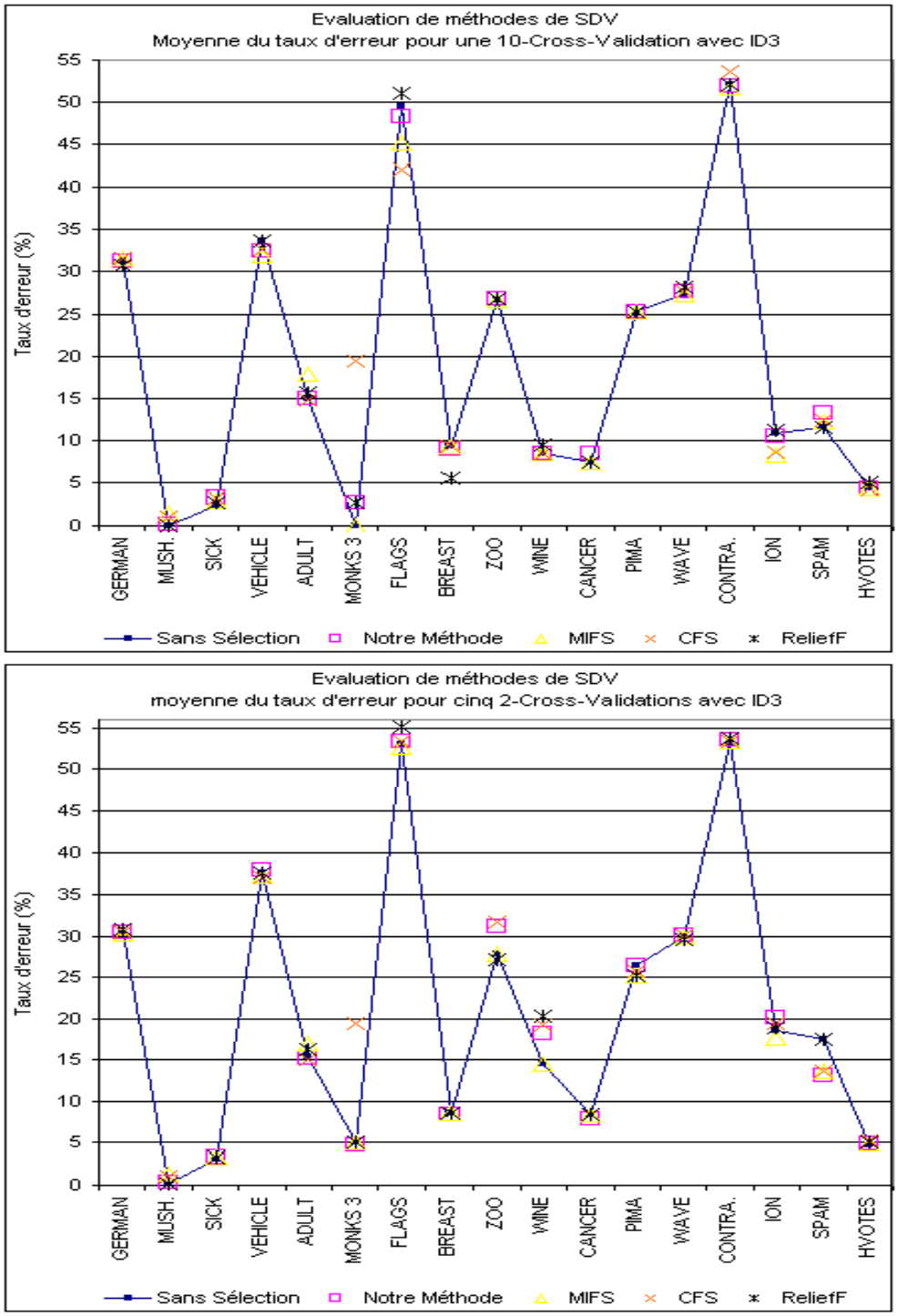


FIG. 5.7 –: *Evaluation Expérimentale de Méthodes de SdV*

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$29^{4.1}$	$27^{5.04}$	$26.1^{3.24}$	$25.7^{4.05}$	$28.3^{4.31}$
MUSH.	0^{0}	0^{0}	$1.48^{0.43}$	$0.98^{0.34}$	0^{0}
SICK	$2.14^{0.81}$	$3.25^{0.91}$	$2.32^{1.01}$	$3.25^{0.82}$	$2.57^{0.69}$
VEHICLE	$32.4^{4.67}$	$32.15^{4.42}$	$32.5^{1.87}$	$34.17^{5.2}$	$32.4^{4.67}$
ADULT	$14.4^{0.59}$	$14.23^{0.64}$	$14.24^{0.49}$	$14.42^{0.54}$	$15.07^{0.37}$
MONKS 3	0^{0}	$2.77^{2.02}$	0^{0}	$19.47^{6.23}$	$2.77^{2.71}$
FLAGS	$34.03^{10.56}$	$27.71^{10.12}$	$28.84^{10.31}$	$28.37^{11.41}$	$31.97^{9.03}$
BREAST	$5.43^{2.28}$	$5.43^{2.1}$	$5.43^{2.28}$	$5.43^{2.28}$	$3.72^{1.31}$
ZOO	$7^{6.4}$	$10.73^{9.97}$	$7^{6.4}$	$8.91^{9.43}$	$5.82^{7.69}$
WINE	$6.14^{6.31}$	$4.54^{5.67}$	$6.14^{6.31}$	$6.14^{3.89}$	$8.4^{7.53}$
CANCER	$5.27^{2.64}$	$4.82^{1.94}$	$5.27^{2.64}$	$5.27^{2.64}$	$5.27^{2.64}$
PIMA	$26.3^{5.07}$	$26.04^{3.26}$	$22.4^{3.98}$	$22.13^{4.57}$	$24.99^{6.03}$
WAVE	$25.5^{1.88}$	$26.12^{1.62}$	$25.5^{1.88}$	$25.98^{1.78}$	$25.92^{1.63}$
CONTRA.	$50.71^{3.6}$	$51.87^{3.83}$	$51.26^{4.06}$	$52.61^{2.6}$	51.53^{3}
ION	$8.55^{3.13}$	$8.54^{3.59}$	$8.29^{5.02}$	$8.27^{4.7}$	$9.42^{4.26}$
SPAM	$7.44^{0.68}$	$11.8^{1.64}$	$7.65^{0.6}$	$8.8^{0.83}$	$7.44^{0.68}$
HVOTES	$6.22^{3.88}$	$5.74^{3.25}$	$6.19^{2.89}$	$4.38^{3.34}$	$5.83^{3.81}$

TAB. 5.10 –: *Evaluation des Méthodes de SdV avec C4.5 pour une 10-Cross-Validation*

Légende : a^y a = moy. taux d'erreur pour la 10-cross-validation, y écart type taux d'erreur

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$30.54^{0.47}$	28.42^{1}	$27.48^{1.3}$	$26.88^{0.89}$	$29.24^{1.35}$
MUSH.	$0.26^{0.05}$	0^{0}	1.48^{0}	$1^{0.04}$	$0.03^{0.04}$
SICK	$3.19^{0.17}$	3.25^{0}	$2.42^{0.11}$	3.25^{0}	$2.63^{0.07}$
VEHICLE	$37.64^{0.62}$	$34.66^{1.96}$	$33.83^{1.92}$	$34.6^{1.55}$	$34.18^{1.97}$
ADULT	$15.34^{0.16}$	$14.34^{0.08}$	$14.25^{0.02}$	$14.47^{0.01}$	$15.27^{0.06}$
MONKS 3	$5^{0.06}$	2.78^{0}	0^{0}	19.44^{0}	$3.29^{1.02}$
FLAGS	$52.99^{1.71}$	$34.85^{1.58}$	$36.91^{3.27}$	$33.4^{3.05}$	$35.46^{3.5}$
BREAST	$8.56^{0.48}$	$5.32^{0.82}$	$5.41^{0.7}$	$5.41^{0.7}$	$5.38^{0.79}$
ZOO	$27.88^{0.95}$	$13.29^{3.23}$	$13.85^{2.44}$	$13.29^{1.48}$	$11.68^{2.69}$
WINE	$14.49^{1.15}$	$8.99^{2.22}$	$8.43^{2.59}$	$7.64^{3.07}$	$7.08^{2.48}$
CANCER	$8.37^{0.55}$	$5.13^{0.71}$	$6.76^{0.3}$	$6.76^{0.3}$	$6.76^{0.3}$
PIMA	$26.43^{1.44}$	$25.16^{0.46}$	$23.2^{1.14}$	$23.83^{1.11}$	24.24^{1}
WAVE	$29.84^{1.38}$	$27.36^{0.27}$	$27.82^{0.48}$	$27.59^{0.61}$	$27.56^{0.66}$
CONTRA.	$53.22^{0.39}$	$52.75^{1.69}$	$51.31^{0.55}$	53.21^{1}	$53.22^{1.12}$
ION	$18.52^{1.39}$	$9.12^{1.23}$	$9.17^{0.83}$	$9.12^{0.74}$	$10.03^{2.44}$
SPAM	$17.45^{0.2}$	$12.3^{0.36}$	$8.77^{0.22}$	$9.71^{0.36}$	$15.38^{0.26}$
HVOTES	$4.64^{0.55}$	$4.87^{0.53}$	$6.67^{0.99}$	$4.6^{0.46}$	$5.29^{0.69}$

TAB. 5.11 –: *Evaluation des Méthodes de SdV avec C4.5 pour cinq 2-Cross-Validations*

Légende : a^y a = moy. taux d'erreur pour cinq 2-cross-validations, y écart type taux d'erreur

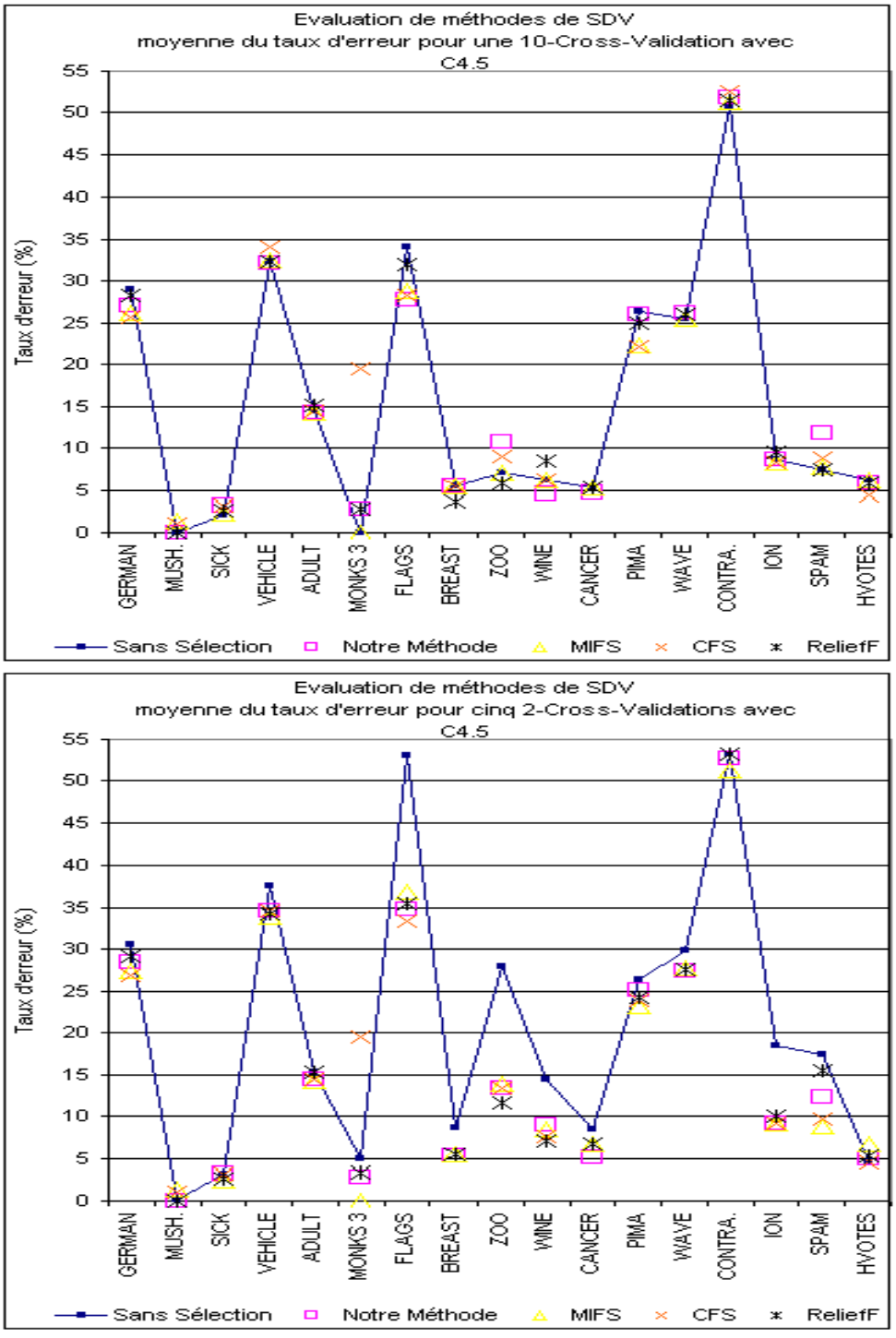


FIG. 5.8 –: *Evaluation Expérimentale de Méthodes de SdV*

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$29.9^{4.5}$	$28.9^{6.01}$	$25.3^{2.83}$	$26.2^{3.49}$	$30.8^{4.21}$
MUSH.	$0.62^{0.17}$	$0.1^{0.12}$	$1.48^{0.28}$	$0.98^{0.35}$	$0.57^{0.21}$
SICK	$2.64^{0.99}$	$3.25^{1.18}$	$2.96^{1.21}$	$3.25^{1.03}$	$2.64^{0.8}$
VEHICLE	$43.51^{1.81}$	$47.86^{6.89}$	$44.8^{4.93}$	$48.57^{6.84}$	$43.51^{1.81}$
ADULT	$17.21^{0.64}$	$17.78^{0.61}$	$17.55^{0.48}$	$17.94^{0.02}$	$20.82^{0.81}$
MONKS 3	0^{0}	$2.77^{1.72}$	0^{0}	$19.46^{6.28}$	$2.78^{2.7}$
FLAGS	$46.37^{11.36}$	$49.58^{9.83}$	$50.11^{7.97}$	$48.92^{10.39}$	$49.97^{12.7}$
BREAST	$6.87^{3.07}$	$5.58^{3.58}$	$6.87^{3.07}$	$6.87^{3.07}$	$5.58^{2.59}$
ZOO	$13.82^{7.88}$	$16.64^{12.06}$	$13.82^{7.88}$	$26.82^{9.16}$	$10.82^{11.23}$
WINE	$7.22^{7.88}$	$7.94^{6.94}$	$7.22^{7.88}$	$8.43^{6.23}$	$6.7^{4.84}$
CANCER	$4.68^{3.44}$	$4.98^{2.54}$	$4.68^{3.44}$	$4.68^{3.44}$	$4.68^{3.44}$
PIMA	$26.05^{5.41}$	$25.26^{5.94}$	$23.04^{4.4}$	$23.44^{5.54}$	$24.73^{3.08}$
WAVE	$23.76^{1.48}$	$24.54^{1.3}$	$23.76^{1.48}$	$27.49^{0.76}$	$24.08^{1.72}$
CONTRA.	$56.29^{5.52}$	$58.32^{3.84}$	$58.86^{2.32}$	$59.67^{3.2}$	$57.29^{2.55}$
ION	$12.26^{5.14}$	$12.24^{3.56}$	$12.26^{5.29}$	$10.55^{5.59}$	$12.52^{4.56}$
SPAM	$10.68^{1.21}$	$13.78^{0.95}$	$11.13^{1.72}$	$11.76^{1.67}$	$10.68^{1.21}$
HVOTES	$4.38^{2.18}$	$4.38^{3.91}$	$4.36^{3.14}$	$4.36^{3.14}$	$4.35^{2.43}$

TAB. 5.12 –: *Evaluation des Méthodes de SdV avec Sipina pour une 10-Cross-Validation*

Légende : a^y a = moy. taux d'erreur pour la 10-cross-validation, y écart type taux d'erreur

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$29.86^{0.71}$	$30.34^{0.73}$	$26.7^{0.7}$	$27.24^{0.73}$	$29.68^{1.12}$
MUSH.	$0.48^{0.16}$	0.1^{0}	1.48^{0}	$1.3^{0.1}$	$0.42^{0.13}$
SICK	$2.41^{0.04}$	3.25^{0}	$3.31^{0.29}$	3.25^{0}	$2.66^{0.1}$
VEHICLE	$46.48^{0.96}$	$49.6^{0.88}$	$50.47^{3.02}$	$47.57^{1.2}$	$46.48^{0.96}$
ADULT	$17.5^{0.15}$	$17.83^{0.05}$	$17.68^{0.06}$	17.92^{0}	$20.84^{0.06}$
MONKS 3	$3.19^{1.13}$	$3.1^{0.65}$	$3.19^{1.13}$	19.44^{0}	2.78^{0}
FLAGS	$54.85^{1.06}$	$49.69^{1.68}$	$54.64^{3.56}$	$50.72^{3.51}$	$54.95^{3.82}$
BREAST	$6.32^{0.6}$	$6.44^{0.45}$	$6.32^{0.6}$	$6.32^{0.6}$	$5.95^{0.47}$
ZOO	$18.23^{0.8}$	$18.41^{1.36}$	$18.23^{0.8}$	$26.92^{0.4}$	$19.4^{1.18}$
WINE	$18.76^{2.09}$	$18.43^{1.79}$	$18.76^{2.09}$	$18.2^{0.98}$	$18.43^{2.9}$
CANCER	$6.53^{0.73}$	$5.89^{0.19}$	$6.53^{0.73}$	$6.53^{0.73}$	$6.53^{0.73}$
PIMA	$26.12^{1.09}$	$25.03^{0.46}$	$23.26^{1.17}$	$24.64^{0.79}$	$25.36^{0.64}$
WAVE	$26.64^{1.08}$	$27.08^{1.26}$	$26.64^{1.08}$	$26.78^{0.79}$	$27.89^{0.69}$
CONTRA.	$58.38^{0.62}$	$57.56^{0.2}$	$58.75^{0.75}$	$59.23^{0.75}$	57.3^{0}
ION	$12.25^{1.62}$	$12.31^{0.38}$	$12.25^{1.1}$	$10.94^{0.74}$	$11.11^{0.31}$
SPAM	$16.93^{0.33}$	$13.87^{0.22}$	$11.46^{0.61}$	$12.87^{0.18}$	$16.93^{0.33}$
HVOTES	$7.22^{3.54}$	$6.11^{2.13}$	$7.95^{1.77}$	$15.22^{8.9}$	$6.66^{3.26}$

TAB. 5.13 –: *Evaluation des Méthodes de SdV avec Sipina pour cinq 2-Cross-Validations*

Légende : a^y a = moy. taux d'erreur pour cinq-2-cross-validations, y écart type taux d'erreur

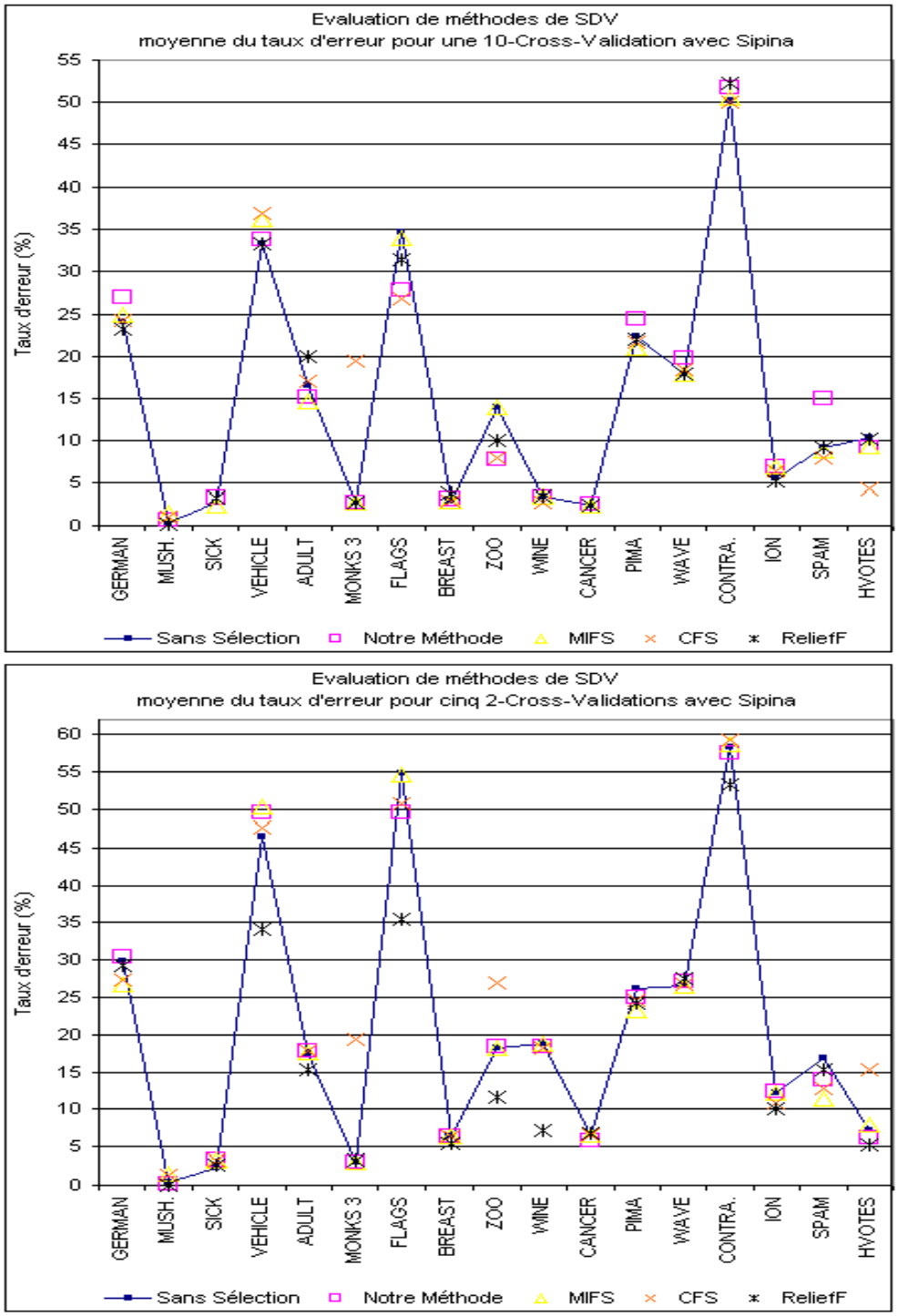


FIG. 5.9 –: *Evaluation Expérimentale de Méthodes de SdV*

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$24^{3.6}$	$26.9^{2.39}$	$25^{4.65}$	$24.2^{2.71}$	$23.2^{4.31}$
MUSH.	$0.31^{0.19}$	$0.65^{0.58}$	$1.48^{0.37}$	$1.08^{0.3}$	$0.23^{0.21}$
SICK	$2.82^{1.45}$	$3.29^{0.98}$	$2.46^{1.18}$	$3.25^{1.25}$	$3.32^{0.89}$
VEHICLE	$33.32^{4.56}$	$33.8^{5.09}$	$36.28^{5.96}$	$37.01^{3.71}$	$33.32^{4.56}$
ADULT	$16.45^{0.72}$	$15.11^{0.9}$	$14.55^{0.57}$	$17.08^{0.54}$	$19.86^{0.78}$
MONKS 3	$2.79^{2.71}$	$2.78^{0.94}$	$2.79^{2.71}$	$19.46^{4.22}$	$2.79^{3.42}$
FLAGS	$34.73^{11.8}$	$27.82^{5.99}$	$34.05^{6.66}$	$26.76^{10.27}$	$31.37^{14.21}$
BREAST	$3^{2.16}$	$3.14^{2.7}$	$3^{2.16}$	$3^{2.16}$	$3.86^{2.48}$
ZOO	$14^{11.14}$	$7.73^{9.2}$	$14^{11.14}$	$7.91^{9.78}$	$9.91^{7.75}$
WINE	$3.4^{4.56}$	$3.33^{3.69}$	$3.4^{4.56}$	$2.78^{3.73}$	$3.4^{2.78}$
CANCER	$2.49^{2.08}$	$2.63^{2.04}$	$2.49^{2.08}$	$2.49^{2.08}$	$2.49^{2.08}$
PIMA	$22.26^{4.13}$	$24.34^{6.39}$	$20.96^{3.11}$	$21.74^{4.05}$	$22.01^{3.23}$
WAVE	$17.8^{1.49}$	$19.78^{1.43}$	$17.8^{1.49}$	$18.44^{2.59}$	$17.82^{2.01}$
CONTRA.	$50.16^{4.49}$	$51.74^{4.35}$	$50.51^{4.26}$	$50.1^{3.57}$	$52.32^{0.72}$
ION	$5.42^{2.99}$	$6.85^{3.19}$	$6.83^{3.65}$	$6.29^{5.54}$	$5.14^{3.08}$
SPAM	$9.06^{0.69}$	$15.02^{1.47}$	$8.76^{1.31}$	$7.87^{1.06}$	$9.06^{0.69}$
HVOTES	$10.34^{4.15}$	$9.18^{4.59}$	$9.2^{2.5}$	$4.37^{2.4}$	$10.2^{3.9}$

TAB. 5.14 –: *Evaluation des Méthodes de SdV avec B.Naïfs pour une 10-Cross-Validation*

Légende : a^y a = moy. taux d'erreur pour la 10-cross-validation, y écart type taux d'erreur

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$24.44^{0.74}$	$26.74^{0.3}$	$26.4^{1.17}$	$24.78^{0.53}$	$23.98^{0.7}$
MUSH.	$0.32^{0.02}$	$0.66^{0.04}$	1.48^{0}	1.08^{0}	$0.31^{0.05}$
SICK	$3.28^{0.24}$	3.32^{0}	$2.59^{0.09}$	3.25^{0}	$3.21^{0.14}$
VEHICLE	$34.54^{1.01}$	$36.03^{1.97}$	$36.78^{0.5}$	$38.53^{2.18}$	$34.54^{1.01}$
ADULT	$16.44^{0.05}$	$15.09^{0.03}$	$14.7^{0.19}$	$17.08^{0.01}$	$19.78^{0.03}$
MONKS 3	2.78^{0}	2.78^{0}	2.78^{0}	19.44^{0}	2.78^{0}
FLAGS	$44.43^{2.77}$	$36.8^{2.58}$	$43.4^{4.34}$	$25.77^{1.63}$	$40^{5.16}$
BREAST	$3^{0.37}$	$2.75^{0.06}$	$3^{0.37}$	$3^{0.37}$	$4.32^{0.11}$
ZOO	$25.7^{4.39}$	$15.67^{0.71}$	$25.7^{4.39}$	$11.9^{3.3}$	$14.29^{3.79}$
WINE	$9.55^{1.88}$	$7.19^{4.53}$	$9.55^{1.88}$	$8.31^{2.11}$	$8.09^{1.76}$
CANCER	$2.9^{0.32}$	$2.64^{0.16}$	$2.9^{0.32}$	$2.9^{0.32}$	$2.9^{0.32}$
PIMA	$22.5^{0.1}$	$24.53^{0.23}$	$21.02^{0.34}$	$21.54^{0.68}$	$21.2^{0.54}$
WAVE	$18.16^{0.19}$	$19.74^{0.1}$	$18.16^{0.19}$	$18.52^{0.1}$	$18.06^{0.17}$
CONTRA.	$49.75^{0.87}$	$52.19^{1.14}$	$51.84^{1.26}$	$50.63^{0.22}$	$53.21^{0.43}$
ION	$9.91^{1.78}$	$8.38^{1.26}$	$7.98^{1.03}$	$7.07^{0.73}$	$8.03^{0.8}$
SPAM	$24.97^{0.32}$	$15.3^{0.09}$	$9.17^{0.09}$	$7.91^{0.09}$	$24.97^{0.32}$
HVOTES	$10.12^{1.21}$	$8.83^{0.42}$	$8.69^{0.49}$	4.37^{0}	$10.02^{1.05}$

TAB. 5.15 –: *Evaluation des Méthodes de SdV avec B.Naïfs pour cinq 2-Cross-Validations*

Légende : a^y a = moy. taux d'erreur pour cinq 2-cross-validations, y écart type taux d'erreur

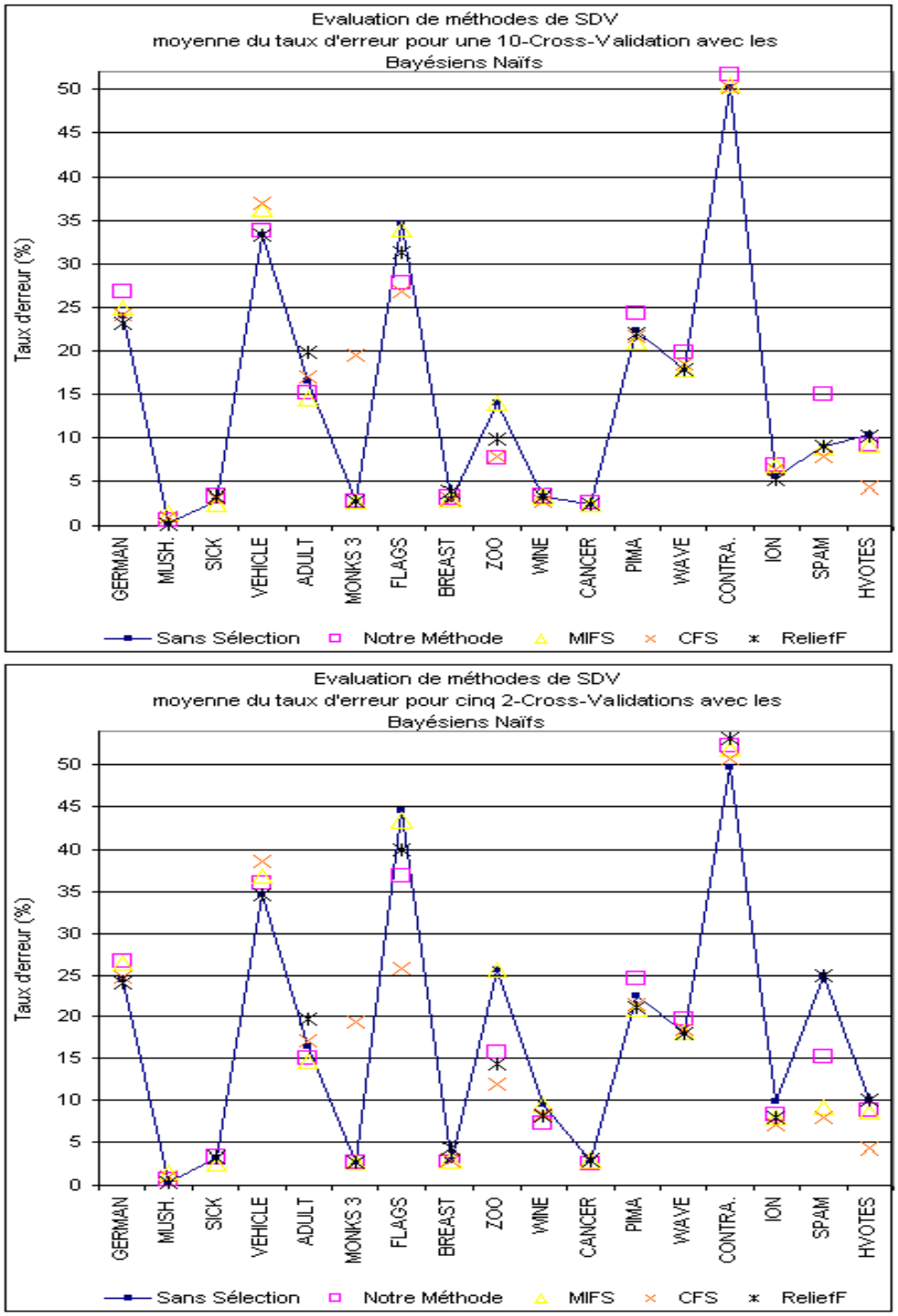


FIG. 5.10 –: *Evaluation Expérimentale de Méthodes de SdV*

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$31.6^{6.07}$	$35.1^{3.78}$	$34.9^{4.25}$	$33^{4.92}$	$31.8^{2.89}$
MUSH.	0^{0}	0^{0}	$1.48^{0.27}$	$0.98^{0.21}$	0^{0}
SICK	$2.79^{0.91}$	$3.68^{1.73}$	$6.11^{1.76}$	$3.25^{1.11}$	$2.39^{0.8}$
VEHICLE	$33.21^{2.73}$	$35.46^{2.88}$	$35.22^{5.72}$	$36.05^{3.21}$	$33.21^{2.73}$
ADULT	$20.19^{0.74}$	$20.02^{0.63}$	$20.05^{0.39}$	$22.16^{0.94}$	$21.66^{0.55}$
MONKS 3	$13.66^{4.68}$	$3.01^{1.48}$	$13.66^{4.68}$	$37.72^{4.39}$	$5.78^{3.31}$
FLAGS	$46.95^{10.21}$	$40.29^{10.92}$	$40.68^{13.85}$	$39.74^{8.25}$	$46.95^{7.73}$
BREAST	$5.58^{3.1}$	$5.29^{2.79}$	$5.58^{3.1}$	$5.58^{3.1}$	$5.72^{2.22}$
ZOO	$3.91^{6.56}$	$3^{4.58}$	$3.91^{6.56}$	$4^{4.9}$	$2.91^{4.45}$
WINE	$4.48^{4.17}$	$4.51^{3.36}$	$4.48^{4.17}$	$2.22^{3.69}$	$4.44^{6.94}$
CANCER	$5.42^{2.54}$	$7.03^{2.68}$	$5.42^{2.54}$	$5.42^{2.54}$	$5.42^{2.54}$
PIMA	$31.65^{4.64}$	$30.98^{4.13}$	$29.7^{5.54}$	$34.38^{5.6}$	33.32^{7}
WAVE	$40.48^{2.22}$	$38.4^{1.32}$	$40.48^{2.22}$	$38.42^{1.32}$	$40.04^{2.43}$
CONTRA.	$56.82^{4.49}$	$61.71^{2.7}$	$56.22^{3.12}$	$55.34^{4.84}$	$55.4^{3.73}$
ION	$13.97^{5.35}$	$10.83^{3.81}$	$14.56^{6.47}$	$12.52^{5.38}$	$12.79^{7.25}$
SPAM	$9^{0.91}$	$11^{1.37}$	$9.89^{0.87}$	$9.82^{1.09}$	$9^{0.91}$
HVOTES	$13.76^{4.37}$	$4.38^{2.21}$	$5.06^{5.21}$	$4.38^{2.84}$	$15.41^{4.5}$

TAB. 5.16 –: *Evaluation des Méthodes de SdV avec 1-PPV pour une 10-Cross-Validation*

Légende : a^y a = moy. taux d'erreur pour la 10-cross-validation, y écart type taux d'erreur

	sans SdV	Notre méthode	MIFS	CFS	ReliefF
GERMAN	$32.92^{1.49}$	$35^{2.83}$	$34.06^{1.15}$	$31.42^{1.33}$	$33.64^{0.39}$
MUSH.	0^{0}	0^{0}	$5.57^{8.19}$	$7.41^{7.87}$	$0.02^{0.04}$
SICK	$3.09^{0.43}$	$4.02^{0.67}$	$3.78^{0.78}$	$4.78^{0.92}$	$3.37^{1.96}$
VEHICLE	$36.41^{0.91}$	$37.92^{1.99}$	$36.41^{2.01}$	$37.71^{0.78}$	$36.41^{0.91}$
ADULT	$20.92^{0.13}$	$20.98^{1.18}$	$19.8^{1.28}$	$20.89^{1.01}$	$20.99^{0.27}$
MONKS 3	$12.13^{1.45}$	4.21^{1}	$12.13^{1.45}$	$27.55^{5.29}$	$4.54^{1.06}$
FLAGS	$48.14^{3.29}$	$46.08^{3.57}$	$48.87^{2.89}$	$32.68^{3.82}$	$50.31^{3.95}$
BREAST	$5.58^{0.6}$	$6.01^{0.68}$	$5.58^{0.6}$	$5.58^{0.6}$	$6.21^{1.07}$
ZOO	$7.13^{1.6}$	$7.15^{4.97}$	$7.13^{1.6}$	$4.56^{2.03}$	$6.53^{2.55}$
WINE	$8.88^{1.96}$	$4.94^{1.3}$	$8.88^{1.96}$	$3.15^{0.57}$	$4.38^{0.42}$
CANCER	$5.04^{0.48}$	$6.85^{0.98}$	$5.04^{0.48}$	$5.04^{0.48}$	$5.04^{0.48}$
PIMA	$31.87^{1.23}$	$36.38^{3.69}$	$26.93^{1.51}$	$28.65^{1.45}$	$30.89^{2.48}$
WAVE	$40.94^{0.51}$	$39.55^{0.43}$	$40.94^{0.51}$	$39.5^{0.62}$	$40.72^{0.72}$
CONTRA.	$59.02^{0.34}$	$60.45^{1.73}$	$57.8^{1.04}$	$56.13^{0.45}$	$55.84^{0.54}$
ION	$13.62^{0.75}$	$13.56^{0.84}$	$12.88^{0.49}$	$12.14^{1.16}$	$13.11^{0.75}$
SPAM	$10.55^{0.25}$	$14.44^{2.92}$	$10.82^{0.36}$	$10.95^{0.35}$	$10.55^{0.25}$
HVOTES	$13.89^{1.12}$	$4.6^{0.46}$	$4.83^{0.52}$	$8.82^{6.96}$	$13.84^{1.88}$

TAB. 5.17 –: *Evaluation des Méthodes de SdV avec 1-PPV pour cinq 2-Cross-Validations*

Légende : a^y a = moy. taux d'erreur pour cinq 2-cross-validations, y écart type taux d'erreur

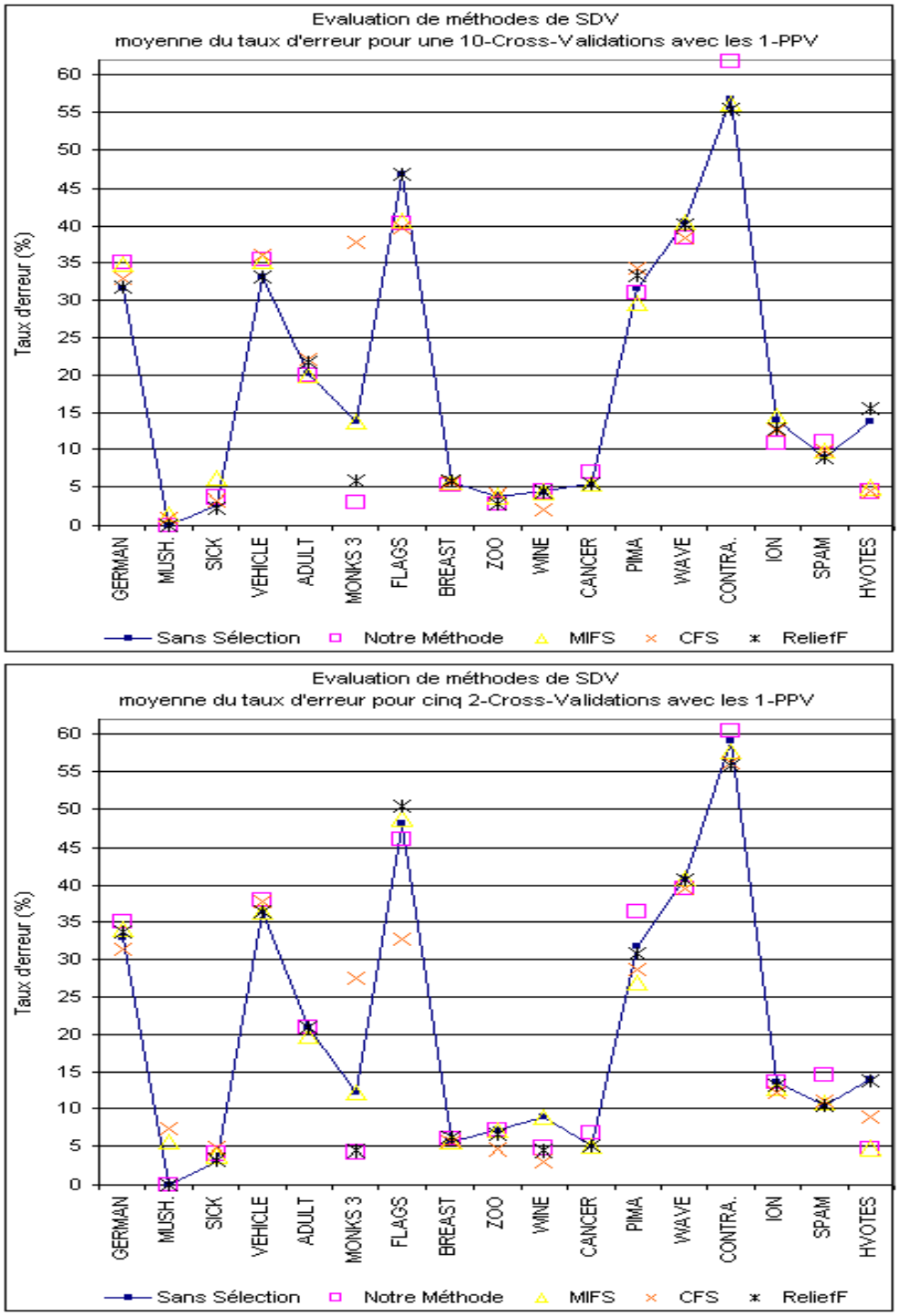


FIG. 5.11 –: *Evaluation Expérimentale de Méthodes de SdV*

impliquant la validité la plus forte pour P.

Les évaluations expérimentales ont montré que :

- concernant la précision prédictive, notre méthode se comporte, en général, comme les 3 autres méthodes testées (qui constituent des méthodes de référence du domaine),
- concernant le nombre de variables sélectionnées, la réduction du nombre de variables due à notre méthode est réelle même si elle est inférieure à celle impliquée par CFS,
- concernant le temps de calcul, notre méthode est un peu plus lente que MIFS qui est une méthode de sélection extrêmement rapide,
- qu'une utilisation simultanée des méthodes CFS, MIFS et la notre semble réalisable et judicieuse.

Nous pouvons également conclure que :

- le paradigme de sélection sous-jacent à notre méthode est relativement différent de ceux de MIFS et CFS et peut être mieux adapté à certains jeux de données ;
- notre méthode peut être améliorée du point de vue du coût calculatoire (voir ci dessous) ;
- on peut aisément modifier la structure de l'AG de manière à pouvoir rechercher non pas l'ensemble "optimal" de variables mais le meilleur ensemble de variables tel qu'il comprenne au plus un nombre fixé de variables, afin de réduire le nombre de variables sélectionnées.

Enfin, bien que l'hypothèse 5 (une classe par modalité du concept à apprendre, cf. page 119) soit forte, notre méthode fournit des résultats de qualité proche ou supérieure à ceux des méthodes existantes. Les travaux futurs seront dirigés vers :

- une amélioration de la méthode, par relaxation de l'hypothèse 5, la relaxation de cette hypothèse pouvant éventuellement être réalisée par modification de la fonction de fitness utilisée et en donnant notamment plus d'importance à l'aspect séparation des classes ($xv_2(P)^{EV_\star}$) par rapport à l'aspect homogénéité interne des classes ($xv_1(P)^{EV_\star}$);
- une réduction du temps de calcul associé par substitution d'une méthode d'optimisation gloutonne à l'AG.

REMARQUE : La méthode que nous venons de proposer peut également être employée si des variables quantitatives sont présentes, cependant, si elle ne nécessite également qu'une unique passe sur les données, sa complexité calculatoire est alors en $O(n^2)$.

5.3 Contribution à la Sélection de Variables pour l'Apprentissage Non Supervisé : Une Nouvelle Méthode Efficiente et Rapide

Si dans le cadre de l'apprentissage supervisé, la problématique de la SdV a été l'objet de nombreux travaux, le constat est radicalement opposé pour l'apprentissage non supervisé. En effet, si l'on exclut les méthodes d'analyse factorielle et celles issues du "multidimensional scaling", seules quelques rares approches ont été proposées [DL03], [DCSL02], [LJF02] (ces approches étant de type enveloppe à l'exception de [DCSL02]). Cela s'explique sans doute notamment par la relative difficulté à déterminer clairement ce que doit permettre la SdV dans ce cadre puisqu'en apprentissage non supervisé il n'existe pas, à proprement parlé et contrairement à l'apprentissage supervisé, de structure objectif connue et devant être extraite.

Selon nous, l'objectif de la SdV dans le cadre de l'apprentissage non supervisé est de réduire l'ERD de manière à ce que la validité de la meilleure cns obtenue par l'intermédiaire d'un algorithme donné appliqué sur un jeu de données complet (ERD complet), et, la validité de la meilleure cns obtenue par l'intermédiaire du même algorithme sur ce jeu de données réduit (ERD réduit) soient proches, ou, que la cns obtenue dans le second cas présente une meilleure validité. Ainsi, d'un point de vue pragmatique, et puisque les algorithmes de cns possèdent un coût calculatoire sensible à la dimension de l'ERD, l'objectif de la SdV sera principalement d'accélérer le temps de traitement nécessaire à la cns tout en assurant le maintien ou l'accroissement de la qualité (validité) de la structure extraite.

Nous proposons ici une méthode de SdV pour l'apprentissage non supervisé, cette méthode dérive d'une certaine manière de la méthode proposée pour l'apprentissage supervisé. En effet, dans le chapitre précédent nous recherchions le sous espace de l'ERD tel que la partition impliquée par la variable endogène apparaisse la plus valide possible, cette fois, tout se passe comme si il n'y avait pas une unique variable endogène mais p variables endogènes (les p variables de l'ERD) pour lesquelles nous devons rechercher le sous espace de l'ERD tel que les partitions impliquées par ces p variables apparaissent dans leur ensemble comme les plus valides. On considère donc ici l'apprentissage non supervisé comme un apprentissage supervisé de p concepts de manière simultanée.

Nous proposons donc une extension de la notion de la validité d'une partition : l'adéquation entre deux ensembles de variables, qui, dans le cadre de données catégorielles, correspond à l'adéquation entre deux ensembles de partitions (car les variables catégorielles définissent des partitions sur un ensemble d'objets). Cette notion permet de définir deux indices pour l'évaluation de l'adéquation entre un ensemble de variables et un sous ensemble de ce dernier. La caractérisation statistique de ces indices, similairement à celle des indices xv_1 et xv_2, mène à une méthodologie d'évaluation/comparaison de l'adéqua-

tion entre sous ensembles de variables et un ensemble particulier de variables. Cela nous permet, de manière identique à la méthodologie d'évaluation de la validité de cns, de dériver une méthode de SdV pour l'apprentissage non supervisé.

5.3.1 Evaluation de l'Adéquation entre deux Ensembles de Variables

Afin d'évaluer l'adéquation entre deux ensembles de variables catégorielles $EV_1 = \{V_{1_i}, i = 1..l\}$ et $EV_2 = \{V_{2_j}, j = 1..m\}$ nous utilisons un ensemble de 4 indices :

- $\mathbf{LL}(\mathbf{EV_1}, \mathbf{EV_2})$: qui comptabilise le nombre de couples formés d'un couple d'objets (o_a, o_b) et d'un couple de variables (V_{1_i}, V_{2_j}) formé d'une variable V_{1_i} de EV_1 et d'une variable EV_{2_j} de V_2 tels que :
  le couple d'objets (o_a, o_b) est caractérisé simultanément par un **lien** selon V_{1_i} et par un **lien** selon V_{2_j}.

$$LL(EV_1, EV_2) = \sum_{a=1..n} \sum_{b=a+1..n} \sum_{i=1..l} \sum_{j=1..m} lien_i(o_{a_i}, o_{b_i}) \times lien_j(o_{a_j}, o_{b_j}) \tag{5.1}$$

- $\overline{\mathbf{LL}}(\mathbf{EV_1}, \mathbf{EV_2})$: qui comptabilise le nombre de couples formés d'un couple d'objets (o_a, o_b) et d'un couple de variables (V_{1_i}, V_{2_j}) formé d'une variable V_{1_i} de EV_1 et d'une variable V_{2_j} de EV_2 tels que :
  le couple d'objets (o_a, o_b) est caractérisé simultanément par un **non-lien** selon V_{1_i} et par un **non-lien** selon V_{2_j}.

$$\overline{LL}(EV_1, EV_2) = \sum_{a=1..n} \sum_{b=a+1..n} \sum_{i=1..l} \sum_{j=1..m} (1 - lien_i(o_{a_i}, o_{b_i})) \times (1 - lien_j(o_{a_j}, o_{b_j})) \tag{5.2}$$

- $\mathbf{L\overline{L}}(\mathbf{EV_1}, \mathbf{EV_2})$: qui comptabilise le nombre de couples formés d'un couple d'objets (o_a, o_b) et d'un couple de variables (V_{1_i}, V_{2_j}) formé d'une variable V_{1_i} de EV_1 et d'une variable V_{2_j} de EV_2 tels que :
  le couple d'objets (o_a, o_b) est caractérisé simultanément par un **lien** selon V_{1_i} et par un **non-lien** selon V_{2_j}.

$$L\overline{L}(EV_1, EV_2) = \sum_{a=1..n} \sum_{b=a+1..n} \sum_{i=1..l} \sum_{j=1..m} lien_i(o_{a_i}, o_{b_i}) \times (1 - lien_j(o_{a_j}, o_{b_j})) \tag{5.3}$$

- $\overline{\mathbf{L}}\mathbf{L}(\mathbf{EV_1}, \mathbf{EV_2})$: qui comptabilise le nombre de couples formés d'un couple d'objets (o_a, o_b) et d'un couple de variables (V_{1_i}, V_{2_j}) formé d'une variable V_{1_i} de EV_1 et d'une variable V_{2_j} de EV_2 tels que :

le couple d'objets (o_a,o_b) est caractérisé simultanément par un **non-lien** selon V_{1_i} et par un **lien** selon V_{2_j}.

$$\overline{L}L(EV_1,EV_2) = \sum_{a=1..n} \sum_{b=a+1..n} \sum_{i=1..l} \sum_{j=1..m} (1-lien_i(o_{a_i},o_{b_i}))\times lien_j(o_{a_j},o_{b_j}) \tag{5.4}$$

REMARQUES :

- Par la suite nous nous contentons d'écrire (sauf indication contraire) LL, $\overline{L}\overline{L}$, $\overline{L}L$, $L\overline{L}$ en lieu et place respective de $LL(EV_1,EV_2)$, $\overline{L}\overline{L}(EV_1,EV_2)$, $\overline{L}L(EV_1,EV_2)$, $L\overline{L}(EV_1,EV_2)$.
- $LL + L\overline{L}$ correspond à m fois le nombre de liens au sein des variables de EV_1
- $\overline{L}L+\overline{L}\overline{L}$ correspond à m fois le nombre de non-liens au sein des variables de EV_1
- $LL + \overline{L}L$ correspond à l fois le nombre de liens au sein des variables de EV_2
- $L\overline{L} + \overline{L}\overline{L}$ correspond à l fois le nombre de liens au sein des variables de EV_2
- Une adéquation forte entre EV_1 et EV_2 se caractérise par de fortes valeurs pour LL et $\overline{L}\overline{L}$.
- L'ensemble de ces relations peuvent être résumées au sein d'une table de contingence :

	Liens dans EV_2	Non-Liens dans EV_2	$Total$
Liens dans EV_1	LL	$L\overline{L}$	$LL + L\overline{L}$
Non-Liens dans EV_1	$\overline{L}L$	$\overline{L}\overline{L}$	$\overline{L}L + \overline{L}\overline{L}$
$Total$	$LL + \overline{L}L$	$L\overline{L} + \overline{L}\overline{L}$	$\frac{n\times(n-1)}{2} \times l \times m$

5.3.2 Remarques Importantes Concernant l'Aspect Calculatoire

Bâtir cette table de contingence ne nécessite qu'une seule passe sur le jeu de données. Dans le cas de données catégorielles uniquement, cela ne requiert que $O(nlm)$ comparaisons[7], ce nombre de comparaisons peut atteindre $O(n^2lm)$ dans le cas de présence de variables quantitatives et d'utilisation de fonctions $lien_i$ telles que définies dans le cas 2 de l'exemple illustratif du chapitre précédent (page 69).

Du point de vue de l'utilisation mémoire, quel que soit la nature des données, le stockage de lm tables de contingence est nécessaire, ce qui correspond à un encombrement mémoire relativement faible et surtout totalement indépendant du nombre d'objets du jeu de données considéré.

7. Intuitivement, les définitions formelles de LL, $\overline{L}L$, $L\overline{L}$ et $\overline{L}\overline{L}$ semblent impliquer $O(n^2lm)$ comparaisons mais des astuces de calcul permettent de réduire ce nombre de comparaisons, ces astuces sont similaires à celles présentées précédemment) afin de bâtir $l \times m$ tables de contingence (croisant les l variables de EV_1 avec les m variables de EV_2)

5.3.3 Evaluation de l'adéquation entre $\mathbf{EV}$ un Ensemble de Variables et $\mathbf{EV}_\star$ un Sous Ensemble de $\mathbf{EV}$ ($\mathbf{EV}_\star \subseteq \mathbf{EV}$)

Afin d'évaluer l'adéquation entre EV un ensemble de variables et $EV_\star$ un sous ensemble de EV ($EV_\star \subseteq EV$) nous utilisons une adaptation des 4 indices que nous venons de présenter :

- $\widetilde{\mathbf{L}}\widetilde{\mathbf{L}}(\mathbf{EV}_\star,\mathbf{EV})$: qui comptabilise le nombre de couples formés d'un couple d'objets (o_a,o_b) et d'un couple de variables (V_{1_i},V_{2_j}) formé d'une variable V_{1_i} de $EV_\star$ et d'une variable V_{2_j} de EV telles que :
  $V_{1_i} \neq V_{2_j}$, le couple d'objets (o_a,o_b) est caractérisé simultanément par un **lien** selon V_{1_i} et par un **lien** selon V_{2_j}.

$$\widetilde{L}\widetilde{L}(EV_\star,EV) = \sum_{a=1..n} \sum_{b=a+1..n} \sum_{i=1..l} \sum_{\substack{j\ =\ 1..m \\ j \text{ tel que } V_{1_i} \neq V_{2_j}}} lien_i(o_{a_i},o_{b_i}) \times lien_j(o_{a_j},o_{b_j}) \tag{5.5}$$

- $\overline{\widetilde{\mathbf{L}}\widetilde{\mathbf{L}}}(\mathbf{EV}_\star,\mathbf{EV})$: qui comptabilise le nombre de couples formés d'un couple d'objets (o_a,o_b) et d'un couple de variables (V_{1_i},V_{2_j}) formé d'une variable V_{1_i} de $EV_\star$ et d'une variable V_{2_j} de EV telles que :
  $V_{1_i} \neq V_{2_j}$, le couple d'objets (o_a,o_b) est caractérisé simultanément par un **non-lien** selon V_{1_i} et par un **non-lien** selon V_{2_j}.

$$\overline{\widetilde{L}\widetilde{L}}(EV_\star,EV) = \sum_{a=1..n} \sum_{b=a+1..n} \sum_{i=1..l} \sum_{\substack{j\ =\ 1..m \\ j \text{ tel que } V_{1_i} \neq V_{2_j}}} (1-lien_i(o_{a_i},o_{b_i})) \times (1-lien_j(o_{a_j},o_{b_j})) \tag{5.6}$$

- $\widetilde{\mathbf{L}}\overline{\widetilde{\mathbf{L}}}(\mathbf{EV}_\star,\mathbf{EV})$: qui comptabilise le nombre de couples formés d'un couple d'objets (o_a,o_b) et d'un couple de variables (V_{1_i},V_{2_j}) formé d'une variable V_{1_i} de $EV_\star$ et d'une variable V_{2_j} de EV telles que :
  $V_{1_i} \neq V_{2_j}$, le couple d'objets (o_a,o_b) est caractérisé simultanément par un **lien** selon V_{1_i} et par un **non-lien** selon V_{2_j}.

$$\widetilde{L}\overline{\widetilde{L}}(EV_\star,EV) = \sum_{a=1..n} \sum_{b=a+1..n} \sum_{i=1..l} \sum_{\substack{j\ =\ 1..m \\ j \text{ tel que } V_{1_i} \neq V_{2_j}}} lien_i(o_{a_i},o_{b_i}) \times (1-lien_j(o_{a_j},o_{b_j})) \tag{5.7}$$

– $\overline{\widetilde{\mathbf{LL}}}(\mathbf{EV}_\star,\mathbf{EV})$ qui comptabilise le nombre de couples formés d'un couple d'objets (o_a,o_b) et d'un couple de variables (V_{1_i},V_{2_j}) formé d'une variable V_{1_i} de $EV_\star$ et d'une variable V_{2_j} de EV telles que :
$V_{1_i} \neq V_{2_j}$, le couple d'objets (o_a,o_b) est caractérisé simultanément par un **non-lien** selon V_{1_i} et par un **lien** selon V_{2_j}.

$$\overline{\widetilde{L}}\widetilde{L}(EV_\star,EV) = \sum_{a=1..n}\sum_{b=a+1..n}\sum_{i=1..l}\sum_{\substack{j\ =\ 1..m \\ j \text{ tel que } V_{1_i} \neq V_{2_j}}} (1-lien_i(o_{a_i},o_{b_i}))\times lien_j(o_{a_j},o_{b_j}) \tag{5.8}$$

REMARQUES :

– Par la suite nous nous contentons d'écrire (sauf indication contraire) $\widetilde{L}\widetilde{L}$, $\overline{\widetilde{L}}\overline{\widetilde{L}}$, $\overline{\widetilde{L}}\widetilde{L}$, $\widetilde{L}\overline{\widetilde{L}}$ en lieu et place respective de $\widetilde{L}\widetilde{L}(EV_1,EV_2)$, $\overline{\widetilde{L}}\overline{\widetilde{L}}(EV_1,EV_2)$, $\overline{\widetilde{L}}\widetilde{L}(EV_1,EV_2)$, $\widetilde{L}\overline{\widetilde{L}}(EV_1,EV_2)$.
– Une adéquation forte entre $EV_\star$ et EV se caractérise par de fortes valeurs pour LL et $\overline{\widetilde{L}}\overline{\widetilde{L}}$.
– L'ensemble de ces relations peuvent être résumées au sein d'une table de contingence :

	$\widetilde{L}$	$\overline{\widetilde{L}}$	$Total$
$\widetilde{L}$	$\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}$	$\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}$	$\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}$
$\overline{\widetilde{L}}$	$\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}$	$\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}$	$\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}$
$Total$	$\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}$	$\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}$	

Ainsi, le niveau d'adéquation entre $EV_\star$ et EV peut être caractérisé par les indices $\widetilde{L}\widetilde{L}$, $\overline{\widetilde{L}}\widetilde{L}$, $\widetilde{L}\overline{\widetilde{L}}$, $\overline{\widetilde{L}}\overline{\widetilde{L}}$. Une forte adéquation étant associée à de fortes valeurs pour $\widetilde{L}\widetilde{L}$, $\overline{\widetilde{L}}\overline{\widetilde{L}}$. Cependant, la signification de fortes valeurs n'étant pas totalement intuitive nous proposons, de manière similaire aux indices concernant l'évaluation de la validité de cns (partitions), de déterminer les lois statistiques suivies par les indices $\widetilde{L}\widetilde{L}$ et $\overline{\widetilde{L}}\overline{\widetilde{L}}$ en cas de non adéquation. Cela permet alors de dériver deux indices $Aq_1(EV_\star,EV)$ et $Aq_2(EV_\star,EV)$ caractérisant respectivement la significativité de $\widetilde{L}\widetilde{L}$ et $\overline{\widetilde{L}}\overline{\widetilde{L}}$ et suivant, dans les conditions de non adéquation, la loi normale centrée réduite :

$$Aq_1(EV_\star,EV) = \frac{\widetilde{\widetilde{L}}\widetilde{\widetilde{L}} - \frac{(\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}})(\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}})}{\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}}}{\sqrt{\frac{(\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}})(\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}})}{\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}} \times \left(1-\frac{\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}}{\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}}\right)}}, \; Aq_1(EV_\star,EV) \hookrightarrow N(0,1)$$

$$Aq_2(EV_\star,EV) = \frac{\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}} - \frac{(\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}})(\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}})}{\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}}}{\sqrt{\frac{(\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}})(\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}})}{\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}} \times \left(1-\frac{\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}}{\widetilde{\widetilde{L}}\widetilde{\widetilde{L}}+\overline{\widetilde{\widetilde{L}}}\widetilde{\widetilde{L}}+\widetilde{\widetilde{L}}\overline{\widetilde{\widetilde{L}}}+\overline{\widetilde{\widetilde{L}}}\overline{\widetilde{\widetilde{L}}}}\right)}}, \; Aq_2(EV_\star,EV) \hookrightarrow N(0,1)$$

5.3.4 Evaluation/Comparaison de l'Adéquation entre un Ensemble de Variables (EV) et des Sous Ensembles de EV

Afin d'évaluer/comparer l'adéquation entre un ensemble de variables (EV) et des sous ensembles de EV, nous utilisons la méthodologie présentée dans le cadre de l'évaluation/comparaison de la validité de cns pour laquelle on substitue toutefois les indices Aq_1 et Aq_2 aux indices xv_1 et xv_2.

Ainsi, la comparaison de l'adéquation de deux sous ensembles ($EV_1 \subseteq EV$ et $EV_2 \subseteq EV$) peut être réalisée par comparaison des couples de valeurs $(Aq_1(EV_\star,EV_1),Aq_2(EV_\star,EV_1))$ et $(Aq_1(EV_\star,EV_2),Aq_2(EV_\star,EV_2))$. Cette comparaison mène à 4 situations différentes :

- EV_1 est considéré comme plus en adéquation avec EV que EV_2 ssi $(Aq_1(EV_\star,EV_2) < Aq_1(EV_\star,EV_1)$ et $Aq_2(EV_\star,EV_2) < Aq_2(EV_\star,EV_1))$ ou $(Aq_1(EV_\star,EV_2) \leq Aq_1(EV_\star,EV_1)$ et $Aq_2(EV_\star,EV_2) < Aq_2(EV_\star,EV_1))$ ou $(Aq_1(EV_\star,EV_2) < Aq_1(EV_\star,EV_1)$ et $Aq_2(EV_\star,EV_2) \leq Aq_2(EV_\star,EV_1))$
  nous notons cette relation : $EV_1 < b > EV_2$
- EV_2 est considéré comme plus en adéquation avec EV que EV_1 ssi $(Aq_1(EV_\star,EV_1) < Aq_1(EV_\star,EV_2)$ et $Aq_2(EV_\star,EV_1) < Aq_2(EV_\star,EV_2))$ ou $(Aq_1(EV_\star,EV_1) \leq Aq_1(EV_\star,EV_2)$ et $Aq_2(EV_\star,EV_1) < Aq_2(EV_\star,EV_2))$ ou $(Aq_1(EV_\star,EV_1) < Aq_1(EV_\star,EV_2)$ et $Aq_2(EV_\star,EV_1) \leq Aq_2(EV_\star,EV_2))$
  nous notons cette relation : $EV_2 < b > EV_1$
- EV_1 et EV_2 sont considérés comme équivalents du point de vue de l'adéquation avec EV ssi $(Aq_1(EV_\star,EV_1) = Aq_1(EV_\star,EV_2)$ et $Aq_2(EV_\star,EV_1) = Aq_2(EV_\star,EV_2))$
  nous notons cette relation : $EV_2 < s > EV_1$
- EV_1 et EV_2 sont considérés comme incomparables du point de vue de l'adéquation avec EV ssi $(Aq_1(EV_\star,EV_1) < Aq_1(EV_\star,EV_2)$ et $Aq_2(EV_\star,EV_1) > Aq_2(EV_\star,EV_2))$ ou $(Aq_1(EV_\star,EV_1) > Aq_1(EV_\star,EV_2)$ et $Aq_2(EV_\star,EV_1) < Aq_2(EV_\star,EV_2))$
  nous notons cette relation : $EV_2 <? > EV_1$

5.3.5 La Nouvelle Méthode de Sélection de Variables

La méthode est ici en tout point identique à celle proposée pour l'apprentissage supervisé à l'unique exception du calcul des indices de validité des sous espaces de l'ERD que nous venons de présenter.

Concernant la définition de la fonction de fitness de l'AG, nous proposons une fonction basée sur l'observation que $Aq_1(EV_\star,EV)$ et $Aq_2(EV_\star,EV)$ doivent être les plus élevées possibles pour que $EV_\star$ et EV soient considérés comme étant en adéquation. Cette fonctions $fit(EV,EV_\star)$ est la suivante :

$$fit(EV,EV_\star) = \begin{cases} \sqrt{(a\tilde{q}_1 - Aq_1(EV_\star,EV))^2 + (a\tilde{q}_2 - Aq_2(EV_\star,EV))^2}, \\ \text{si } Aq_1(EV_\star,EV) > 0 \text{ et } Aq_2(EV_\star,EV) > 0 \\ 0 \text{ sinon} \end{cases}$$

qui correspond en quelque sorte à une distance du point de vue de la validité entre un ensemble virtuel particulier de variables (dont les valeurs Aq_1

Algorithme 6 Sélection de Variables pour l'Apprentissage Non Supervisé

1. Données : l'ERD EV

2. En une unique passe sur les données bâtir les $\frac{p(p-1)}{2}$ tables de contingence nécessaires aux calculs des mesures d'adéquation entre ensembles de variables présentées préalablement.

3. Fixer les paramètres de l'AG : *nombre de générations, taille de la population, Probabilité de Croisement, Probabilité de mutation*

4. Lancer l'AG utilisant la fonction de fitness spécifique définie ci-dessous.

5. Sélectionner le meilleur sous espace déterminé par l'AG

et Aq_2 seraient respectivement $a\tilde{q}_1$ et $a\tilde{q}_2$) et l'espace $EV_\star$. En fait, nous fixons $a\tilde{q}_1 = a\tilde{q}_2 = très\ forte\ valeur$ de manière à conférer à l'espace virtuel particulier l'aspect d'une sorte d'espace idéal du point de vue de l'adéquation avec EV. Ainsi, cette fonction de fitness correspond en somme à une distance du point de vue de la validité entre un espace virtuel idéal du point de vue de l'adéquation avec EV. Cette fonction de fitness doit donc être minimisée.

5.3.6 Evaluations Expérimentales

Afin d'évaluer la qualité et l'intérêt de la méthode que nous proposons nous présentons ici deux types d'expérimentations : l'une sur des jeux de données synthétiques, l'autre sur des jeux de données provenant de la collection de l'UCI.

5.3.6.1 Expérience #1 : Evaluation expérimentale sur jeux de données synthétiques

Description L'objectif est ici de tester dans quelle mesure notre méthode détecte les variables pertinentes (vecteur d'une véritable source d'informations), pour cela nous avons bâti le jeu de données synthétique suivant :
Ce jeu de données comprend 1000 objets caractérisés par 9 variables véritablement porteuses d'information V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, et par un ensemble d'autres variables correspondant à du bruit.

En définitive, les 250 premiers objets possèdent tous la même valeur D pour les variables V_1, V_2, V_3 ; quant aux variables restantes une valeur parmi A, B et C leur est assignée de manière aléatoire (la probabilité d'assignation de chaque valeur est $\frac{1}{3}$).

Pour les 250 objets suivants, ils possèdent tous la même valeur D pour les variables V_3, V_4, V_5 ; quant aux variables restantes une valeur parmi A, B et C leur est assignée de manière aléatoire (la probabilité d'assignation de chaque valeur est $\frac{1}{3}$).

Les 250 objets suivants possèdent tous la même valeur D pour les variables V_5, V_6, V_7 ; quant aux variables restantes une valeur parmi A, B et C leur est

assignée de manière aléatoire (la probabilité d'assignation de chaque valeur est $\frac{1}{3}$).

Finalement, les 250 objets restants possèdent tous la même valeur D pour les variables V_7, V_8, V_9 ; quant aux variables restantes une valeur parmi A, B et C leur est assignée de manière aléatoire (la probabilité d'assignation de chaque valeur est $\frac{1}{3}$).

Nous illustrons dans la figure 5.12, la composition du jeu de données. On peut ainsi se rendre compte que seules les 9 premières variables sont sources d'informations et que la structure des données est donc une partition des objets en 4 classes.

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	...	V_i	...	V_p
o_1	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_a	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_{250}	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_{251}	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_b	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_{500}	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_{501}	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_c	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_{750}	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_{751}	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_d	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C
o_{1000}	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C	D	D	D	A, B, C	A, B, C	A, B, C	A, B, C	A, B, C

FIG. 5.12 –: *Jeu de données synthétiques*

Les expérimentations menées sont les suivantes : nous avons exécuté plusieurs processus de SdV pour 6 jeux de données composés des 1000 objets caractérisés par les variables V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9 ainsi que par respectivement :

- 9 variables "bruit" pour le premier jeu de données (soit un jeu de données composé de 18 variables dont 50% sont sources d'informations)
- 18 variables "bruit" pour le deuxième jeu de données (soit un jeu de données composé de 27 variables dont $\frac{1}{3}$ sont sources d'informations)
- 27 variables "bruit" pour le troisième jeu de données (soit un jeu de données composé de 36 variables dont 25% sont sources d'informations)
- 36 variables "bruit" pour le quatrième jeu de données (soit un jeu de données composé de 45 variables dont 20% sont sources d'informations)
- 81 variables "bruit" pour le cinquième jeu de données (soit un jeu de données composé de 90 variables dont 10% sont sources d'informations)

– 171 variables "bruit" pour le sixième jeu de données (soit un jeu de données composé de 180 variables dont 5% sont sources d'informations).

Pour chacun des 6 jeux de données, nous avons ensuite lancé 5 séries de 5 processus de SdV :

– la première série étant caractérisée par un nombre de générations valant *50* pour l'AG utilisé ;
– la deuxième série étant caractérisée par un nombre de générations valant *100* pour l'AG utilisé ;
– la troisième série étant caractérisée par un nombre de générations valant *500* pour l'AG utilisé ;
– la quatrième série étant caractérisée par un nombres de générations valant *1000* pour l'AG utilisé ;
– la cinquième série étant caractérisée par un nombres de générations valant *2500* pour l'AG utilisé.

Les autres paramètres de l'AG ont été fixés à : *nombre de chromosomes par génération = 30* ; *probabilité de croisement = 0,98* ; *probabilité de mutation = 0,4* ; *élitisme = oui*.

Analyse des Résultats Les résultats sont présentés dans la figure 5.13 (voir page 153), ils nécessitent toutefois des explications... Notons tout d'abord que chacun des $6 \times 5 \times 5 = 150$[8] processus de SdV réalisés a mené à l'obtention d'un sous-espace de variables comprenant les 9 variables porteuses d'informations (V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9). Ainsi, les différentes courbes décrivent combien de variables "bruit" ont été simultanément sélectionnées avec les 9 variables pertinentes pour chaque série de 5 processus de SdV, elles détaillent pour chaque série :

– la moyenne du pourcentage de variables "bruit" sélectionnées par les 5 processus de SdV de la série;
– le pourcentage le plus faible de variables "bruit" sélectionnées (i.e. le pourcentage de variables "bruit" sélectionnées par le processus de SdV que l'on peut qualifier de "meilleur") ;
– le pourcentage le plus fort de variables "bruit" sélectionnées (i.e. le pourcentage de variables "bruit" sélectionnées par le processus de SdV que l'on peut qualifier de "moins bon").

Le premier point intéressant réside dans la capacité de la méthode à ne pas omettre de variables pertinentes dans la sélection qu'elle effectue, et ce, même lorsque la portion des variables pertinentes est très faible (5%) et que, simultanément, le nombre de générations de l'AG est très faible (50) (pour des nombres si faibles de générations on peut réellement considérer que le processus d'optimisation associé à l'utilisation de l'AG n'est pas arrivé à terme).

8. $=$ *nombre de jeux de données différents*$\times$*nombre de séries de processus de SdV différentes* $\times$ *nombre de processus de SdV par séries de processus de SdV*

Notons que le temps de calcul associé à ces traitements n'a été au maximum que d'une quinzaine de minutes pour les processus les plus longs (i.e. ceux impliquant le plus de variables et le plus grand nombre de générations) (temps de calcul obtenu pour un logiciel développé en Pascal Objet sous Delphi et exécuté sur un PC 128 Mo Ram, 600 Mhz). De plus, la complexité algorithmique de la méthode est indépendante du nombre d'individus une fois la passe sur le jeu de données réalisée et les tables de contingence dérivées.

Concernant le pourcentage de variables non pertinentes (donc "indésirables") introduites dans les sous ensembles de variables sélectionnées, on observe :

- qu'il est nul (resp. quasi nul) pour les jeux de données composés d'au moins 25% (resp. 20%) de variables pertinentes; et ce même pour des nombres de générations très faibles (50) ;
- que, concernant les jeux de données comportant 10% ou moins de 10% de variables pertinentes, la sélection de l'ensemble optimal de variables ($EV_\star = \{V_1,V_2,V_3,V_4,V_5,V_6,V_7,V_8,V_9\}$) est obtenue pour des nombres de générations supérieurs ou égaux à 1000.

La méthode apparaît donc comme excellente ici, car, les indices ainsi que la fonction de fitness utilisés rendent réellement compte de ce qu'est un bon sous ensemble de variables, et de plus, le processus d'optimisation utilisé permet la découverte du sous ensemble optimal tout en n'impliquant pas un temps de calcul démesuré (une quinzaine de minutes pour le cas le moins favorable). A titre indicatif, pour le jeu de données comportant 180 variables, le nombre de sous ensembles non vides de l'ERD est $2^{180} - 1 = 1{,}53 \times 10^{54}$, le nombre maximal de sous ensembles testés (dans le cas de 2500 générations et en admettant qu'un sous espace n'est évalué qu'une seule fois par l'AG) est $2500 \times 30 = 75000$, la comparaison entre ces deux valeurs montre bien l'efficacité du processus de recherche...

Ainsi, sur ces exemples synthétiques (certes relativement simplistes) la méthode que nous proposons semble d'une efficacité redoutable. Notons enfin que l'application d'algorithmes de cns sur le jeu de données "réduit" mènerait bien à la découverte de la structure en 4 classes et que le temps de calcul associé serait réduit d'un facteur allant de 2 à 20 (resp. 4 à 400) dans le cas d'algorithme possédant une complexité linéaire (resp. quadratique) selon le nombre de variables.

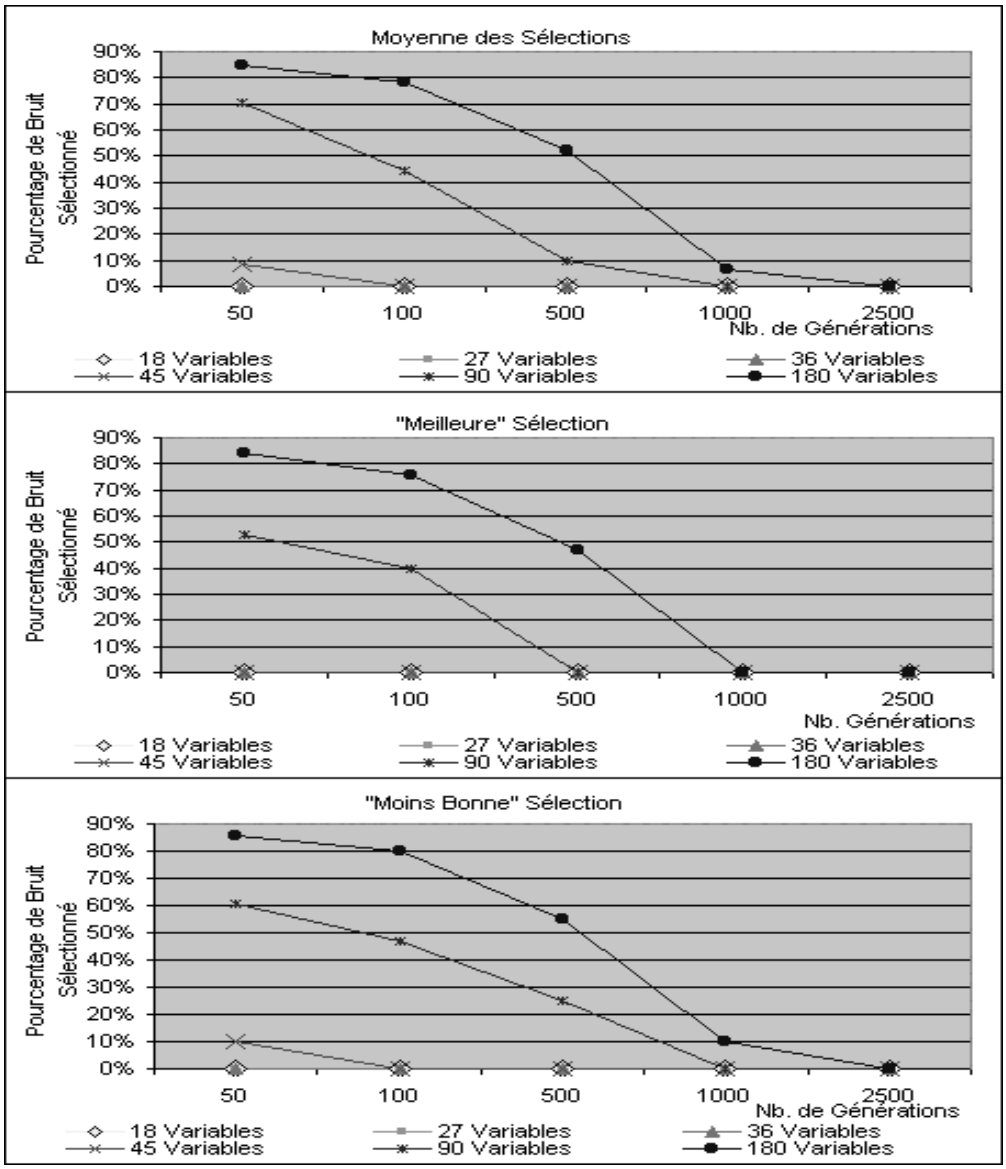


FIG. 5.13 –: *Résultats des expériences sur jeux de données synthétiques pour l'évaluation de la méthode de SdV en apprentissage non supervisé*

5.3.6.2 Expérience #2 : Evaluation Expérimentale sur Jeux de Données de l'UCI

Description Nous avons mené ici les mêmes expériences qu'au chapitre précédent : les jeux de données Small Soybean Diseases ainsi que Mushrooms sont utilisés pour réaliser diverses cns puis tester la validité de ces cns en considérant les jeux de données (les ERDs) dans leur intégralité.

De plus, nous avons évidemment mené les mêmes expérimentations en ne considérant pour chaque jeu de données que les variables sélectionnées par notre méthodologie de SdV :

- Pour le jeu de données : Small Soybean Diseases : seules 9 variables (plant-stand, precip., temp, area-damaged, stem-cankers, canker-lesion, int-discolor, sclerotia, fruit-pods) ont été sélectionnées parmi les 35 variables du jeu de données ;
- pour le jeu de données Mushrooms seules 15 variables (bruises?, odor, gill-color, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil-type, spore-print-color, population, habitat) ont été sélectionnées parmi les 22 variables.

La question est alors de savoir si la validité des cns obtenues par application des algorithmes de cns (KEROUAC et K-Modes) sur le jeu de données "réduit" (ERD "réduit") est aussi bonne ou meilleure que celle des cns obtenues par application des algorithmes de cns (KEROUAC et K-Modes) sur le jeu de données "complet" (ERD "complet").

Nous avons donc utilisé les méthodes de cns pour données catégorielles KEROUAC et K-Modes avec des paramètres différents (des valeurs différentes pour le facteur de granularité pour KEROUAC et des nombres de classes différents pour les K-Modes) de manière à générer des cns possédant des nombres de classes différents (les paramètres ont été fixés de manière à obtenir pour le jeu de données Small Soybean des cns en 2, 3, 4, 5, 6, 7, 8, 9, 10 classes et des cns en 2, 3, 4, 5, 6, 7, 8, 9 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 classes pour le jeu de données Mushrooms). Ces expériences ont donc été menées à la fois sur le jeu de données "complet" et sur le jeu de données "réduit".

REMARQUE : Pour la méthode des K-Modes nous avons réalisé pour chaque nombre de classes 10 expériences différentes et conservé la cns correspondant à la meilleure valeur pour le critère QKM (critère sous-jacent à cette méthode).

Les résultats sont exposés (à titre indicatif), pour le jeu de données Mushrooms, dans les tableaux des figures 5.16 et 5.17 (page 159 et 160). Ces tableaux

donnent les informations suivantes pour chaque partition (cns):

- le nombre de classes (#Cl.),
- la valeur à laquelle a été fixée le facteur de granularité (α) (si la cns a été obtenue grâce à KEROUAC),
- les valeurs de xv_1 et xv_2 calculées dans l'ERD "complet" (pour l'ensemble des cns qu'elles aient été obtenues par application d'un algorithme de cns sur l'intégralité du jeu de données (ERD "complet") ou non (ERD "réduit")),
- les valeurs de xv_1 et xv_2 calculées dans l'ERD "réduit" (si la cns a été obtenue par application d'un algorithme de cns sur le jeu de données "réduit" (ERD "réduit")),
- la valeur du critère à minimiser sous-jacent à la méthode K-Modes (QKM) calculée dans l'ERD "complet", mais également la valeur calculée dans l'ERD "réduit" si la cns a été obtenue par application d'un algorithme de cns sur le jeu de données "réduit" (ERD "réduit"),
- la valeur du critère à minimiser sous-jacent à la méthode KEROUAC (NCC) calculée dans l'ERD "complet", mais également la valeur calculée dans l'ERD "réduit" si la cns a été obtenue par application d'un algorithme de cns sur le jeu de données "réduit" (ERD "réduit"),
- le taux de correction (T.C.) de chaque partition pour le concept "pathologie".

Analyse des Résultats

Jeu de Données Mushrooms Considérons tout d'abord les résultats associés aux expérimentations sur le jeu de données Mushrooms (voir tableaux des figures 5.16 et 5.17 ainsi que les figures 5.14 et 5.15).

Concernant l'évaluation de la validité des cns par l'intermédiaire de la **méthodologie présentée au chapitre 4** (évaluation réalisée en considérant l'ERD "complet"), on observe sur la figure 5.14 que, du point de vue de la validité, les cns réalisées sur le jeu de données considéré dans son intégralité (ERD "complet") ou partiellement (ERD "réduit") sont très proches voire équivalentes que la méthode employée soit KEROUAC ou les K-Modes. On peut même observer que la validité des cns obtenues sur l'ERD "réduit" semble de manière générale meilleure. Concernant la cns la plus valide, la cns la plus valide obtenue dans l'ERD "complet" est celle comprenant 19 classes obtenue par KEROUAC pour le jeu de donnée "complet" et elle est du point de vue de la validité quasi-équivalente à celle comprenant 17 classes obtenue par KEROUAC sur le jeu de données réduit. Pour les cns obtenues par l'intermédiaire des K-Modes, l'utilisation de l'ERD "réduit" mène à des cns équivalentes ou meilleures du point de vue de la validité.

On peut également observer que l'évaluation de la validité des cns obtenues dans l'ERD "réduit" donne un profil de courbes très similaire à celui observé dans l'ERD "complet" (voir courbes Kerouac, Kerouac + SdV, K-Modes (Meilleur), K-Modes + SdV (Meilleur) et les courbes Kerouac + SdV (dans l'ERD "réduit"), K-Modes + SdV (Meilleur) (dans l'ERD "réduit") de la figure 5.14). De plus, les valeurs des indices xv_1 et xv_2 sont proches. Ce dernier point est très intéressant car il semble montrer que l'évaluation de la validité réalisée dans l'ERD "réduit" est conforme à celle réalisée dans l'ERD "complet" ce qui implique que l'on peut se contenter de procéder à l'évaluation de la validité dans l'ERD "réduit" (et ainsi limiter le coût calculatoire nécessaire à la validation).

D'après ce mode d'évaluation de la validité de cns, notre méthodologie de sélection de variables fournit de très bons ensembles de variables puisque la validité des cns bâties sur l'ERD "réduit" est quasi-équivalente à celle des cns obtenues à partir de l'ERD "complet"...

Si on considère maintenant les valeurs des **critères** QKM **et** NCC ainsi que celle du **taux de correction** (T.C.) pour chaque cns, on observe là encore (voir figure 5.15) la presque parfaite adéquation des résultats obtenus lors de l'utilisation de l'ERD "complet" et lors de l'utilisation de l'ERD "réduit". Ces résultats étayent, eux aussi, la conclusion que les cns obtenues par utilisation de l'ERD "réduit" présentent un niveau de validité équivalent à celles obtenues en accédant à l'ERD complet.

Les résultats obtenus sur ce jeu de données sont frappants et semblent démontrer la très grande efficacité de notre méthodologie de sélection de variables pour réduire la dimension de l'ERD tout en conservant un niveau de validité des cns d'excellente qualité. Notons de plus, que la réduction de l'ERD permet à la fois la réduction des coûts calculatoire et de stockage associés au processus de cns et la réduction de ces mêmes coûts pour le processus d'évaluation de la validité des cns.

Jeu de Données Small Soybean Disease Nous ne détaillons pas l'analyse des résultats obtenus sur le jeu K-Modes (voir figure 5.18) qui tendent à apporter les mêmes conclusions. De plus, cette fois-ci, la réduction de la dimension de l'ERD est plus importante : seulement 25,7% des variables sont conservées (68,2% des variables étaient conservées pour le jeu de données Mushrooms).

Notons cependant l'intégration d'un élément supplémentaire pour l'évaluation de la validité des cns obtenues par application des algorithmes de cns sur l'ERD "réduit" : un des graphiques de la figure 5.18 permet l'évaluation de l'adéquation entre les couples de cns possédant le même nombre de classes[9].

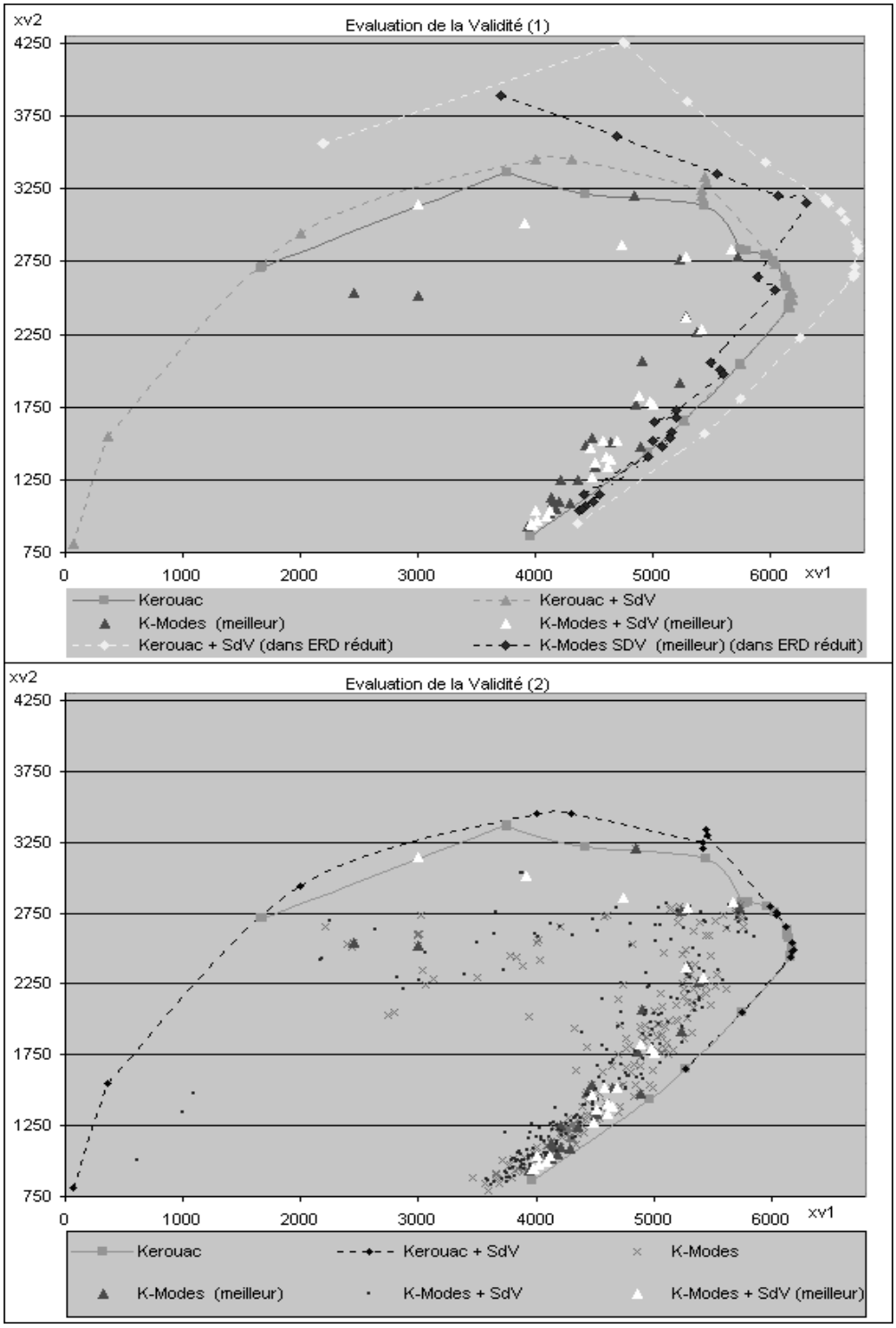


FIG. 5.14 –: *Résultats des expériences sur le jeu de données Mushrooms pour l'évaluation de la méthode de SdV en apprentissage non supervisé*

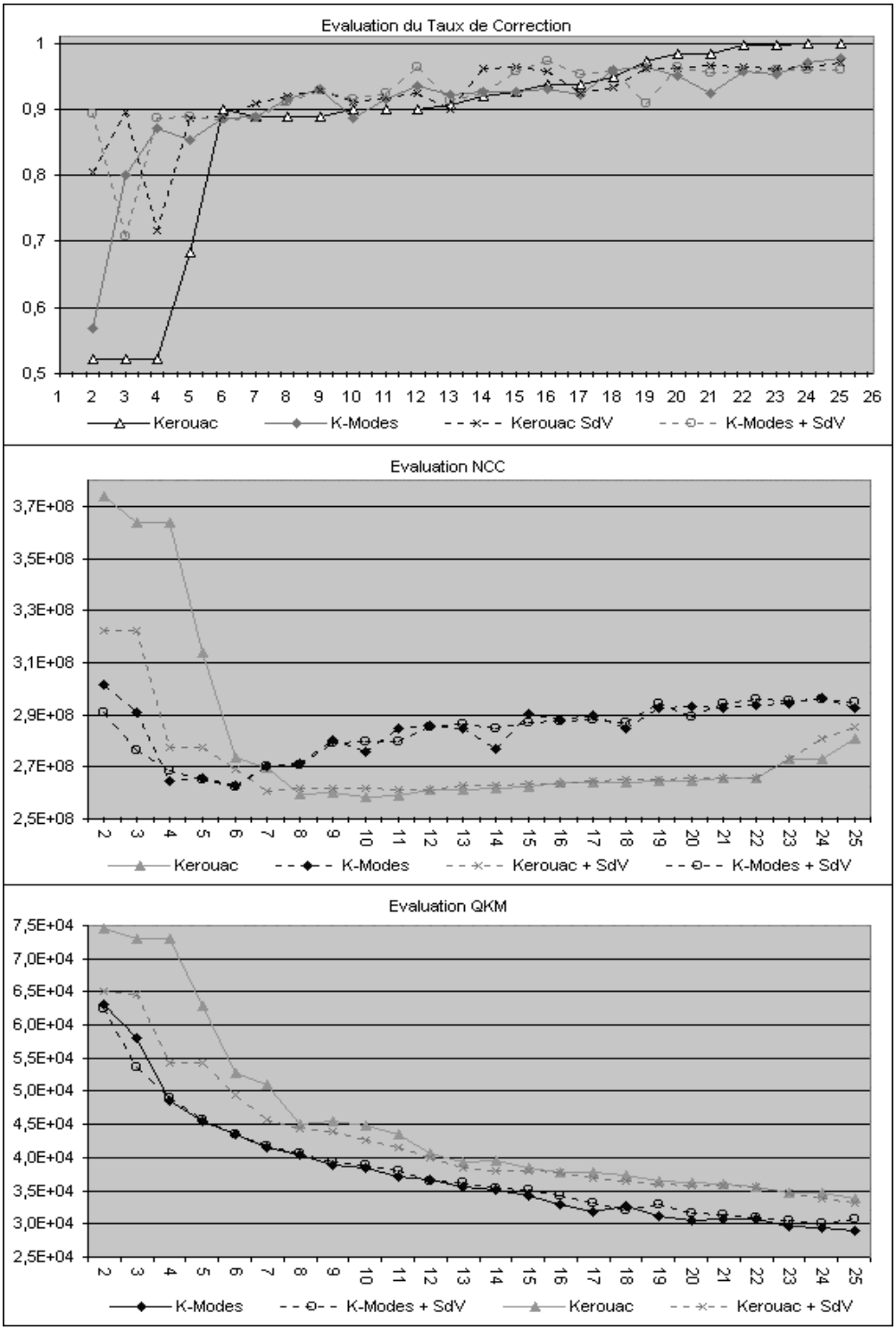


FIG. 5.15 –: *Résultats des expériences sur le jeu de données Mushrooms pour l'évaluation de la méthode de SdV en apprentissage non supervisé*

Algo.	SdV	#Cl.	α	*xv1* (ERD complet)	*xv2* (ERD complet)	*xv1* (ERD réduit)	*xv2* (ERD réduit)	*QKM* (ERD complet)	*QKM* (ERD réduit)	*NCC* (ERD complet)	*NCC* (ERD réduit)	T.C.
Kerouac	Non	2	0,5	65,28	814,63	x	x	74555	x	3,74E+08	x	52,1%
Kerouac	Non	3	0,55	364,19	1548,93	x	x	73003	x	3,64E+08	x	52,1%
Kerouac	Non	4	0,6	364,21	1548,96	x	x	72909	x	3,64E+08	x	52,1%
Kerouac	Non	5	0,65	1998,05	2942,31	x	x	62858	x	3,14E+08	x	68,3%
Kerouac	Non	6	0,7	4005,27	3455,12	x	x	52758	x	2,74E+08	x	90,0%
Kerouac	Non	7	0,8	4302,08	3451,16	x	x	51066	x	2,69E+08	x	88,8%
Kerouac	Non	8	0,85	5416,25	3247,15	x	x	45102	x	2,60E+08	x	88,8%
Kerouac	Non	9	0,865	5414,86	3206,88	x	x	45466	x	2,60E+08	x	88,8%
Kerouac	Non	10	1	5440,97	3334,72	x	x	44790	x	2,59E+08	x	89,9%
Kerouac	Non	11	1,075	5455,89	3294,50	x	x	43414	x	2,59E+08	x	89,9%
Kerouac	Non	12	1,1	5990,08	2799,89	x	x	40746	x	2,61E+08	x	89,9%
Kerouac	Non	13	1,2	6038,25	2750,24	x	x	39308	x	2,61E+08	x	90,5%
Kerouac	Non	14	1,225	6041,13	2731,23	x	x	39536	x	2,62E+08	x	91,9%
Kerouac	Non	15	1,25	6115,69	2655,86	x	x	38456	x	2,62E+08	x	92,5%
Kerouac	Non	16	1,475	6179,03	2544,00	x	x	37748	x	2,64E+08	x	93,7%
Kerouac	Non	17	1,482	6179,07	2543,76	x	x	37692	x	2,64E+08	x	93,7%
Kerouac	Non	18	1,5	6180,72	2535,83	x	x	37260	x	2,64E+08	x	94,9%
Kerouac	Non	19	1,825	6181,50	2486,96	x	x	36540	x	2,65E+08	x	97,2%
Kerouac	Non	20	1,85	6178,93	2480,13	x	x	36300	x	2,65E+08	x	98,4%
Kerouac	Non	21	2	6163,76	2443,12	x	x	35916	x	2,65E+08	x	98,4%
Kerouac	Non	22	2,5	6158,23	2433,17	x	x	35628	x	2,66E+08	x	99,6%
Kerouac	Non	23	2,85	5748,99	2045,88	x	x	34764	x	2,73E+08	x	99,6%
Kerouac	Non	24	3	5747,60	2044,67	x	x	34700	x	2,73E+08	x	100,0%
Kerouac	Non	25	3,05	5264,11	1652,58	x	x	33836	x	2,81E+08	x	100,0%
Kerouac	Oui	2	0,4	1665,16	2709,24	2189,10	3561,69	65034	53328	3,23E+08	2,39E+08	68,2%
Kerouac	Oui	3	0,408	1674,90	2715,43	2196,79	3561,54	64680	53010	3,22E+08	2,38E+08	68,2%
Kerouac	Oui	4	0,5	3753,67	3365,00	4746,65	4255,17	54350	42766	2,78E+08	1,72E+08	88,8%
Kerouac	Oui	5	0,55	3758,18	3359,97	4756,24	4252,28	54192	42608	2,77E+08	1,72E+08	89,1%
Kerouac	Oui	6	0,6	4423,19	3218,01	5297,42	3854,04	49350	38826	2,69E+08	1,60E+08	89,4%
Kerouac	Oui	7	0,7	5441,62	3129,19	5963,08	3429,05	45782	35834	2,60E+08	1,49E+08	89,4%
Kerouac	Oui	8	0,725	5752,20	2829,95	6464,36	3180,32	44294	34054	2,62E+08	1,45E+08	89,2%
Kerouac	Oui	9	0,8	5781,35	2821,14	6485,33	3164,67	43934	33742	2,62E+08	1,44E+08	89,9%
Kerouac	Oui	10	0,85	5805,33	2822,65	6493,72	3157,36	42574	32862	2,62E+08	1,44E+08	89,9%
Kerouac	Oui	11	0,9	5961,70	2792,17	6595,79	3089,15	41480	32290	2,61E+08	1,44E+08	91,6%
Kerouac	Oui	12	0,915	6026,27	2758,08	6635,10	3036,73	40040	31138	2,61E+08	1,43E+08	91,6%
Kerouac	Oui	13	1	6131,39	2629,16	6733,76	2887,45	38400	29794	2,63E+08	1,43E+08	93,6%
Kerouac	Oui	14	1,01	6134,28	2626,20	6735,96	2883,79	38096	29538	2,63E+08	1,43E+08	94,2%
Kerouac	Oui	15	1,05	6129,08	2591,43	6746,70	2852,57	37984	29382	2,63E+08	1,43E+08	97,4%
Kerouac	Oui	16	1,3	6144,52	2575,06	6746,34	2827,27	37872	29242	2,63E+08	1,44E+08	97,8%
Kerouac	Oui	17	1,4	6173,85	2496,00	6725,89	2719,18	36976	28494	2,65E+08	1,44E+08	96,9%
Kerouac	Oui	18	1,5	6166,37	2450,71	6713,71	2668,24	36356	27942	2,65E+08	1,44E+08	96,8%
Kerouac	Oui	19	1,75	6163,88	2443,91	6710,33	2660,57	36116	27702	2,65E+08	1,45E+08	98,0%
Kerouac	Oui	20	1,825	6158,34	2433,96	6702,07	2648,86	35828	27510	2,66E+08	1,45E+08	99,2%
Kerouac	Oui	21	2	6158,16	2433,40	6701,53	2648,12	35684	27398	2,66E+08	1,45E+08	99,6%
Kerouac	Oui	22	2,1	6158,23	2433,17	6701,17	2647,69	35628	27358	2,66E+08	1,45E+08	99,6%
Kerouac	Oui	23	2,5	5748,99	2045,88	6250,31	2224,29	34764	26494	2,73E+08	1,49E+08	99,6%
Kerouac	Oui	24	2,4	5265,61	1653,76	5749,70	1805,80	33900	25630	2,81E+08	1,54E+08	99,6%
Kerouac	Oui	25	2,75	4964,02	1433,71	5437,69	1570,52	33236	24950	2,85E+08	1,57E+08	99,9%

FIG. 5.16 –: *Résultats des expériences sur le jeu de données Mushrooms pour l'évaluation de la méthode de SdV en apprentissage non supervisé*

Algo.	SdV	#Cl.	α	$xv1$ (ERD complet)	$xv2$ (ERD complet)	$xv1$ (ERD réduit)	$xv2$ (ERD réduit)	QKM (ERD complet)	QKM (ERD réduit)	NCC (ERD complet)	NCC (ERD réduit)	T.C.
K-Modes	Non	2	x	2446,97	2536,24	x	x	63002	x	3,01E+08	x	56,8%
K-Modes	Non	3	x	2997,53	2518,52	x	x	57950	x	2,91E+08	x	80,1%
K-Modes	Non	4	x	4843,13	3203,64	x	x	48639	x	2,65E+08	x	87,1%
K-Modes	Non	5	x	5224,51	2761,32	x	x	45558	x	2,66E+08	x	85,3%
K-Modes	Non	6	x	5718,20	2783,41	x	x	43395	x	2,63E+08	x	88,7%
K-Modes	Non	7	x	5279,42	2372,88	x	x	41509	x	2,70E+08	x	88,8%
K-Modes	Non	8	x	5373,55	2262,90	x	x	40382	x	2,71E+08	x	91,5%
K-Modes	Non	9	x	4859,74	1771,83	x	x	38830	x	2,80E+08	x	92,9%
K-Modes	Non	10	x	4902,09	2068,90	x	x	38479	x	2,76E+08	x	88,7%
K-Modes	Non	11	x	4475,37	1542,21	x	x	37016	x	2,85E+08	x	91,6%
K-Modes	Non	12	x	4425,79	1486,61	x	x	36708	x	2,86E+08	x	93,4%
K-Modes	Non	13	x	4645,16	1512,24	x	x	35587	x	2,85E+08	x	92,2%
K-Modes	Non	14	x	5230,93	1914,67	x	x	35026	x	2,77E+08	x	92,7%
K-Modes	Non	15	x	4207,03	1249,32	x	x	34157	x	2,90E+08	x	92,5%
K-Modes	Non	16	x	4508,57	1334,15	x	x	32918	x	2,88E+08	x	93,0%
K-Modes	Non	17	x	4356,21	1243,66	x	x	31816	x	2,90E+08	x	92,1%
K-Modes	Non	18	x	4888,91	1474,66	x	x	32615	x	2,85E+08	x	95,9%
K-Modes	Non	19	x	4202,00	1099,96	x	x	31235	x	2,93E+08	x	96,3%
K-Modes	Non	20	x	4141,25	1097,39	x	x	30545	x	2,93E+08	x	95,1%
K-Modes	Non	21	x	4127,52	1125,23	x	x	30717	x	2,93E+08	x	92,5%
K-Modes	Non	22	x	4173,15	1046,59	x	x	30620	x	2,94E+08	x	95,7%
K-Modes	Non	23	x	4123,91	1015,15	x	x	29524	x	2,94E+08	x	95,3%
K-Modes	Non	24	x	3933,08	932,79	x	x	29412	x	2,96E+08	x	97,0%
K-Modes	Non	25	x	4289,07	1092,48	x	x	28878	x	2,93E+08	x	97,6%
K-Modes	Oui	2	x	2995,12	3146,26	3708,20	3895,33	62524	50820	2,91E+08	1,93E+08	89,2%
K-Modes	Oui	3	x	3910,78	3012,38	4694,96	3616,42	53594	42440	2,76E+08	1,69E+08	70,9%
K-Modes	Oui	4	x	4737,88	2859,79	5548,91	3349,33	49055	37948	2,69E+08	1,54E+08	88,7%
K-Modes	Oui	5	x	5283,26	2786,86	6063,65	3198,50	45755	35518	2,65E+08	1,48E+08	88,9%
K-Modes	Oui	6	x	5662,85	2830,27	6301,53	3149,48	43528	33588	2,62E+08	1,46E+08	88,4%
K-Modes	Oui	7	x	5275,70	2366,85	5890,17	2642,53	41818	31883	2,70E+08	1,50E+08	88,6%
K-Modes	Oui	8	x	5410,69	2290,32	6045,29	2558,94	40725	30886	2,71E+08	1,49E+08	91,2%
K-Modes	Oui	9	x	4878,43	1827,13	5499,81	2059,85	39425	29420	2,79E+08	1,54E+08	92,8%
K-Modes	Oui	10	x	4979,68	1791,95	5567,07	2003,33	38844	29000	2,80E+08	1,54E+08	91,5%
K-Modes	Oui	11	x	5006,49	1765,97	5598,28	1974,72	37990	28176	2,80E+08	1,54E+08	92,3%
K-Modes	Oui	12	x	4575,60	1515,89	5205,80	1724,68	36494	26726	2,85E+08	1,57E+08	96,3%
K-Modes	Oui	13	x	4469,83	1467,21	5013,00	1645,51	36182	26619	2,86E+08	1,59E+08	91,2%
K-Modes	Oui	14	x	4687,35	1513,87	5194,15	1677,55	35411	26126	2,85E+08	1,58E+08	92,4%
K-Modes	Oui	15	x	4602,39	1407,95	5166,33	1580,47	35056	25624	2,87E+08	1,58E+08	95,6%
K-Modes	Oui	16	x	4509,15	1365,88	5005,07	1516,10	34145	25165	2,88E+08	1,59E+08	97,1%
K-Modes	Oui	17	x	4608,15	1337,92	5075,67	1473,66	33063	24279	2,88E+08	1,59E+08	95,3%
K-Modes	Oui	18	x	4641,12	1385,40	5152,63	1538,09	32072	23228	2,87E+08	1,58E+08	95,7%
K-Modes	Oui	19	x	4005,76	1037,15	4418,26	1143,95	32850	23820	2,94E+08	1,64E+08	90,8%
K-Modes	Oui	20	x	4481,29	1268,39	4966,05	1405,59	31577	22792	2,89E+08	1,60E+08	96,3%
K-Modes	Oui	21	x	4117,64	1041,47	4550,10	1150,85	31303	22476	2,94E+08	1,63E+08	95,5%
K-Modes	Oui	22	x	3972,32	942,44	4378,35	1038,78	30987	22247	2,96E+08	1,64E+08	95,6%
K-Modes	Oui	23	x	4013,55	970,78	4421,22	1069,38	30608	21884	2,95E+08	1,64E+08	95,8%
K-Modes	Oui	24	x	3964,29	945,64	4395,04	1048,40	30040	21174	2,96E+08	1,64E+08	95,9%
K-Modes	Oui	25	x	4091,70	999,62	4496,15	1098,43	30664	21970	2,95E+08	1,64E+08	96,0%

FIG. 5.17 –: *Résultats des expériences sur le jeu de données Mushrooms pour l'évaluation de la méthode de SdV en apprentissage non supervisé*

Nous évaluons en effet l'adéquation entre :

- les couples de cns composés de cns possédant le même nombre de classes et tels que l'une des cns corresponde à celle obtenue par la méthode KEROUAC sur l'ERD "complet", l'autre à celle obtenue par la méthode KEROUAC sur l'ERD "réduit" (courbe nommée KEROUAC) ;
- les couples de cns composés de cns possédant le même nombre de classes et tels que l'une des cns correspond à la meilleure cns[10] obtenue par la méthode K-Modes sur l'ERD "complet", l'autre à la meilleure cns par la méthode K-Modes sur l'ERD "réduit" (courbe nommée K-Modes + SdV);
- les couples de cns composés de cns possédant le même nombre de classes tels que l'une des cns correspond à la meilleure cns obtenue par la méthode K-Modes sur l'ERD "complet", l'autre à la cns obtenue par la méthode K-Modes sur l'ERD "réduit" et impliquant la meilleure adéquation (i.e. la plus faible valeur de l'indice)(courbe nommée min. K-Modes + SdV);
- les couples de cns composés de cns possédant le même nombre de classes tesl que l'une des cns correspond à la meilleure cns obtenue par la méthode K-Modes sur l'ERD "complet", l'autre à la cns obtenue par la méthode K-Modes sur l'ERD "réduit" et impliquant la moins bonne adéquation (i.e. la plus forte valeur de l'indice)(courbe nommée max. K-Modes + SdV);
- les couples de cns composés de cns possédant le même nombre de classes tels que l'une des cns correspondant à la meilleure cns obtenue par la méthode K-Modes sur l'ERD "complet", l'autre à la cns obtenue par la méthode K-Modes sur l'ERD "complet" et impliquant la meilleure adéquation (i.e. la plus faible valeur de l'indice)(courbe nommée min. K-Modes);
- les couples de cns composés de cns possédant le même nombre de classes tels que l'une des cns correspond à la meilleure cns obtenue par la méthode K-Modes sur l'ERD "complet", l'autre à la cns obtenue par la méthode K-Modes sur l'ERD "complet" et impliquant la moins bonne adéquation (i.e. la plus forte valeur de l'indice)(courbe nommée max. K-Modes);

La dernière série de valeur sert de témoin : elle montre les valeurs maximales de l'indice d'adéquation pour un couple de cns (ayant le même nombre de classes) lors de l'application du même algorithme de cns (les K-Modes) sur l'intégralité du jeu de données. Cette série montre donc la variabilité maximale pour l'adéquation en considérant des processus de cns appliqués sur le même ERD : l'ERD "complet". Les valeurs obtenues pour l'ensemble des autres

9. l'indice utilisé ici pour l'évaluation de l'adéquation entre cns est l'indice Adq présenté au chapitre suivant, page 176. Plus sa valeur est proche de 0 plus l'adéquation est forte.

10. meilleure cns signifie ici qu'il s'agit de la cns ayant la plus faible valeur pour le critère QKM (critère sous-jacent à la méthode K-Modes)

courbes sont inférieures ou très proches. Cela prouve que les cns obtenues par application des algorithmes de cns sur l'ERD "réduit" présentent une excellente adéquation avec celles obtenues par application des algorithmes de cns sur l'ERD "complet". (Dans le cas de la méthode KEROUAC la valeur de l'indice d'adéquation vaut même 0 pour les couples de cns en 2, 3 et 4 classes ce qui signifie que ces couples de cns sont composés de cns identiques). Cette dernière expérience constitue une nouvelle indication de la qualité de notre méthodologie de SdV pour l'apprentissage non supervisé.

5.3.7 Conclusion

En résumé, nous proposons, une méthode de sélection de variables pour l'apprentissage non supervisé:

- ne nécessitant qu'une unique passe sur le jeu de données, et une complexité algorithmique faible (dans le cas de données catégorielles, la complexité est linéaire selon le nombre d'objets du jeu de données et quadratique selon le nombre de variables du jeu de données) ce qui lui confère une rapidité très intéressante ;
- possédant un coût de stockage faible
- utilisant une extension de la méthodologie préalablement introduite pour la comparaison de la validité de cns et un AG ;

Les évaluations expérimentales ont montré que :

- les cns obtenues sur l'ERD "réduit" sont d'excellente qualité ;
- l'ERD peut être parfois extrêmement "réduit" et la présence d'un fort bruit ne semble pas handicaper cette méthode ;
- concernant le temps de calcul, notre méthode est bonne.

Les remarques concernant les éventuelles améliorations de la méthode sont les mêmes que celles apportées pour la méthode de SdV pour l'apprentissage supervisé :

- notre méthode peut être améliorée du point de vue du coût calculatoire (une réduction du temps de calcul associé par substitution d'une méthode d'optimisation gloutonne à l'AG).
- on peut aisément modifier la structure de l'AG de manière à pouvoir rechercher non pas l'ensemble "optimal" de variables mais le meilleur ensemble de variables tel qu'il comprenne au plus un nombre fixé de variables.

Notons également que cette méthode de SdV pour l'apprentissage non supervisé permet une sélection des variables de l'ERD initial et non une sélection de variables correspondant à une transformation des variables de l'ERD initial comme le permet, par exemple, les approches basées sur l'analyse factorielle.

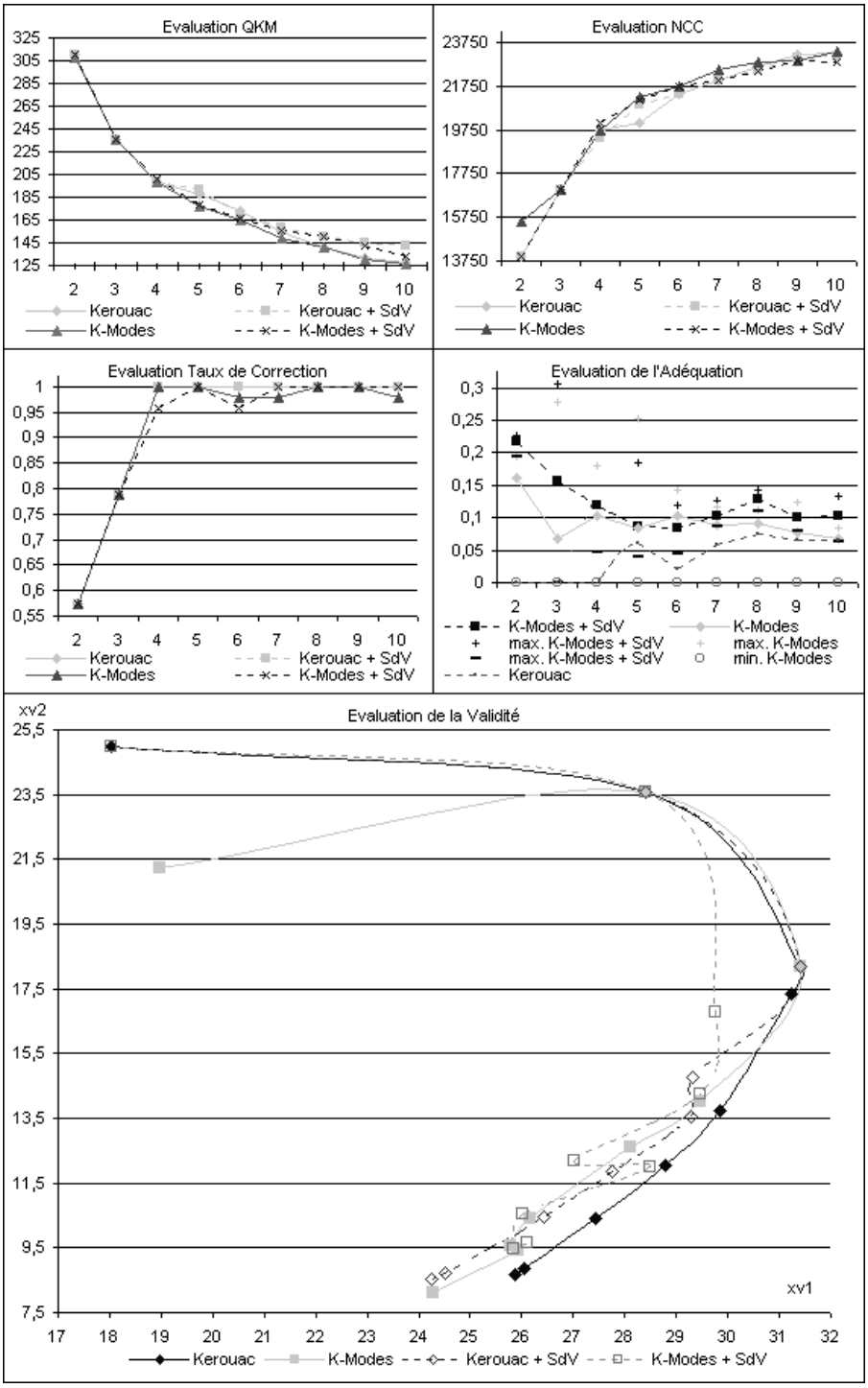


FIG. 5.18 –: *Résultats des expériences sur le jeu de données Small Soybean Disease pour l'évaluation de la méthode de sélection de variables en apprentissage non supervisé*

Ce point est particulièrement intéressant si on veut bâtir un modèle aisément interprétable.

REMARQUE : La méthode que nous venons de proposer peut également être employée si des variables quantitatives sont présentes, cependant, si elle ne nécessité alors qu'une unique passe sur les données, et implique un coût de stockage équivalent, sa complexité quadratique selon le nombre d'objets du jeu de données et linéaire selon le nombre de variables du jeu de données est handicapante du point de vue du coût calculatoire.

6 Agrégation de Classifications Non Supervisées : La Problématique "Cluster Ensembles"

"Toute partie tient à se réunir à son tout pour échapper ainsi à sa propre imperfection."

- Léonard De Vinci -
Les Carnets

6.1 Introduction

L'idée d'intégrer de multiples sources de données et/ou de modèles se retrouve dans diverses problématiques de l'ECD telles que la fusion de données (voir [Gra89] par exemple), ou l'apprentissage supervisé. Ainsi en apprentissage supervisé, des algorithmes d'agrégation de modèles basés sur des stratégies adaptatives (le boosting [FS96], [FS97]) ou aléatoires (le bagging [Bre96b], [Bre96a], les forêts aléatoires (random forests) [BFOS01]) permettent d'améliorer la qualité des modèles bâtis par agrégation d'un grand nombre de modèles tout en évitant le sur-ajustement (over-fitting). De nombreux articles comparatifs montrent leur efficacité sur des exemples de données synthétiques et surtout pour des problèmes réels complexes (voir par exemple [Gha00]) tandis que leurs propriétés théoriques sont un thème de recherche actif.

Bien que les premiers travaux concernant l'agrégation soient antérieurs à la Révolution Française (les travaux de Condorcet et Borda notamment), la dernière décade constitue la période de réelle émergence et d'intérêt prééminent pour cette problématique (une série de workshops lui a ainsi été spécifiquement dédiée [KR02]). Jusqu'à ces deux dernières années l'objectif avoué de ces méthodes d'agrégation était l'accroissement de la précision et de la robustesse de tâches de classification supervisée ou de régression. Les très récents travaux de Strehl et Gosh ([SG02a], [SG02b], [Str02], [GSS02]), Geurts [Geu03], ou les notres ([JN03d], [JN03e]), proposent d'élargir le champs des objectifs de ces méthodes à des notions comme la réduction du temps de calcul associé à des processus de classification (supervisée pour les travaux de Geurts, non supervisée pour les travaux de Strehl et Gosh ainsi que pour nos travaux) ou encore

à l'élargissement des types de données traitables par les méthodes de classification...

Contrairement à la classification supervisée ou à la régression, relativement peu d'approches ont été proposées pour la combinaison de multiples cns, les exceptions les plus notables incluent :

- des méthodes d'agrégation de type consensus strict telles celles employées pour les arbres phylogénétiques, toutefois ces dernières impliquent une résolution de la cns résultant de l'agrégation beaucoup plus fine que la résolution des cns ayant été agrégées ;
- des méthodes de combinaisons de cns telles que chacune des cns participant à l'agrégation provient d'un processus de cns exécuté sur un jeu de données commun.

Dans ce chapitre, nous considérons la problématique particulière de la combinaisons de multiples cns tout en n'accédant pas au jeu de données initial, et ce, sans imposer aux cns à agréger de traiter les mêmes objets ou de considérer les mêmes descripteurs (variables) pour leurs traitements.

Ainsi poser, la problématique n'apparaît pas clairement et la différentiation avec les approches classiques pour la combinaison de cns n'est, elle aussi, pas évidente. Afin de clarifier nos propos, notons dès maintenant que la résolution de cette problématique permet d'apporter des solutions à une gamme plus vaste de problèmes ne se restreignant pas uniquement à l'accroissement de la qualité des cns bâties mais touchant également à l'accélération du processus de cns, à sa réalisation sur des données distribuées physiquement, à l'élargissement des types de données traitables...

Nous avions abordé cette problématique dès 2002, sans connaissance de travaux plus aboutis réalisés de manière concomitante par Alexander Strehl et Joydeep Gosh. En effet, dans une série d'article de la même année ([SG02a], [SG02b], [Str02], [GSS02]), Strehl et Gosh définissent cette problématique et présente largement les divers intérêts qu'elle revêt. Nous nous basons donc ici sur ces travaux afin d'introduire correctement cette problématique qu'ils ont baptisée "Cluster Ensembles".

Le problème sous jacent à la problématique "Cluster Ensembles" est donc le suivant :

Déterminer la cns la plus "en accord" avec un ensemble de cns sachant que :

- *La composition (en terme d'objets) de chaque classe de chacune des cns devant être agrégées est l'unique information dont on dispose (i.e. pour chaque objet du jeu de données et pour chaque cns on ne dispose que du numéro de la classe à laquelle l'objet appartient);*

- *chacune des cns devant être agrégée ne portent pas forcément sur le même ensemble d'objets ;*
- *les processus de classification ayant menés à l'établissement des diverses cns à agréger ne sont pas similaires (emplois de différentes méthodes, de différentes mesures de similarité/distance, traitement d'ensembles de variables différents, nombre et forme des classes différents...)*

6.1.1 Illustration de la Problématique "Cluster Ensembles"

Nous illustrons cette définition non formelle de la problématique grâce à un exemple tiré de la thèse de Alexander Strehl, exemple visant justement à introduire la problématique "Cluster Ensembles" :

Considérons les 7 points de l'espace bidimensionnel de la figure 6.1. Quatre vues (quatre cns) différentes (celles de quatre observateurs virtuels par exemple) sont proposées et correspondent à des projections orthogonales des données sur des segments de droites. A chaque segment correspond ainsi une zone d'observation qui n'inclue pas obligatoirement l'ensemble des 7 points. Chaque classe des quatre cns est représentée par une ellipse, notons que ces ellipses possèdent des formes différentes à l'intérieur de même cns ou encore d'une cns à l'autre, notons également que le nombre de classes peut varier d'une cns à l'autre.

Le problème est alors de déterminer la cns la plus en adéquation avec ces 4 cns, et ce, sur l'unique base de la composition des classes de chacune des cns qu'on peut noter:

- $P_1 = \{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o_6,o_7\}\}$
- $P_2 = \{\{o_6,o_7\},\{o_4,o_5\},\{o_1,o_2,o_3\}\}$
- $P_3 = \{\{o_1,o_2\},\{o_3,o_4\},\{o_5,o_6,o_7\}\}$
- $P_4 = \{\{o_1,o_4\},\{o_2,o_5\}\}$.

Comme nous pouvons l'observer les 2 premières cns (P_1 et P_2) sont identiques, la troisième cns (P_3) introduit des différences essentiellement pour les objets o_3 et o_5, quant à la quatrième (P_4) elle est extrêmement différente des autres cns et ne considère pas l'ensemble complet des 7 objets. Si on recherche une cns correspondant à une bonne agrégation des cns initiales, il apparaît alors intuitivement que cette cns agrégée doit partager le plus d'information avec les 4 cns initiales. Ainsi, la cns en 3 classes $P_5 = \{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o6,o_7\}\}$ semble un choix intéressant, ce choix se confirmerait si on considérait les 310 cns (partitions) possibles de 7 objets en 3 classes (et que l'on jugeait de l'adéquation de cette cns avec les 4 cns initiales en utilisant les mesures introduites ultérieurement).

Métaphoriquement on peut donc voir la problématique "Cluster Ensembles" comme un problème d'agrégation de "vues" différentes sur un jeu de données

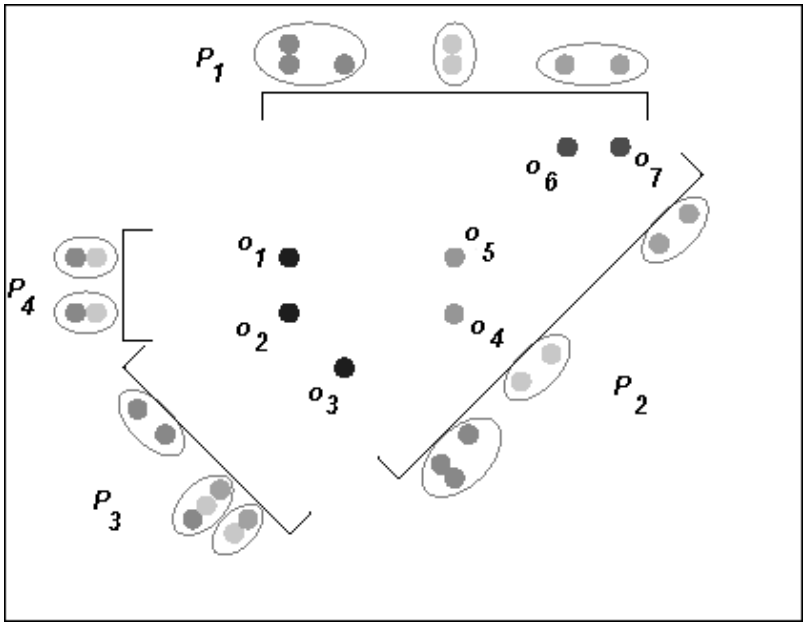


FIG. 6.1 –: *Illustration de la Problématique "Cluster Ensembles"*

(chacune des cns à agréger constituant alors une des "vues" sur le jeu de données) ou encore comme un problème d'agrégation d'opinions de juges évaluant une similarité entre objets du jeu de données (chacune des cns à agréger constituant alors un juge exprimant son opinion sur la similarité entre objets d'un sous-ensemble des objets du jeu de données).

6.1.2 Motivations, Objectifs de la Problématique "Cluster Ensembles"

La mise au point de méthodes pour la résolution de la problématique "Cluster Ensembles" peut schématiquement être motivée par différentes raisons :

- la réutilisation de connaissances existantes sous formes de cns,
- la réalisation de cns sur des données physiquement distribuées sans impliquer une centralisation totale de ces données,
- la réalisation de cns sur des données très hétérogènes ne pouvant être raisonnablement traitées par une unique méthode de cns,
- l'accélération du processus de cns, l'accroissement de la qualité et de la robustesse de cns...

Chacun de ces points possède un intérêt indéniable pour le processus E.C.D. . Nous les abordons maintenant au travers de deux points particuliers : la réutilisation de connaissances et le calcul distribué pour la cns.

6.1.2.1 Réutilisation de Connaissances

La possibilité de réutilisation de cns pré-existantes constitue un apport intéressant de la problématique "Cluster Ensembles". En effet, dans nombre de situations, une gamme de cns concernant un ensemble particulier d'objets peut déjà exister, et on peut alors désirer intégrer l'ensemble de ces cns en une unique ou encore réutiliser ces informations existantes de manière à influencer une nouvelle cns (basée ou non sur les mêmes variables). (On peut par exemple penser aux sociétés désireuses de réaliser une typologie de leurs clients et possédant déjà d'anciennes typologies pertinentes qu'il serait intéressant d'utiliser pour la mise au point de la nouvelle typologie ou encore à l'intégration d'avis d'experts pour la réalisation de typologies...) La réutilisation de connaissances pré-existantes peut donc nous aider à établir une nouvelle cns en exploitant l'information qu'elles contiennent, et ce, sans revenir sur les données et processus ayant permis de les mettre à jour.

6.1.2.2 Calcul Distribué pour la cns

Le besoin de mener à bien des processus de fouille de données de manière distribuée s'accroît continuellement. Cela s'explique naturellement par l'acquisition et le stockage de données dans des lieux géographiquement distribués pour des raisons organisationnelles ou opérationnelles [KC00], et par la nécessité de traiter autant que possible les données in situ. Notons que cette situation contraste avec l'hypothèse généralement admise pour les méthodes de fouille de données qui implique une disponibilité des données en un lieu unique. On pourrait argumenter que par un transfert de l'ensemble des données provenant des différents sites en un site unique, transfert, associé à une série de regroupement des données, il est possible d'obtenir un unique (et certainement très volumineux) fichier plat tel que l'utilisation d'algorithmes classiques est alors réalisable (au prix parfois d'un échantillonnage dans le fichier s'il est trop volumineux). Cependant, en pratique, la centralisation des données distribuées peut s'avérer impossible, ou au moins largement pénalisante, pour des raisons touchant aux coûts calculatoire, de stockage, ou de transfert des données. Des contraintes extérieures à ces considérations informatiques et technologiques peuvent également rendre la centralisation impossible, on peut ainsi penser à des raisons de sécurité, de propriétés des données, ou encore de distribution des données associée à des contraintes légales, des contraintes de traitement en temps réel, etc [PCS00]... Notons enfin que la sévérité de telles contraintes est devenue récemment évidente aux USA simultanément aux essais de plusieurs agences gouvernementales pour intégrer leurs bases de données et leurs techniques analytiques[GSS02].

La réalisation de cns de manière distribuée constitue l'avantage principal de la résolution de la problématique "Cluster Ensembles" car outre le traitement de données physiquement distribuées cela permet également :

– l'accroissement de la qualité et de la robustesse de cns,

– l'accélération du processus de classification (qu'on utilise réellement une distribution (parallélisation) des calculs ou qu'on simule cette distribution en procédant séquentiellement à certains types de calculs),
– le traitement de données très hétérogènes.

Nous abordons ici ces divers points en introduisant notamment 3 scénarios différents pour la distribution des données : (1) données distribuées sur les objets (**DDO**), (2) données distribuées sur les variables (**DDV**), (3)données distribuées sur objets et variables (**DDOV**).

1. *Données distribuées sur les variables* (**DDV**), dans ce scénario chacun des processus de classification ayant donné le jour aux cns initiales (à agréger) n'a pas eu accès à l'intégralité des variables caractérisant les objets mais seulement à un ensemble particulier de variables caractérisant les objets à traiter. On peut donc voir ce scénario comme l'agrégation de "vues" partielles et différentes sur les variables caractérisant les objets du jeu de données. Un exemple représentatif pourrait par exemple être la réalisation d'une taxonomie de patients d'un service hospitalier atteint d'une pathologie organique similaire : admettons que pour chacun des patients on dispose d'une "image" (IRM, radiographie,...) de l'organe malade, d'un ensemble de compte-rendus d'experts médicaux ainsi qu'un ensemble d'informations numériques concernant par exemple l'évolution de la température du patient, de sa pression sanguine... L'utilisation du scénario **DDV** s'avère ici très intéressante puisqu'elle peut permettre d'agréger des taxonomies des patients réalisées selon le point de vue de chacun des types d'informations à disposition et ce en utilisant à chaque fois une méthode adaptée pour la découverte de la taxonomie (i.e. on peut ainsi agréger une taxonomie réalisée sur la base des "images" des patients obtenues par une méthodes d'analyse particulière, une taxonomie des patients obtenues par application de méthodes de text-mining sur les comptes-rendu d'experts,...).
   Le scénario **DDV** permet donc d'**adapter le processus de cns à une distribution physique des données** (pour l'exemple, on peut imaginer que chaque type d'information est stockée dans une machine spécifique éventuellement associée à un service hospitalier spécifique) ; ce scénario permet également la **réalisation de cns sur des données très hétérogènes** (dans l'exemple illustratif les données sont fortement hétérogènes puisqu'elles intègrent simultanément les média textes, images ainsi que des séries chronologiques) ; on peut également penser à **limiter les problèmes d'encombrement mémoire liés au processus de cns** dans des cas où le nombre de variables est très important ; enfin ce scénario peut également engendrer, comme nous le montrerons ultérieurement, une **amélioration de la qualité des cns** grâce au phénomène d'agrégation de "vues" différentes sur les données.

2. *Données distribuées sur les objets* (**DDO**), il s'agit ici d'un scénario complémentaire au scénario **DDV**. Dans ce scénario chacune des cns initiales n'a pas été réalisée en considérant l'intégralité des objets devant être présent dans la cns agrégée mais un sous-ensemble spécifique d'objets. Ce scénario peut naturellement résulter de contraintes opérationnelles dans des situations réelles. Considérons par exemple le cas de datamarts de boutiques d'une même chaîne de magasin, ces datamarts stockent uniquement les informations concernant les ventes et clients locaux. Des segmentations de la clientèle sont souvent réalisées localement (au niveau d'une boutique particulière), agréger les résultats de ces analyses locales peut alors permettre une segmentation holistique de la clientèle de la chaîne de magasins sans pour autant nécessiter une centralisation des informations stockées localement. Notons qu'il est évidemment indispensable que les objets devant être présent dans la cns agrégée doivent également être présents dans au moins une des cns à agréger (i.e. pour qu'une personne apparaisse dans la segmentation globale de la clientèle il faut qu'elle soit présente dans au moins une des segmentations locales), et, qu'un certain niveau de recouvrement entre les ensembles d'objets traités par les cns à agréger est également indispensable si on désire réaliser l'agrégation avec succès (i.e. il est indispensable que des clients soient présents dans plusieurs segmentations locales pour obtenir une segmentation globale de bonne qualité). Pour l'exemple illustratif, la première contrainte est évidemment respectée, quant à la contrainte de recouvrement, on peut espérer qu'un nombre suffisant de clients s'approvisionnent dans plusieurs magasins de la chaîne de manière à ce que le niveau de recouvrement nécessaire soit atteint.

   Ce scénario est en premier lieu propice à l'**accélération du processus de classification**. Considérons en effet que l'on a réalisé un échantillonnage des objets d'un jeu de données comprenant n objets en k échantillons différents tels que chacun des objets soit en moyenne présents dans v échantillons différents (afin qu'il y ait un recouvrement entre échantillons). Pour simplifier, considérons que chacun de ces échantillons possèdent le même nombre d'objets : le nombre d'objets présents dans chaque échantillon est alors $\frac{nv}{k}$. Considérons également que les algorithmes de cns utilisés possèdent une complexité en $O(n^2)$ et que la complexité de la méthode d'agrégation soit linéaire selon le nombre d'objets ($O(n)$) ou encore log-linéaire selon le nombre d'objets ($O(n \times log(n))$). Réaliser une cns sur l'ensemble des objets implique un coût calculatoire en $O(n^2)$. Réaliser séquentiellement les cns sur les k échantillons puis procéder à l'agrégation implique par contre un coût calculatoire en $O(k \times \frac{n^2v^2}{k^2} + n \times log(n))$ (dans le cas de l'utilisation d'une méthode d'agrégation de complexité log-linéaire). Asymptotiquement (si le nombre d'objets n est important) le coût calculatoire associé à la méthode d'agrégation devient négligeable devant celui nécessaire aux cns et l'on peut approximer le coût calcu-

latoire du processus de cns ainsi réalisé par $O(k \times \frac{n^2v^2}{k^2})$. Il en résulte ainsi une diminution du coût calculatoire d'un facteur $\frac{k}{v^2}$. On peut également envisager réaliser les k processus de cns non pas séquentiellement mais de manière parallèle, le coût calculatoire étant alors réduit d'un facteur $\frac{k^2}{v^2}$. Notons toutefois que nous avons ici négligé le coût calculatoire associé à l'échantillonnage ainsi que celui associé au transfert. Cependant dans le cas d'une distribution physique réelles des données le coût d'échantillonnage est nul (car inexistant) et le coût de transfert est plus important pour la réalisation d'une cns en centralisant les données puisqu'il faut transmettre l'ensemble des caractéristiques des objets alors que si on réalise la cns de manière distribuée le coût de transfert est amoindri puisqu'il suffit de transmettre uniquement la composition des classes des différentes cns. Tout comme pour le scénario **DDV** on peut ici aussi envisager une **amélioration de la qualité des cns** grâce au phénomène d'agrégation de "vues" différentes sur les données.

3. *Données distribuées sur objets et variables* (**DDOV**), ce scénario correspond à une combinaison des deux scénarios précédemment décrits : dans ce cas les processus de classification ayant donné le jour aux cns initiales n'ont eu accès qu'à un sous ensemble des objets devant être présent dans la cns agrégée, ainsi qu'à un sous ensemble des variables caractérisant ces mêmes objets. Ce scénario possède les avantages combinés des deux scénarios précédents.

Enfin, on peut envisager d'améliorer la qualité ainsi que la robustesse des cns par agrégation d'un ensemble de cns considérant l'ensemble des objets et des variables mais obtenues par l'intermédiaire d'algorithmes multiples ou encore d'algorithmes similaires mais paramétrés différemment. Cette pratique appelée Robust Centralized Clustering par Strehl et Gosh permet également à l'utilisateur de s'affranchir des tâches ardues que sont le choix de l'algorithme de cns à adopter ainsi que le choix des paramètres.

6.1.3 Travaux Liés

Nous l'avons indiqué précédemment, il existe une multitude de travaux concernant l'agrégation de modèles de classification supervisé ou de régression mais relativement peu concernant l'agrégation de cns. Nous listons ici un ensemble de travaux contribuant cependant à ce champ de recherche :

– dans le cadre de la reconnaissance de formes, un ensemble de travaux essentiellement théoriques concernant la mise au point de cns consensuelles ont été réalisés au milieu des années 80 [BLM86]. Pour ces études, le terme de cns est à prendre dans une acceptation relativement large puisqu'il regroupe les notions de partitions, dendrogrammes, n-arbres.

L'objectif était alors, étant donné un ensemble de cns, d'obtenir une cns reflétant un consensus strict si bien que les résultats obtenus s'apparentaient souvent à une classification grossière de tous les objets d'un jeu de données (une partition en une unique classe) ou alors en une classification extrêmement fine (partition possédant un nombre de classes proche du nombre d'objets compris dans le jeu de données). De plus, les techniques mises au point possédaient dans la plupart des cas un coût calculatoire important et très largement prohibitif pour une utilisation sur des jeux de données volumineux. Nous pouvons également citer des travaux plus récent de cette communauté concernant la mise au point de partitions de partitions [GV98].

- Des techniques telles que celles présentées dans [DLP82], [FRB98] proposent de combiner les résultats de plusieurs cns d'un jeu de données commun (objets et variables identiques) (il s'agit surtout ici de l'agrégation de cns provenant de processus de classification de type K-Means initialisés différemment).
- Une méthode de fouille de données collective (Collective Data mining) introduite dans [JK99] permet de combiner des cns distribuées obtenues par des processus d'agrégation n'accédant qu'à des sous-ensembles partielles des variables.
- Des méthodes utilisant le paradigme Rough Sets, méthodes proposées dans [HT01] et [HTO$^+$02], sont également relativement proche de cette problématique.

Ces travaux, bien que concernant l'agrégation de cns, n'abordent toutefois que partiellement la problématique "Cluster Ensembles" ; en effet, les diverses approches proposées ne considèrent jamais l'intégralité des scénarios de distribution des données (**DDV**, **DDO**, **DDOV**), ni ne prennent en compte et présentent l'ensemble des avantages associés à la mise en œuvre de méthodes d'agrégation de cns. Seuls les papiers de Strehl et Gosh, et dans une moindre mesure les nôtres, introduisent l'ensemble des ces éléments.

Enfin, simultanément à leur définition de la problématique "Cluster Ensembles", Strehl et Ghosh ont proposé trois approches différentes pour la résolution de ce problème :

- la méthode **CSPA** (Cluster based Similarity Partitioning Algorithm) qui consiste en une heuristique recherchant une cns en k classes qui minimise une mesure d'adéquation spéciale (cette mesure est proche de celle que nous introduisons par la suite, quant à la méthode, son fonctionnement est également proche de celui de la première méthode que nous proposons). Sa complexité calculatoire est en $O(n^2kr)$ avec n le nombre d'objets, k le nombre de classes de la cns provenant de l'agrégation et r le nombre initiales de cns à agréger.
- la méthode **HGPA** (HyperGraph Partitioning Algorithm) basée sur l'approche HMETIS pour le partitionnement d'hypergraphes [KARS97], cette

méthode recherche une cns en k classes. Sa complexité calculatoire est en $O(nkr)$.

– la méthode **MCLA** (Meta-CLustering Algorithm) basée sur la classification de cns ; cette méthode permet de déterminer une cns en k classes. Sa complexité calculatoire est en $O(nk^2r^2)$.

Selon Strehl et Gosh, MCLA et CSPA semble fournir des cns de qualités similaires tandis que HGPA semble fournir des cns de moins bonne qualité. Ces 3 méthodes partagent la nécessité de fixer a priori le nombre final de classes.

Chacune de ces méthodes consiste en définitive en un processus d'optimisation visant à déterminer la cns "la plus en accord" avec un ensemble donné de cns. Ainsi, étant donné un ensemble E de r cns ($E = \{P_i, i = 1..r\}$), le problème est de déterminer la cns $P_\star$ telle qu'elle optimise une fonction $\Gamma(E,P_\star)$ rendant compte de "l'accord" entre les cns de E et $P_\star$. Pour Strehl et Gosh, dans la mesure où la cns $P_\star$ se doit de partager le plus d'information possible avec les cns de E, la fonction Γ utilisée se base sur la théorie de l'information. Plus précisément, il propose d'utiliser l'information mutuelle qui est une mesure symétrique permettant de quantifier l'information statistique partagée par deux distributions. Cette mesure est définie de la manière suivante :
supposons que nous disposions de deux cns P_1 et P_2 telles que P_1 (resp. P_2) est composée de k^{P_1} (resp. k^{P_2}) classes. Soient n le nombre total d'objets, $n^{(h)}$ le nombre d'objets appartenant à la classe C_h de P_1 et n_l le nombre d'objets appartenant à la classe C_l de P_2. Soit $n_l^{(h)}$ le nombre d'objets présents à la fois dans la classe C_h de P_1 et dans la classe C_l de P_2. La mesure symétrique normalisée d'information mutuelle entre deux cns P_1 et P_2 $\varphi^{(NSMI)}(P_1,P_2)$ (NSMI : Normalized Symmetric Mutual Information) est définie ainsi :
$\varphi^{(NSMI)}(P_1,P_2) = \frac{2}{n} \sum_{l=1..k^{(P_1)}} \sum_{h=1..k^{(P_2)}} n_l^{(h)} log_{k^{(P_1)}.k^{(P_2)}} \left(\frac{n_l^{(h)} n}{n^{(h)} n_l} \right)$
($\varphi^{(NSMI)}(P_1,P_2) \in [0;1]$).

On peut également définir $\varphi^{(ANSMI)}(E,P_\#)$ (ANSMI : Average Normalized Symmetric Mutual Information) la moyenne de la mesure Symétrique Normalisée d'Information Mutuelle entre un ensemble E de r cns et une cns $P_\#$:
$\varphi^{(ANSMI)}(E,P_\#) = \frac{1}{r} \sum_{q=1..r} \varphi^{(NSMI)}(P_\#,P_q)$.

A partir de cette mesure, Strehl et Gosh ont défini la cns $P_\star$ comme la cns maximisant la valeur de la mesure $\varphi^{(ANSMI)}(E,P_\#)$.

Notons que cette mesure symétrique est biaisée en faveur de cns possédant un nombre relativement faible de classes. Il existe une mesure du même type mais non symétrique $\varphi^{(ANAMI)}(E,P_\#)$ (ANAMI : Average Normalized Asymmetric Mutual Information), cette mesure peut également être utilisée, mais, elle est biaisée en faveur de cns présentant un nombre de classes plus important.

6.1.4 Principaux Challenges pour la Problématique "Cluster Ensembles"

Toujours selon Strehl et Gosh, les principaux problèmes associés à la problématique "Cluster Ensembles" concernent :

- l'agrégation de cns aux formes différentes et ayant des classes en nombre différents ;
- la "non-connaissance" a priori du nombre final de classes pour la cns résultant de l'agrégation.

Nous proposons dans les sections suivantes :

- *une mesure alternative à la mesure d'information mutuelle pour la définition de la cns "la plus en accord" avec un ensemble donné de cns ;*
- *trois méthodes permettant l'agrégation de cns dans le cadre de la problématique "Cluster Ensembles". Ces trois méthodes se basant sur une optimisation "directe" de la mesure alternative préalablement introduite :*
  - *deux de ces méthodes possèdent le fort avantage de ne pas nécessiter de fixer a priori le nombre final de classes pour la cns résultant de l'agrégation ;*
  - *une des méthodes proposées exhibe un coût calculatoire extrêmement réduit ;*
  - *de plus, aucune des 3 méthodes n'est handicapée lors de l'agrégation de cns ayant des classes en nombres très différents et chacune permet l'obtention de résultats de bonne qualité.*

Nous introduisons donc dans un premier temps la mesure alternative, puis présentons les trois méthodes, et enfin procédons à l'évaluation expérimentale des deux méthodes les plus intéressantes.

6.2 Mesures d'Adéquation

Nous introduisons maintenant l'ensemble des notations et formalismes que nous utilisons par la suite afin de présenter les trois méthodes que nous proposons pour la résolution de la problématique "Cluster Ensembles". L'objectif final de cette section est de présenter une mesure d'adéquation entre un ensemble de cns et une unique cns. La découverte de la cns minimisant la valeur de cette mesure pour un ensemble donné de cns (ce qui signifie la découverte de la cns la plus en adéquation avec un ensemble donné de cns) constituera plus tard le problème d'optimisation à résoudre pour la problématique "Cluster Ensembles".

Notation 2
$O = \{o_i, i = 1..n\}$ l'ensemble des objets de la cns issue de l'agrégation de multiples cns,
$C_k^O = \{o_i, i = 1..n_{C_k^O}\}$ un ensemble d'objets de O ($C_k^O \subseteq O$),

$P_w = \{C_1^O,...,C_h^O\}$ *une cns de* O *en* h *classes* ($\forall i = 1..h$, $\forall j = 1..h$, $j \neq i$, $\forall o \in C_i^O$, $o \notin C_j^O$)

6.2.1 Adéquation entre Classifications Non Supervisées

6.2.2 Adéquation pour un Couple de Classification Non Supervisée

Afin de représenter l'adéquation entre 2 cns P_1 et P_2 (avec $P_1 = \{C_1^{O_1},...,C_l^{O_1}\}$, $O_1 \subseteq O$ et $P_2 = \{C_1^{O_2},...,C_m^{O_2}\}$, $O_2 \subseteq O$), nous utilisons une mesure classique d'adéquation entre cns définie comme le ratio suivant :
$\frac{nombre\ de\ désaccords\ entre\ les\ 2\ cns}{nombre\ de\ désaccords\ et\ d'accords\ entre\ les\ 2\ cns}$.

Cette mesure, notée $Adq(P_1,P_2)$, est plus formellement définie comme suit :

$$Adq(P_1,P_2) = \begin{cases} \frac{DisAgg(P_1,P_2)}{Agg(P_1,P_2)+DisAgg(P_1,P_2)} & \text{si} \quad Agg(P_1,P_2) + DisAgg(P_1,P_2) \neq 0 \\ 0 & \text{si} \quad Agg(P_1,P_2) + DisAgg(P_1,P_2) = 0 \end{cases} \tag{6.1}$$

avec,

$$Agg(P_1,P_2) = \sum_{o_i \in O_1, o_j \in O_2, o_i \neq o_j} \delta_1(o_i,o_j) \tag{6.2}$$

$$DisAgg(P_1,P_2) = \sum_{o_i \in O_1, o_j \in O_2, o_i \neq o_j} \delta_2(o_i,o_j) \tag{6.3}$$

$$\delta_1(o_i,o_j) = \begin{cases} 1 \text{ si : } (\exists C_f^{O_1}\ (f \in \{1,...,l\}) \text{ telle que } o_i \in C_f^{O_1} \text{ et } o_j \in C_f^{O_1}) \\ \text{et } (\ \exists C_g^{O_2}\ (g \in \{1,...,m\}) \text{ telle que } o_i \in C_g^{O_2} \text{ et } o_j \in C_g^{O_2}) \\ 1 \text{ si : } (o_i \in O_1 \text{ et } o_j \in O_1 \text{ et } \nexists C_f^{O_1} \text{ telle que } o_i \in C_f^{O_1} \text{ et } o_j \in C_f^{O_1}) \\ \text{et } (o_i \in O_2 \text{ et } o_j \in O_2 \text{ et } \nexists C_g^{O_2} \text{ telle que } o_i \in C_g^{O_2} \text{ et } o_j \in C_g^{O_2}) \\ 0 \text{ sinon} \end{cases} \tag{6.4}$$

$$\delta_2(o_i,o_j) = \begin{cases} 1 - \delta_1(o_i,o_j) \text{ si : } (o_i \in O_1 \text{ et } o_j \in O_1) \text{ et } (o_i \in O_2 \text{ et } o_j \in O_2) \\ 0 \text{ sinon} \end{cases} \tag{6.5}$$

Conséquemment, plus $Adq(P_1,P_2)$ est proche de 0 plus les 2 cns peuvent être considérées comme étant en adéquation. Cependant, $Agg(P_1,P_2)+DisAgg(P_1,P_2) = 0$ implique $DisAgg(P_1,P_2) = 0$ ce qui ne signifie pas une bonne adéquation entre ces 2 cns mais simplement qu'elles ne possèdent aucun objet en commun.

REMARQUE :
$Agg(P_1,P_2) + DisAgg(P_1,P_2) = \frac{card(O_1 \bigcap O_2)(card(O_1 \bigcap O_2)-1)}{2}$.
Ainsi, la valeur $Agg(P_1,P_2) + DisAgg(P_1,P_2)$ est indépendante de la forme

des cns P_1 et P_2, elle dépend seulement du nombre d'objets que possèdent en commun ces 2 cns.

6.2.3 Adéquation entre une Classification Non Supervisée et un Ensemble de Classifications Non Supervisées

Afin de représenter l'adéquation entre un ensemble de cns $E = \{P_1,...,P_z\}$ ($\forall i = 1..z, P_i = \{C_1^{O_i},...,C_{li}^{O_i}\}, O_i \subseteq O$, et $\bigcup O_i = O$) et une unique cns de O ($P = \{C_1^O,...,C_l^O\}$), nous utilisons une généralisation de la mesure d'adéquation pour un couple de cns, nous la notons $Adq(E,P)$. Elle est définie comme le ratio : $\frac{nombre\ de\ désaccords\ entre\ P\ et\ les\ cns\ de\ E}{nombre\ d'accords\ et\ de\ désaccords\ entre\ P\ et\ les\ cns\ de\ E}$.

Cette mesure est formellement définie de la manière suivante :

$$Adq(E,P) = \frac{\sum_{P_i \in E}(DisAgg(P_i,P))}{\sum_{P_i \in E}(Agg(P_i,P) + DisAgg(P_i,P))} \tag{6.6}$$

Conséquemment, plus $Adq(E,P)$ est proche de 0 plus l'ensemble de cns E et la cns P peuvent être considérés comme étant en adéquation (remarquons que $\sum_{P_i \in E} Agg(P_i,P) + \sum_{P_i \in E} DisAgg(P_i,P) > 0$ car les cns de E et P ont forcément des objets en commun, car $\bigcup O_i = O$).

REMARQUE :

$$\sum_{P_i \in E}(Agg(P_i,P) + DisAgg(P_i,P)) = \sum_{P_i \in E} \frac{card(O_i \bigcap O)(card(O_i \bigcap O) - 1)}{2} \tag{6.7}$$

Ainsi, la valeur $\sum_{P_i \in E}(Agg(P_i,P) + DisAgg(P_i,P))$ est indépendante de la forme des cns de E et de la forme de P, elle dépend seulement du nombre d'objets en commun pour chaque paire de cns (P_i, P) .

6.3 Contribution à la Problématique "Cluster Ensembles" : Trois Méthodes pour l'Agrégation de Classifications Non Supervisées

Nous avons proposé dans [JN03e], [JN03d] deux nouvelles méthodes pour l'agrégation de cns dans le cadre de la Problématique "Cluster Ensembles", nous adjoindrons ici la description d'une troisième méthode.

Le problème à résoudre est :
"Etant donné un ensemble de cns $E = \{P_1,...,P_z\}$, déterminer la cns $P_\star$ telle que $Adq(E,P_\star)$ soit minimisée, i.e. déterminer la cns $P_\star$ la plus en adéquation avec E"
($\forall i = 1..z, P_i = \{C_1^{O_i},...,C_{li}^{O_i}\}$; $O_i \subseteq O$ et $\bigcup O_i = O$; $P_\star = \{C_1^{O_\star},...,C_{l_\star}^{O_\star}\}$, $O_\star = O$).

Ce problème est combinatoire et le coût de la recherche d'une solution optimale est extrêmement élevé si O est composé d'un grand nombre d'objets. Nous proposons donc ici trois algorithmes gloutons pour la découverte d'une solution, qui si elle n'est pas toujours optimale, constitue en tout cas une solution de bonne qualité. Les 2 premiers algorithmes proposés possèdent l'avantage de ne pas nécessiter de faire d'hypothèses sur la forme de la cns $P_\star$: il n'est pas nécessaire de spécifier a priori le nombre de classe de cette cns. Enfin, la complexité du premier est quadratique selon le nombre d'objets de O, la complexité est log-linéaire selon le nombre d'objets de O pour le deuxième et la complexité est linéaire selon le nombre d'objets de O pour le troisième.

6.3.1 Première Méthode pour l'Agrégation de cns : Une Méthode Intuitive

La première méthode proposée, que nous n'introduisons que partiellement ici [1] suit le principe intuitif suivant :

- Pour chaque couple d'objets (o_i,o_j) de $P_\star$, on évalue tout d'abord combien de fois (s) les 2 objets sont réunis au sein d'une même classe d'une cns de l'ensemble E puis on évalue combien de fois (d) les 2 objets sont séparés dans deux classes différentes d'une cns de E. Pour chaque couple, les valeurs s et d donnent une idée du traitement majoritaire du couple d'objets dans les cns de E dans lesquelles ces 2 objets sont présents (i.e. cela montre si o_i et o_j sont plus souvent réunis au sein d'une même classe ou séparés dans deux classes différentes).
- Grâce à ces informations, on peut déterminer les objets devant être prioritairement séparés dans deux classes différentes de $P_\star$ ou réunis au sein d'une même classe de $P_\star$ afin de maximiser l'adéquation entre E et $P_\star$. En effet, plus la valeur $|s-d|$ est élevée pour un couple d'objets, plus $P_\star$ doit respecter le traitement majoritaire imposé par les cns de E (séparation ou union) pour ce couple d'objets si on veut que $adq(E,P_\star)$ soit minimisée (i.e. si on veut maximiser l'adéquation entre E et $P_\star$).
- Ainsi, on utilise une méthode gloutonne qui construit élément par élément la matrice d'adjacence de $P_\star$ en considérant les couples d'objets selon l'ordre décroissant sur leur valeur $|s-d|$ afin de déterminer quels objets doivent être réunis ou séparés.

REMARQUE : Cet algorithme peut ne pas aboutir à une unique cns mais à un ensemble de cns équivalentes du point de vue de la mesure d'adéquation avec l'ensemble E. Sa complexité est quadratique selon n le nombre d'objets de O.

1. la présentation de cette méthode n'est que partielle car son intérêt est moindre par rapport à la seconde méthode introduite : en effet, à l'instar de la seconde méthode, cette méthode ne nécessite pas de fixer le nombre final de classes de la cns résultant de l'agrégation ; le niveau de qualité de ses résultats est équivalent à ceux des deux autres méthodes présentées, mais par contre, son coût calculatoire est plus important.

6.3.2 Seconde Méthode pour l'Agrégation de Classifications Non Supervisées : Utilisation de la Méthode KEROUAC

La méthode de cns KEROUAC présentée au chapitre 3 peut être utilisée afin de procéder à l'agrégation de cns, en lui imposant cependant un ensemble de contraintes. Nous explicitons cela dans cette section.

La méthode de cns pour données catégorielles KEROUAC consiste en la découverte d'une cns minimisant le critère $NCC^\star$ par l'intermédiaire d'un processus similaire aux graphes d'induction (i.e. à partir de la partition grossière, une succession de segmentations/fusions de classes permet de déterminer une partition minimisant le critère $NCC^\star$). Cette méthode possède de plus la capacité à déterminer par elle-même le nombre final de classes de la cns. Le résultat est alors une cns $P_\alpha = \{C_1^O,...,C_h^O\}$ telle que $NCC^\star(P_\alpha)$ est minimisé.

Nous rappelons ici la définition du critère $NCC^\star$

$$NCC^\star(P_\alpha) = \sum_{i=1..h,j=1..h,i>j} Sim(C_i^O,C_j^O) + gran \times \sum_{i=1}^{h} Dissim(C_i^O,C_i^O) \quad (6.8)$$

$gran$ est un scalaire positif, appelé facteur de granularité, dont la valeur est fixée par l'utilisateur

$$\begin{aligned} Sim(C_i^O,C_j^O) &= \sum_{o_a \in C_i^O, o_b \in C_i^O, a>b} sim(o_a,o_b) \\ sim(o_a,o_b) &= \sum_{i=1}^{p} \delta_{sim}(o_{a_i},o_{b_i}) \\ Dissim(C_i^O,C_j^O) &= \sum_{\substack{o_a \in C_\alpha^{O_i}, \\ o_b \in C_\alpha^{O_j}, a>b}} dissim(o_a,o_b) \\ dissim(o_a,o_b) &= \sum_{i=1}^{p} 1 - \delta_{dissim}(o_{a_i},o_{b_i}) \end{aligned}$$

$$\delta_{sim}(o_{a_i},o_{b_i}) = \delta_{dissim}(o_{a_i},o_{b_i}) = \begin{cases} 1 \text{ if } o_{a_i} = o_{b_i} \\ 0 \text{ if } o_{a_i} \neq o_{b_i} \end{cases} \quad (6.9)$$

6.3.2.1 Utilisation de KEROUAC pour la cns en considérant des Méta-Variables

Considérons la correspondance suivante :
Chaque cns $P_i \in E$ ($P_i = \{C_1^{O_i},...,C_{li}^{O_i}\}, O_i \subseteq O$) peut être vue comme une méta-variable que l'on peut coder sous la forme d'une variable catégorielle possédant un nombre de modalités égal à l_i le nombre de classes de P_i. De plus, on peut ajouter une modalité supplémentaire afin de coder l'éventuelle

absence d'un ou plusieurs objets de O dans O_i (i.e. absence d'objet(s) de O dans la cns P_i).

Ainsi, on peut dériver de l'ensemble de z cns E un ensemble de z meta-variables (nous notons cet ensemble $MF = \{mf_1,...,mf_z\}$).

On peut ensuite utiliser la méthode KEROUAC en fixant le facteur de granularité à 1 ($gran = 1$; la raison de ce choix est donnée ultérieurement) afin de réaliser une cns des objets de O décrits par les méta variables de MF. Dès lors chaque objet o_i de O peut être représenté par $o_i = \{o_{i_{mf_1}},...,o_{i_{mf_z}}\}$. La cns obtenue finalement, notée $P_\beta = \{C_1^O,...,C_g^O\}$ est telle qu'elle minimise le critère $NCC^\star$.

P_β minimise donc :

$$NCC^\star(P_\beta) = \sum_{i=1..g,j=1..g,i>j} Sim(C_i^O,C_j^O) + \sum_{i=1}^{g} Dissim(C_i^O,C_i^O) \qquad (6.10)$$

6.3.2.2 Relation entre $P_\star$ and P_β

La cns $P_\star$ doit être telle que son adéquation avec l'ensemble de cns E est maximisée, i.e. $Adq(E,P_\star)$ est minimisé. Concernant la cns P_β, elle doit être telle que $NCC^\star(P_\beta)$ est minimisé. Si nous étudions plus en détail les critères $Adq(E,P_\star)$ ainsi que $NCC^\star(P_\beta)$ et que nous adoptons une modification légère de la définition du critère $NCC^\star$ nous pouvons déterminer une forte relation unissant ces deux critères : ils sont unis par une relation de proportionnalité.

Explication :

- Pour $P_\star$, étant donné E, $P_\star$ est telle que $Adq(E,P_\star)$ est minimisée.
  $Adq(E,P_\star) = \frac{\sum_{P_i \in E} DisAgg(P_i,P_\star)}{\sum_{P_i \in E} Agg(P_i,P_\star)+DisAgg(P_i,P_\star)}$
- Pour P_β, à chaque cns à agréger P_i ($P_i = \{C_1^{O_i},...,C_{li}^{O_i}\}, O_i \subseteq O$) correspond une variable catégorielle dont le nombre de modalités est égal à $l_i + 1$, ces modalités sont notées $mod_{i_{mf_1}},...,mod_{i_{mf_{li}}},absent$, la modalité $absent$ est utilisée pour rendre compte du cas d'objets de O non présents dans O_i. Considérons que nous utilisons la version suivante légèrement modifiée des opérateurs $\delta_{sim}(o_{a_{mf_i}},o_{b_{mf_i}})$ et $\delta_{dissim}(o_{a_{mf_i}},o_{b_{mf_i}})$ pour définir le critère $NCC^\star$:

  $$\delta_{sim}(o_{a_{mf_i}},o_{b_{mf_i}}) = \begin{cases} 1 \text{ si } o_{a_{mf_i}} = o_{b_{mf_i}} \text{ et } o_{a_{mf_i}} \neq absent \\ 0 \text{ si } o_{a_{mf_i}} \neq o_{b_{mf_i}} \text{ ou si } o_{a_{mf_i}} = absent \text{ ou si } o_{b_{mf_i}} = absent \end{cases}$$

  $$\delta_{dissim}(o_{a_{mf_i}},o_{b_{mf_i}}) = \begin{cases} 1 \text{ si } o_{a_{mf_i}} = o_{b_{mf_i}} \text{ ou } o_{a_{mf_i}} = absent \text{ ou } o_{b_{mf_i}} = absent \\ 0 \text{ si } o_{a_{mf_i}} \neq o_{b_{mf_i}} \text{ et } o_{a_{mf_i}} \neq absent \text{ et } o_{b_{mf_i}} \neq absent \end{cases}$$

  Appliquer ces modifications correspond à ne pas prendre en compte les similarités et dissimilarités impliquées par la modalités $absent$, ce qui est

totalement naturel (on ne peut pas dire a priori que deux objets sont similaires car ils ne sont pas présents dans O_i, ou qu'un objet présent dans O_i est dissimilaire d'un autre objet non présent dans O_i).[2]

– Avec cette légère et naturelle modification nous avons alors :

$$\begin{aligned}\sum_{P_i \in E} DisAgg(P_i,P_\star) &= \sum_{i=1..g,j=1..g,i>j} Sim(C_\beta^{O_i},C_\beta^{O_j}) + \sum_{i=1}^{g} Dissim(C_\beta^{O_i},C_\beta^{O_i}) \\ &= NCC^\star(P_\beta)\end{aligned}$$

d'où,

$$\begin{aligned}Adq(E,P_\star) &= \frac{\sum_{P_i \in E}(DisAgg(P_i,P_\star)}{\sum_{P_i \in E} Agg(P_i,P_\star) + DisAgg(P_i,P_\star)} \\ &= \frac{NCC^\star(P_\beta)}{\sum_{P_i \in E} Agg(P_i,P_\star) + DisAgg(P_i,P_\star)}\end{aligned}$$

étant donné que
$\sum_{P_i \in E}(Agg(P_i,P_\star) + DisAgg(P_i,P_\star)) = \sum_{P_i \in E} \frac{card(O_i \bigcap O)(card(O_i \bigcap O)-1)}{2}$
(voir remarque page 177)

Nous avons donc :
$Adq(E,P_\star) = \frac{NCC^\star(P_\beta)}{\sum_{P_i \in E} \frac{card(O_i \bigcap O)(card(O_i \bigcap O)-1)}{2}}$

cela signifie clairement que $Adq(E,P_\star)$ et $NCC^\star(P_\beta)$ sont proportionnels.

6.3.2.3 Conclusion

Conséquemment, une cns $P_\sharp$ qui minimise $NCC^\star(P_\sharp)$ minimise alors également $Adq(E,P_\sharp)$. Ainsi, utiliser la méthode KEROUAC en accédant à l'ensemble méta-variables MF permet de résoudre le problème de l'agrégation de partitions dans le cadre de la problématique "Cluster Ensembles".

6.3.2.4 Illustration

Nous illustrons nos précédents propos sur l'exemple illustratif du début de chapitre : on considère l'ensemble d'objets $O = \{o_1,o_2,o_3,o_4,o_5,o_6,o_7\}$, l'ensemble de cns $E = \{P_1,P_2,P_3,P_4\}$ avec : $P_1 = \{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o_6,o_7\}\}$,

2. On peut également noter que ces travaux résultent d'une application directe des développements proposés au chapitre 4 pour l'introduction de contraintes et données manquantes dans la méthode KEROUAC.

$P_2 = \{\{o_4,o_5\},\{o_1,o_2,o_3\},\{o_6,o_7\}\}$, $P_3 = \{\{o_1,o_2\},\{o_3,o_4\},\{o_5,o_6,o_7\}\}$, $P_4 = \{\{o_1,o_4\},\{o_2,o_5\}\}$.

Nous pouvons ainsi bâtir l'ensemble de 4 méta-variables $MF = \{f_1,f_2,f_3,f_4\}$ afin de décrire les objets de O et associer à ces méta-variables 4 variables catégorielles (voir tableau 6.1).

Puis, nous pouvons utiliser la méthode de cns KEROUAC (en y intégrant la modification pour $NCC^\star$ afin de prendre en compte correctement la modalité $absent$). KEROUAC mènerait alors à l'obtention de la cns $\{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o_6,o_7\}\}$ qui correspond en définitive à la cns la plus en adéquation avec l'ensemble des cns de E. Nous résumons graphiquement la méthode dans la figure 6.2.

	f_1	f_2	f_3	f_4
o_1	a	b	a	a
o_2	a	b	a	b
o_3	a	b	b	$absent$
o_4	b	c	b	a
o_5	b	c	c	b
o_6	c	a	c	$absent$
o_7	c	a	c	$absent$

TAB. 6.1 –: *Description des objets par des Méta-Variables*

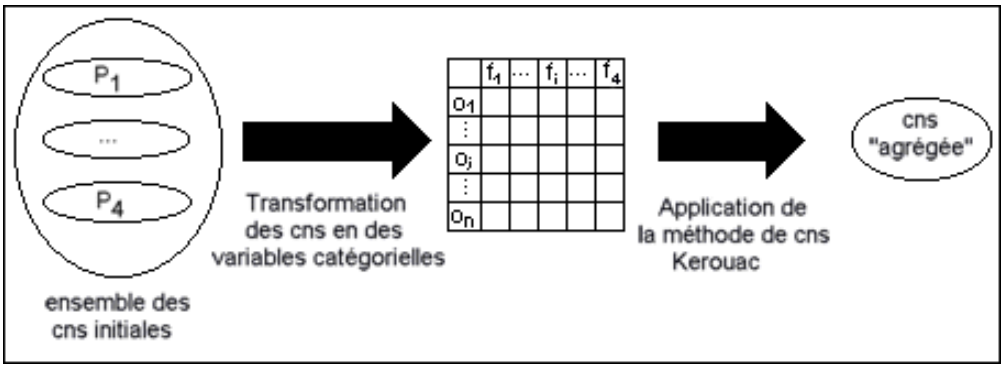


FIG. 6.2 –: *Utilisation de* KEROUAC *pour la problématique "Cluster Ensembles"*

6.3.2.5 Propriétés de la Méthode

Cette méthode hérite ces propriétés de la méthode de cns KEROUAC (voir chapitre 3) :

- sa complexité calculatoire est celle des graphes d'induction : $O((nr + k^2)log(n))$ avec n le nombre d'objets, k le nombre de classes de la cns obtenue par agrégation et r le nombre de cns initiales ;
- la scalabilité de la méthode semble bonne ;

– elle ne nécessite pas de fixer a priori le nombre final de classes pour la cns agrégée.

Etant donnée sa complexité calculatoire relativement faible, ainsi que sa capacité à déterminer automatiquement le nombre final de classes pour la cns agrégée notre méthode semble attractive. En outre, les évaluations expérimentales menées dans les sections suivantes montrent la très bonne qualité des cns agrégées ainsi que le faible impact de la présence de cns aux nombres de classes très différents sur la qualité des cns agrégées obtenues.

6.3.3 Troisième Méthode pour l'Agrégation de Classifications Non Supervisées : Utilisation de la Méthode K-Modes

L'idée est ici aussi d'utiliser une méthode de cns pour réaliser l'agrégation. Nous utilisons cette fois-ci la méthode de cns K-Modes qui consiste en définitive en une méthode d'optimisation recherchant une partition $P_\alpha = \{C_1^O,...,C_h^O\}$ en un nombre fixé de classes h telle qu'elle minimise le critère QKM.
$QKM(P_\alpha) = \sum_{i=1..h} \sum_{x \in C_i^O} d(x,mode^{C_i^O})$ avec $d(x,mode^{C_i^O}) = dissim(x,mode^{C_i^O})$).
Nous apporterons cependant à cette méthode quelques modifications comparables à celles mises en œuvre pour la méthode KEROUAC.

On considère tout comme pour la méthode KEROUAC l'ensemble de z meta-variables ($MF = \{mf_1,...,mf_z\}$) issues de l'ensemble de z cns E. Ainsi à chaque cns à agréger P_i ($P_i = \{C_1^{O_i},...,C_{li}^{O_i}\}$,$O_i \subseteq O$) correspond une variable catégorielle dont le nombre de modalités est égal à $l_i + 1$. Ces modalités sont notées $mod_{i_{mf_1}},...,mod_{i_{ncl_{mf_i}}},absent$, la modalité $absent$ est utilisée pour rendre compte du cas d'objets de O non présents dans O_i.

Les modifications apportées à la méthode K-Modes sont les suivantes :

– nous utilisons une version légèrement modifiée de l'opérateur $\delta_{dissim}(o_{a_{mf_i}},o_{b_{mf_i}})$ pour définir le critère QKM :

$$\delta_{dissim}(o_{a_{mf_i}},o_{b_{mf_i}}) = \begin{cases} 1 \text{ si } o_{a_{mf_i}} = o_{b_{mf_i}} \text{ ou } o_{a_{mf_i}} = absent \text{ ou } o_{b_{mf_i}} = absent \\ 0 \text{ si } o_{a_{mf_i}} \neq o_{b_{mf_i}} \text{ et } o_{a_{mf_i}} \neq absent \text{ et } o_{b_{mf_i}} \neq absent \end{cases}$$

– nous utilisons la définition particulière suivante pour le mode d'une classe :

Définition 11 *Le mode d'un ensemble d'objet C est l'objet virtuel $mode^C$ ($mode^C = \{mode_j^C, j = 1..p\}$) tel que pour toute variable $V_j \in EV$ la valeur d'attribut de $mode^C$ est, celle, la plus représentée pour cette variable au sein de la classe C en excluant toutefois la valeur $absent$:*
$\forall j = 1..p, \forall o_i \in C, f_r(V_j = mode_j^C|C) \geq f_r(V_j = o_{i_j}, o_{i_j} \neq absent|C)$.

Ainsi, en considérant l'ensemble des méta-variables, la méthode K-Modes recherche la cns $P_\beta = \{C_1^O,...,C_g^O\}$ est telle qu'elle minimise le critère QKM.

Si on considère les modifications apportées à la méthode K-Modes, on peut montrer qu'il existe une relation entre le critère QKM et le critère Adq. Si cette relation n'est pas aussi claire que celle unissant le critère $NCC^\star$ et le le critère Adq (dans ce cas la cns P_β minimisant $NCC(P_\beta)$ est identique à la partition $P_\star$ qui minimise $Adq(E,P_\star)$) : on peut montrer que la partition P_β en h classes minimisant le critère QKM tend à être proche de la partition $P_\star$ en h classes minimisant le critère $Adq(E,P_\star)$.

Ainsi, utiliser la méthode des K-Modes (légèrement modifiée) permet de déterminer une cns en h classes proche de la cns en h classes minimisant $Adq(E,P_\star)$. On peut ainsi utiliser cette méthode pour la résolution de la problématique d'agrégation de cns.

6.3.3.1 Illustration

Soit l'exemple illustratif de la section précédente :
Soient $O = \{o_1,o_2,o_3,o_4,o_5,o_6,o_7\}$, l'ensemble de cns $E = \{P_1,P_2,P_3,P_4\}$ avec :
$P_1 = \{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o_6,o_7\}\}$, $P_2 = \{\{o_4,o_5\},\{o_1,o_2,o_3\},\{o_6,o_7\}\}$,
$P_3 = \{\{o_1,o_2\},\{o_3,o_4\},\{o_5,o_6,o_7\}\}$, $P_4 = \{\{o_1,o_4\},\{o_2,o_5\}\}$.

Utiliser la méthode des K-Modes (en fixant le nombre final de classes à 3) pour réaliser l'agrégation mènerait soit à l'obtention de la cns
$P_a = \{\{o_1,o_2,o_3\},\{o_4,o_5\},\{o_6,o_7\}\}$ qui correspond en définitive à la cns la plus en adéquation avec l'ensemble des cns de E, soit à l'obtention de la cns $P_b = \{\{o_1,o_2,o_3\},\{o_4\},\{o_5,o_6,o_7\}\}$.

En effet, si ces deux cns ne présentent pas le même niveau d'adéquation avec les 4 cns de E ($Adq(E,P_a) = 10$ et $Adq(E,P_b) = 12$, elles possèdent la même valeur pour le critère QKM ($QKM(P_a) = QKM(P_b) = 4$).

6.3.3.2 Propriétés de la Méthode

Cette méthode hérite ces propriétés de la méthode de cns K-Modes (voir chapitre 3) :

- sa complexité algorithmique linéaire selon le nombre d'objets n ;
- la scalabilité de la méthode semble bonne ;
- nécessite de fixer a priori le nombre final de classes pour la cns agrégée.

6.3.4 Evaluations Expérimentales

6.3.4.1 Evaluations, Comparaisons et Discussions Préliminaires

Afin de proposer une comparaison des méthodes proposées (utilisation de KEROUAC ou des K-Modes) pour l'agrégation de cns avec les méthodes in-

troduites par Strehl et Gosh nous utiliserons la méthode d'évaluation que ces derniers avaient employée afin d'effectuer des comparaisons entre leurs 3 méthodes (CSPA, MCLA, HGPA). Cette méthodologie de comparaison procède en deux phases : la comparaison des complexités algorithmiques théoriques des méthodes, et l'analyse des résultats d'une expérience spécifique.

- **Comparaison des complexités algorithmiques théoriques :** Les méthodes HGPA et K-Modes constituent les méthodes les plus rapides, puis suit la méthode MCLA , puis KEROUAC et enfin la méthode CSPA qui devient quant à elle inutilisable si le nombre d'objets à traiter est trop important.

- **Expérience de Comparaison :**

  Description de l'expérience :
  Nous reprenons l'expérience réalisée par Strehl et Gosh :
  On partitionne un ensemble de $n = 500$ objets en $k = 10$ classes de manière aléatoire afin d'obtenir une cns initiale κ[3]. On réplique cette cns $r = 10$ fois. Ces cns sont notées λ_i ($i = 1..r$), on note E l'ensemble de ces r cns : $E = \{\lambda_i, i = 1..r\}$.
  Puis, pour différents niveaux de bruits, et pour chaque cns λ_i une fraction des objets est aléatoirement déplacée de leur classe initiale vers une autre classe (le choix de la classe de destination est géré aléatoirement selon une distribution uniforme selon les k classes).
  On utilise ensuite, pour chaque niveau de bruit, les différentes méthodes d'agrégation de cns afin d'agréger les r cns différentes en une unique cns notée Λ.
  Les cns résultant de l'agrégation sont alors évaluées selon plusieurs points :

  1. Evaluation de l'information mutuelle symétrique normalisée moyenne entre chacune des cns résultant d'un processus d'agrégation et l'ensemble des r cns dont elle constitue l'agrégation (évaluation de $\varphi^{ANSMI}(\Lambda, E)$). (Il s'agit en fait d'évaluer la fonction à maximiser sous-jacente aux méthodes de Strehl et Gosh.)(figure 6.3)
  2. Evaluation de l'information mutuelle symétrique normalisée entre chacune des cns résultant d'un processus d'agrégation et la cns κ (évaluation de $\varphi^{NSMI}(\kappa, \Lambda)$. (figure 6.4)
  3. Evaluation de l'information mutuelle asymétrique normalisée moyenne entre chacune des cns résultant d'un processus d'agrégation et l'ensemble des r cns dont elle constitue l'agrégation (évaluation de $\varphi^{ANAMI}(\Lambda, E)$. (figure 6.3)
  4. Evaluation de l'information mutuelle asymétrique normalisée entre chacune des cns résultant d'un processus d'agrégation et la cns κ (évaluation de $\varphi^{NAMI}(\kappa, \Lambda)$. (figure 6.4)

3. La classe de chacun des objets est choisie aléatoirement selon une distribution uniforme entre les k classes. Ainsi, les k classes comprennent approximativement le même nombre d'objets.

5. Evaluation de l'adéquation entre chacune des cns résultant d'un processus d'agrégation et l'ensemble des r cns dont elle constitue l'agrégation (évaluation de $Adq(\Lambda,E)$. (figure 6.5)
6. Evaluation de l'adéquation entre chacune des cns résultant d'un processus d'agrégation et la cns κ (évaluation de $Adq(\kappa,\Lambda)$). (figure 6.5)

Lors de leur expérimentation Strehl et Gosh se sont contentés d'une analyse selon les deux premiers points car, d'une part les méthodes d'agrégation qu'ils ont proposées nécessitent de fixer a priori le nombre de classes de la cns résultant de l'agrégation (ainsi l'utilisation des points 3 et 4 pour l'analyse est ici inutile puisque la totalité des cns à comparer possèdent le même nombre de classes), et d'autre part car, pour eux, l'objectif est d'optimiser le critère $\varphi^{ANSMI}(\Lambda,E)$ et non le critère $Adq(\Lambda,E)$ (même s'il existe une relation unissant ces critères). Nous utiliserons quant à nous l'ensemble de ces 6 points pour procéder à l'analyse des résultats de cette expérience. Notons également que concernant les résultats des méthodes HGPA, MCLA et CSPA nous reportons les résultats obtenus par Strehl et Gosh donnés dans [Str02] (ainsi, seuls les deux premiers points sont évalués pour ces 3 méthodes). De plus, afin de permettre une meilleure analyse, Strehl et Gosh avaient introduit également dans leur expérience les résultats associés :

- à une méthode d'agrégation aléatoire (notée *random labels*),
- à une hypothétique méthode d'agrégation qui fournirait toujours comme résultat la cns κ (cette méthode est notée *original labels*).

Ces deux dernières méthodes jouent en définitive le rôle de témoin.

Analyse des résultats de l'expérience :
Les figures 6.3, 6.4 et 6.5 donnent les résultats de cette expérience. On observe que :

- plus le bruit augmente, moins les r cns de E partagent d'informations et donc la valeur maximale que l'on peut obtenir pour $\varphi^{ANSMI}(\Lambda,E)$ diminue quelle que soit la méthode d'agrégation employée :
  - HGPA possède les plus mauvaise performance pour cette expérience.
  - L'ensemble des méthodes restantes (MCLA, CSPA, KEROUAC, K-Modes) montrent un niveau de performance sensiblement équivalent pour des fractions de bruits relativement faible (inférieures à 40%).
  - Pour des niveaux de bruits intermédiaire à fort (supérieurs à 40%) la méthode KEROUAC surpasse les méthodes MCLA, CSPA, K-Modes qui se comportent de manière identique. Ces comportements s'expliquent, selon nous, par le fait que le nombre de classes soit fixe pour les méthodes MCLA, CSPA et K-Modes (et ce nombre de classes vaut ici 10) alors que la méthode Kerouac détermine automatiquement le nombre de classes.

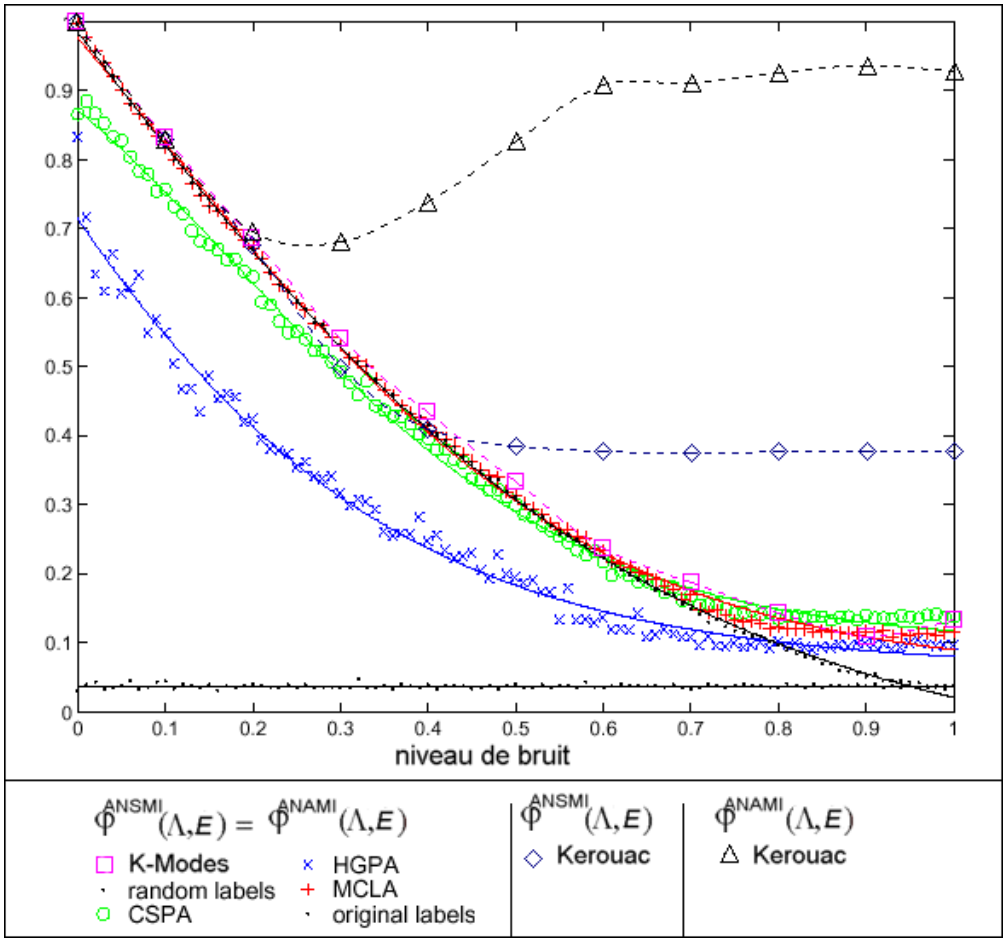


FIG. 6.3 –: *Evaluation de l'information mutuelle symétrique (resp. asymétrique) normalisée moyenne entre chacune des cns résultant d'un processus d'agrégation et l'ensemble des r cns dont elle constitue l'agrégation (évaluation de $\varphi^{ANSMI}(\Lambda,E)$ (resp. $\varphi^{ANAMI}(\Lambda,E)$))*

Ainsi, lorsque le bruit est modéré à fort, la cns κ ne correspond plus beaucoup aux cns λ_i et fixer le nombre de classes à 10 devient une contrainte handicapante tandis que KEROUAC de par sa capacité à déterminer automatiquement le nombre de classes proposera un meilleur résultat puisque la structure de la cns résultant de l'agrégation pourra être mieux adaptée. (On peut corréler partiellement cette analyse avec le fait que pour des niveaux élevés de bruits la cns κ (représentée par la méthode virtuelle *original labels*) présente la valeur la plus faible pour $\varphi^{ANSMI}(\Lambda,E)$. (Cette valeur est sensiblement égale à celle obtenue pour une cns en 10 classes déterminée aléatoirement, voir *random labels*). En effet, l'explication de ce phénomène est que les cns λ_i ont subies, pour ces niveaux de bruits, un

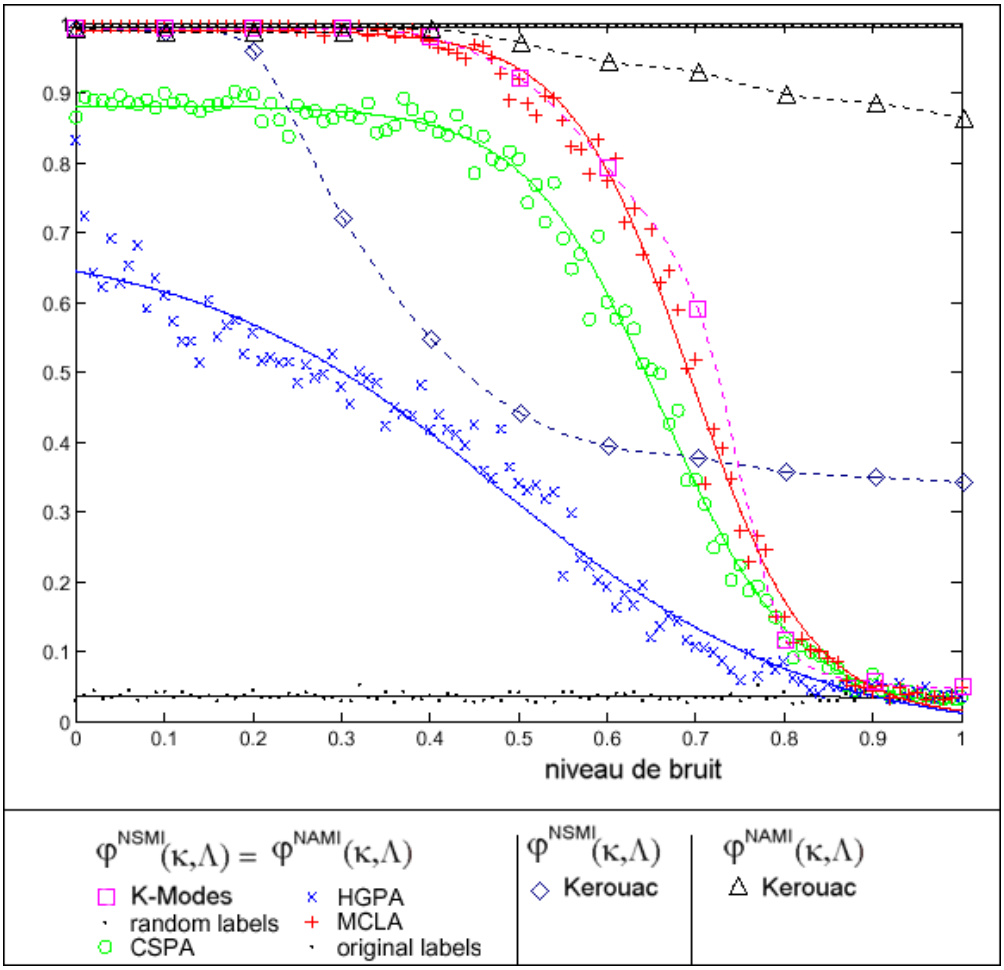


FIG. 6.4 –: *Evaluation de l'information mutuelle symétrique (resp. asymétrique) normalisée entre chacune des cns résultant d'un processus d'agrégation et la cns κ (évaluation de $\varphi^{NSMI}(\kappa,\Lambda)$ (resp. $\varphi^{NAMI}(\kappa,\Lambda)$))*

ensemble de modifications tel qu'elles ne présentent alors presque aucun lien avec la cns κ.)

- Il apparaît que le niveau de bruit ne doit pas dépasser 50% et que dans ces conditions 3 méthodes semblent supérieures et à peu près également valide : MCLA, K-Modes et KEROUAC.
- Concernant la capacité des méthodes à "retrouver" (par agrégation) la cns κ en présence de bruit, plus le bruit augmente, moins les r cns de Λ partagent d'informations avec la cns κ et donc la valeur maximale que l'on peut obtenir pour $\varphi^{ANSMI}(\kappa,\Lambda)$ diminue quelle que soit la méthode d'agrégation employée :
  - HGPA possède les plus mauvaise performances pour cette expérience.

- L'ensemble des méthodes restantes (MCLA, KEROUAC, K-Modes) montrent un niveau de performance sensiblement équivalent pour des fractions de bruits relativement faible (inférieur à 20%). Dans ces conditions, la méthode CSPA est, elle, légèrement en retrait.
- Pour des niveaux de bruits intermédiaire à assez fort (approximativement entre 30% et 70%) les méthodes MCLA, CSPA, K-Modes se comportent de manière identique (avec un léger retrait pour CSPA) et surpassent KEROUAC. Par contre, pour de forts niveaux de bruit (supérieurs à 70%), la tendance s'inverse et KEROUAC surpasse ces méthodes. Ces comportement s'expliquent par :
  - Le fait que le nombre de classes soit fixe pour les méthodes MCLA, CSPA et K-Modes (et ce nombre de classes vaut ici 10) alors que la méthode Kerouac détermine automatiquement le nombre de classes. Ainsi, lorsque le bruit est modéré, fixer le nombre de classes à 10 (i.e. au nombre de classes de la cns κ) revient à intégrer une connaissance importante au processus d'agrégation et donc "facilite" le processus d'agrégation pour les méthodes MCLA, CSPA et K-Modes. Par contre, lorsque le bruit est plus fort, la cns κ ne correspond plus beaucoup aux cns λ_i et fixer le nombre de classes à 10 devient une contrainte handicapante tandis que KEROUAC de part sa capacité à déterminer automatiquement le nombre de classes proposera un meilleur résultat. (On peut corréler partiellement cette analyse avec le fait que pour des niveaux élevés de bruits la cns κ (représentée par la méthode virtuelle *original labels*) présente la valeur la plus faible pour $\varphi^{NSMI}(\kappa,\Lambda)$).
  - Concernant les niveaux de bruit modérés à assez fort, on peut également donner comme explication le fait que la mesure $\varphi^{NSMI}(\kappa,\Lambda)$ est biaisée en faveur de cns Λ possédant un nombre faible de classes. Or les cns résultant de KEROUAC possèdent un nombre de classes le plus souvent largement supérieur à 10 (voir figure 6.5). Ainsi, utiliser cette mesure pour évaluer la qualité des cns résultants de l'agrégation tend à favoriser les méthodes MCLA, CSPA et K-Modes. Notons également que, si on avait par contre utilisé la mesure $\varphi^{NAMI}(\kappa,\Lambda)$, qui est biaisée en faveur de cns possédant un nombre de classes élevé la tendance aurait été inversée (voir figure 6.4).
- On peut observer (figure 6.4, figure 6.5) que MCLA, K-Modes, KEROUAC proposent, pour des niveaux de bruit inférieur à 40%, des cns dont les classes sont quasiment pures du point de vue de la classe d'appartenance dans κ des objets qu'elles contiennent. (Notons que cette plage s'étend jusqu'à 50% de bruit pour KEROUAC, cela s'expliquant en partie par sa capacité à déterminer automatiquement le nombre de classes de

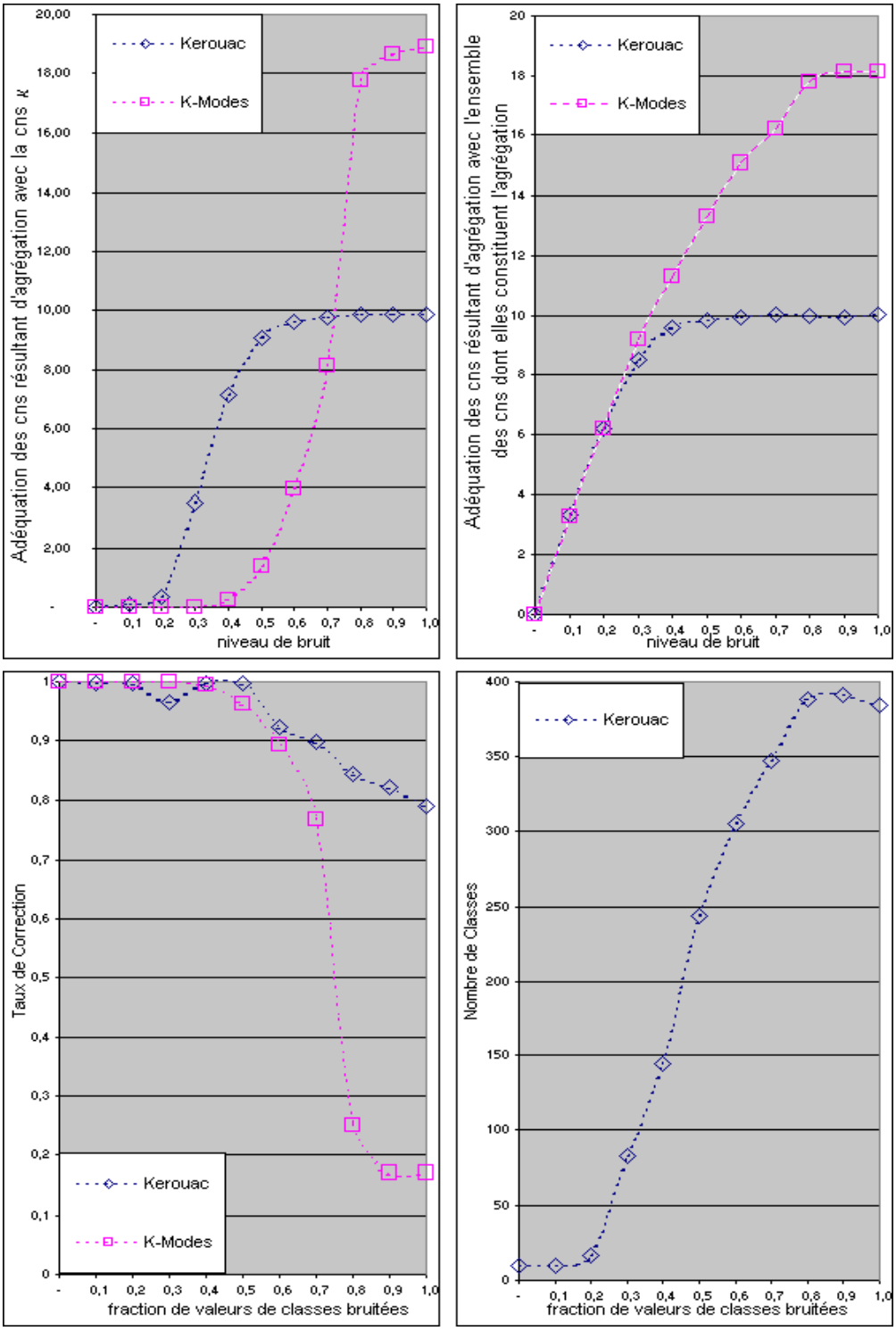


FIG. 6.5 –: *Evaluation de l'adéquation entre chacune des cns résultant d'un processus d'agrégation et l'ensemble des r cns dont elle constitue l'agrégation (évaluation de $Adq(\Lambda,E)$ et $Adq(\kappa,\Lambda)$)*

la cns résultant de l'agrégation.)

Cette expériences indique que la fonction $\varphi^{ANSMI}(\Lambda,E)$ proposée comme fonction à optimiser par Strehl et Gosh est réellement appropriée pour la résolution du problème d'agrégation car, en pratique, la valeur $\varphi^{NSMI}(\kappa,\Lambda)$ est non disponible, et on observe un fort lien entre $\varphi^{NSMI}(\kappa,\Lambda)$ $\varphi^{ANSMI}(\Lambda,E)$. On peut cependant noter les problèmes de biais liés au nombre de classes des cns à agréger et celle résultant des agrégation. La fonction que nous proposons $Adq(\Lambda,E)$ est elle aussi tout aussi adaptée (une observation simultanée des figures 6.3, 6.4, 6.5 le montre) et semble de plus ne pas exhiber de biais en relation avec le nombre de classes des cns.

Les complexités théoriques ainsi que l'expérience menée montre tout l'intérêt que revêt l'introduction des méthodes KEROUAC et K-Modes pour l'agrégation de cns :

- La méthode K-Modes semble fournir des résultats d'excellente qualité (comparable aux meilleurs résultats) tout en possédant la complexité algorithmique la plus faible;
- La méthode KEROUAC semble, elle aussi, fournir des résultats de bonne qualité, et possède une complexité algorithmique tout à fait acceptable. De plus elle ne nécessite pas de fixer a priori le nombre de classes de la cns résultant de l'agrégation ce qui lui permet de s'adapter aux situations réelles pour lesquelles on ne connaît que rarement le nombre de classes que doit comporter cette cns.

6.3.4.2 Evaluations, Comparaisons et Discussions Complémentaires

Nous procédons maintenant à un ensemble d'évaluations expérimentales supplémentaires visant, d'une part, à illustrer l'intérêt de l'utilisation des méthodes d'agrégation de cns basées sur les méthodes KEROUAC et K-Modes dans le cadre du scénario de distribution des données **DDV**, et d'autre part, à évaluer les capacités de ces deux méthodes à agréger "correctement" un ensemble de cns (nous nous appuierons pour cela sur des expérimentations dans le cadre du scénario **DDOV**) ainsi que la capacité de la méthode utilisant KEROUAC à agréger des cns ayant des nombre de classes différents.

Les jeux de données utilisés correspondent aux jeux de données utilisés par Strehl et Gosh afin de conserver une certaine uniformité avec leurs travaux et d'autoriser des comparaisons plus aisées avec leurs méthodes. Ces jeux de données sont :

- le jeu de données PenDigits de l'UCI [MM96] ; ce jeu de données correspond à la description par 16 variables quantitatives de 7494 chiffres manuscrits. Les chiffres sont classés en 10 classes (chaque classe correspondant à un chiffre) qui comprennent sensiblement le même nombre

d'objets (voir page 217 pour de plus amples informations sur ce jeu de données).

– un jeu synthétique 8D5K (ce jeu de données est composé de 1000 objets correspondant à 5 distributions Gaussienne multivariée dans un espace à 8 dimensions. Chacune des distributions est représentée par 200 objets). Chacune de ces distributions possède la même variance (0.1) mais elles possèdent toutes des moyennes différentes) (ce jeu de données est disponible pour téléchargement sur le site *http://strehl.com* et sa composition est plus largement commentée dans [Str02]).

Données distribuées sur les variables (DDV) Pour ce scénario de distribution des données, les expérimentations menées visent à illustrer comment l'agrégation de cns obtenues par application d'algorithmes de cns sur des "vues partielles" des données permet d'obtenir une cns de meilleure qualité.

Nous reprenons ici encore l'expérimentations menée par Strehl et Gosh sur le jeu de données 8D5K (ce jeu de données ayant été utilisé car les résultats de l'expérience se prêtent aisément à l'illustration).

Ainsi, le scénario DDV est simulé : plusieurs processus de cns sont lancés, ces processus de cns sont tels que chacun des processus de cns a accès à la totalité des objets du jeu de données et seulement à un nombre limité des variables. Le résultat de chacun des processus de cns (i.e. la composition de chacune des classes) est alors transmis à une des méthode d'agrégation.

Pour cette expérience 5 processus de cns différents sont exécutés (par l'intermédiaire de la méthode K-Means paramétrée de manière à ce qu'elle fournisse une cns en 5 classes) ; chacun de ces processus ayant uniquement accès à deux des huit variables du jeu de données. Puis les méthodes d'agrégation utilisant les méthodes KEROUAC et K-Modes sont employées pour réaliser l'agrégation.

Préalablement à l'analyse des résultats de l'expérience sur le jeu de données 8D5K, il est important de se remémorer que (d'après le processus qui a permis de bâtir ce jeu de données synthétique) on peut associer à chacun des objets une des 5 distributions gaussiennes constituant le jeu de données. Ainsi on peut considérer qu'il existe une classification "naturelle" en 5 classes des objets du jeu de données selon la distribution à laquelle ils sont associés. Notons que ces 5 classes sont linéairement séparables dans l'espace à 8 dimensions et que cette classification en 5 classes sera appelée par la suite classification de référence.

La figure 6.6 présente (en haut) les objets de ce jeu de données projeté dans l'espace à deux dimensions constituée par les 2 axes principaux de l'analyse en composantes principales (ACP). On observe que les 5 classes de la classification de référence sont relativement bien séparées dans cet espace. Sur la même figure on peut également observer les résultats de chacun des 5 processus de cns. Chacune des cns issues de ces processus est représentée à la fois dans l'espace bi-dimensionnel constitué par les axes des composantes principales de

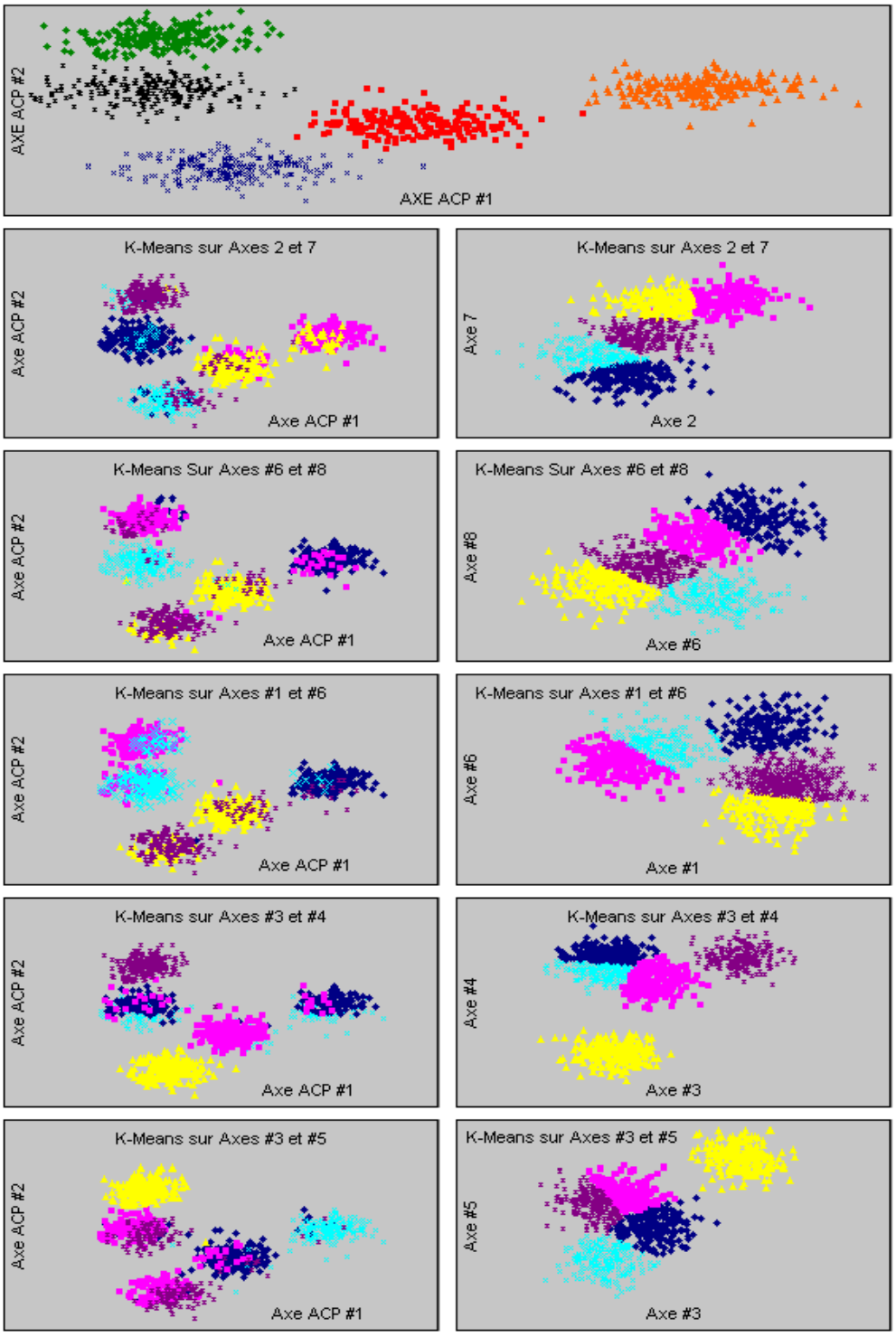


FIG. 6.6 –: *Scénario **DDV** : Expérience sur le jeu de données 8D5K*

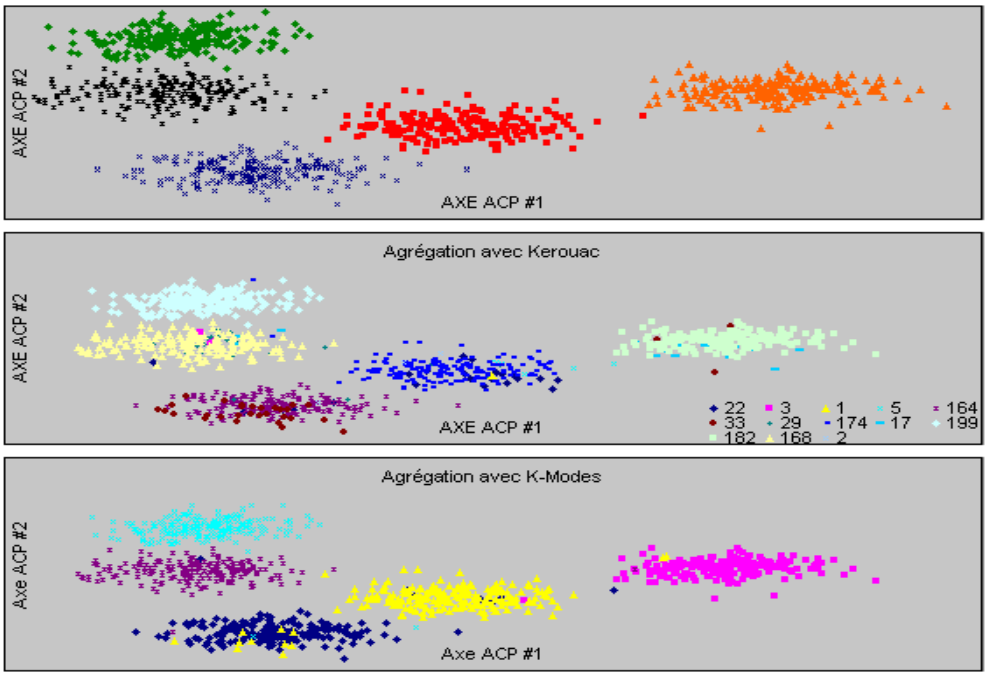


FIG. 6.7 –: *Scénario **DDV** : Expérience sur le jeu de données 8D5K*

l'ACP (à gauche) et dans l'espace bi-dimensionnel constitué par les deux variables auxquelles le processus de cns correspondant a eu accès (à droite). On peut ainsi observer qu'aucune de ces 5 cns ne correspond véritablement à la cns de référence.

Par contre, sur la figure 6.7, on peut observer que les cns résultant de l'agrégation de ces 5 cns soit par l'utilisation de KEROUAC, soit par l'utilisation de la méthode K-Modes correspondent, elles, relativement fidèlement à la classification de référence.

Ainsi, la cns obtenue par l'intermédiaire des K-Modes est très fidèle à la classification de référence puisque seulement une vingtaine d'objets (sur un total de mille) sont classifiés différemment. Pour la cns obtenue par KEROUAC la correspondance est, elle aussi, bonne, d'autant plus que le nombre de classes de la cns résultant de l'agrégation n'a pas été fixé a priori. On observe en fait qu'il existe 5 classes principales correspondant parfaitement aux cinq classes de la classification de référence (le nombre d'objets par classes est donné en bas à droite de la figure 6.7).

Cette expérience illustre l'intérêt de l'agrégation de cns obtenues à partir de vues partielles sur les données : la cns obtenue par agrégation présente une qualité augmentée. Ce type d'agrégation doit être particulièrement intéressant dans le cas de données hétérogènes ne pouvant être traitées par une méthode unique, cela permet également de s'adapter "naturellement" au cas où les données sont distribuées physiquement, et permet d'envisager le traitement de

jeux de données présentant un très grand nombre de variables.

Données distribuées sur les objets et les variables (scénario DDOV) L'objectif des expérimentations est d'évaluer la qualité des cns issues de l'agrégation dans le cadre des scénarios DDV, DDO et DDOV. Il s'agit donc d'évaluer la qualité des méthodes que nous proposons pour l'agrégation d'un ensemble $E = \{P_1,...,P_z\}$ de cns sachant que chacune des cns de E est bâtie en accédant uniquement à un sous ensemble de l'ensemble des variables et à un sous-ensemble de l'ensemble des objets et qu'aucun accès à ces sous-ensembles n'est autorisé lors de l'agrégation (on ne dispose que de la composition des classes de chacune de cns P_i).

Description des expériences :

Les expériences suivantes ont été menées sur le jeu de données PenDigit : plusieurs séries de 30 processus de cns ont été lancées, chacune de ces séries correspondant à un niveau d'échantillonnage pour les variables et à un niveau d'échantillonnage pour les objets. Les niveaux d'échantillonnage suivants ont été employés pour les variables : 100%, 75%, 50%, 25%, 12,5%[4], quant aux niveaux d'échantillonnage employés pour les objets ils étaient les suivants : 100%, 75%, 50%, 25%, 10%, 5%[5].

Ainsi, chaque série de 30 processus de cns étant caractérisée par :

- un niveau d'échantillonnage parmi 5 possibles pour les variables,
- un niveau d'échantillonnage parmi 6 possibles pour les objets,

$6 \times 5 = 30$ séries ont donc été lancées. Chaque série est notée par la suite $SerieX,Y$. X fait référence au niveau d'échantillonnage pour les variables, Y au niveau d'échantillonnage pour les objets (par exemple, la série notée $Serie50,10$ correspond au niveau d'échantillonnage 50% pour les variables et 10% pour les objets).

Pour chaque séries $SerieX,Y$; 30 processus de cns ont été réalisés, chaque processus de cns ayant accès à un échantillon "quasi-aléatoire" de $X\%$ des variables du jeu de données PenDigit et à un échantillon "quasi-aléatoire" de $Y\%$ des objets de ce jeu de données. Les schémas de tirage sont dit "quasi-aléatoires" car pour chaque série de cns les échantillons de variables ont été obtenus d'une manière telle que :

- chaque variable est présente dans au moins un des échantillons, et au plus une fois dans un échantillon donné,
- chaque objet est présent dans au moins un échantillon et au plus une fois dans un échantillon donné.

L'ensemble des processus de cns ont été réalisés par l'intermédiaire de la méthode K-Means paramétrée de manière à obtenir des cns en 10 classes.

4. soit respectivement 16, 12, 8, 4 et 2 variables

5. soit respectivement 7194, 5395, 3597, 1798, 719 et 360 objets

Puis, pour chaque série $SerieX,Y$ (i.e. pour chaque couple de niveau d'échantillonnage), les méthodes d'agrégation de cns utilisant les K-Modes ou Kerouac ont été utilisées afin d'agréger l'ensemble des 30 cns composant la série. La cns ainsi obtenue pour la série $SerieX,Y$ est notée par la suite $P_{x,y}^{K-Modes}$ si l'agrégation a été réalisée en utilisant la méthode K-Modes, ou $P_{x,y}^{Kerouac}$ si l'agrégation a été réalisée en utilisant la méthode KEROUAC.

Enfin, nous avons étudié l'ensemble des cns de la série $Serie100,100$ (i.e. la série de cns obtenue par application des K-Means sur l'ensemble des objets et des variables) afin de déterminer la meilleure (resp. la moins bonne) de ces cns au sens du critère à optimiser sous-jacent à cette méthode. Cette cns est notée P_{ref} (resp. P_{-}). La cns P_{ref} constitue la cns de référence pour le jeu de données PenDigit et la méthode K-Means.

Analyse des expériences

Nous considérons un ensemble de 4 indices Q_1, Q_2, Q_3, Q_4 afin d'évaluer la qualité des différentes cns $P_{x,y}$ obtenues par agrégation :

- l'indice Q_1 est défini comme le rapport suivant :
  $Q_1(P_1,P_2) = \frac{Adq(P_2,P_{ref})}{Adq(P_1,P_{ref})}$.
  Cet indice permet donc de comparer l'adéquation entre la cns P_1 et la cns de référence P_{ref} avec l'adéquation entre la cns P_2 et la cns de référence. Notons que plus la valeur de $Adq(P_i,P_{ref})$ est faible, plus l'adéquation entre ces deux cns est forte. Ainsi, une valeur de l'indice $Q_1(P_1,P_2)$ supérieure (resp. inférieure) à 1 signifie que la cns P_1 (resp. P_2) présente la meilleure adéquation avec la cns de référence. A partir de cet indice nous définissons 2 indices permettant l'évaluation de la qualité d'une cns P_1 :
  - $Q_1^{-}(P_1) = Q_1(P_1,P_{-})$.
  - $Q_1^{moy}(P_1) = \frac{1}{card(Serie100,100)} \sum_{P \in Serie100,100} Q_1(P_1,P)$.

- l'indice Q_2 est défini comme le rapport suivant :
  $Q_2(P_1,P_2) = \frac{\phi^{NSMI}(P_1,P_{ref})}{\phi^{NSMI}(P_2,P_{ref})}$.
  Cet indice permet donc de comparer l'adéquation entre la cns P_1 et la cns de référence P_{ref} avec l'adéquation entre la cns P_2 et la cns de référence. Notons que plus la valeur de $\phi^{NSMI}(P_i,P_{ref})$ est forte, plus l'adéquation entre ces deux cns est forte. Ainsi, une valeur de l'indice $Q_2(P_1,P_2)$ supérieure (resp. inférieure) à 1 signifie que la cns P_1 (resp. P_2) présente la meilleure adéquation avec la cns de référence. A partir de cet indice nous définissons 2 indices permettant l'évaluation de la qualité d'une cns P_1 :
  - $Q_2^{-}(P_1) = Q_2(P_1,P_{-})$.
  - $Q_2^{moy}(P_1) = \frac{1}{card(Serie100,100)} \sum_{P \in Serie100,100} Q_2(P_1,P)$.

– l'indice Q_3 permet de comparer la qualité d'une cns P_1 et celle d'une cns P_2 par l'intermédiaire de la valeur du taux de correction de ces cns par rapport aux 10 groupes "naturels" du jeu de données. Le taux de correction d'une partition P_i est noté $TC(P_i)$; plus la valeur de ce critère est fort, meilleure est la cns. L'indice Q_3 est défini ici comme le rapport : $Q_3(P_1,P_2) = \frac{TC(P_1)}{TC(P_2)}$.
Ainsi, une valeur de l'indice $Q_3(P_1,P_2)$ supérieure (resp. inférieure) à 1 signifie que la cns P_1 présente une meilleure (resp. moins bonne) qualité que la cns P_2. A partir de cet indice nous définissons 2 indices permettant l'évaluation de la qualité d'une cns P_1 :
  – $Q_3^-(P_1) = Q_3(P_1,P_-)$.
  – $Q_3^{moy}(P_1) = \frac{1}{card(Serie100,100)} \sum_{P \in Serie100,100} Q_3(P_1,P)$.

– l'indice Q_4 permet de comparer la qualité d'une cns P_1 et celle d'une cns P_2 par l'intermédiaire de la valeur du critère $QKMeans$ (critère sous-jacent à la méthode des K-Means, plus la valeur de $QKMeans$ est faible, meilleure est la cns). L'indice Q_4 est défini ici comme le rapport : $Q_4(P_1,P_2) = \frac{QKmeans(P_2)}{QKMeans(P_1)}$.
Ainsi, une valeur de l'indice $Q_4(P_1,P_2)$ supérieure (resp. inférieure) à 1 signifie que la cns P_1 présente selon le critère $QKmeans$ une meilleure (resp. moins bonne) qualité que la cns P_2. (Remarque : le critère $QKmeans$ étant sensible au nombre de classes des cns, l'utilisation du critère Q_4 est essentiellement envisageable pour la comparaison de cns possédant le même nombre de classes.)
A partir de cet indice nous définissons 2 indices permettant l'évaluation de la qualité d'une cns P_1 :
  – $Q_4^-(P_1) = Q_4(P_1,P_-)$.
  – $Q_4^{moy}(P_1) = \frac{1}{card(Serie100,100)} \sum_{P \in Serie100,100} Q_4(P_1,P)$.

Les indices $Q_1^-(P_{x,y})$, $Q_2^-(P_{x,y})$, $Q_3^-(P_{x,y})$, $Q_4^-(P_{x,y})$ permettent ainsi de comparer la qualité d'une cns $P_{x,y}$ obtenue par agrégation à la "moins bonne" des cns obtenues par application de la méthode K-Means sur l'intégralité du jeu de données.

Les indices $Q_1^{moy}(P_{x,y})$, $Q_2^{moy}(P_{x,y})$, $Q_3^{moy}(P_{x,y})$, $Q_4^{moy}(P_{x,y})$ permettent quant à eux une comparaison "en moyenne" de la qualité de l'ensemble des cns de la série Serie100,100 avec une cns $P_{x,y}$ obtenue par agrégation.

Les figures 6.8, 6.9, 6.10, 6.11, illustrent les valeurs de ces différents indices pour les différentes cns provenant d'agrégation (les cns $P_{x,y}^{kerouac}$ et $P_{x,y}^{K-Modes}$).
Notons que pour la figure 6.11 :
  – Seules les différentes valeurs des indices Q_4^- et Q_4^{moy} pour les cns issues d'agrégation par le biais de la méthode des K-Modes sont indiquée. Les

valeurs correspondantes dans le cas de l'utilisation de KEROUAC ne sont pas répertoriées car le nombre de classes des cns obtenues par cette méthode est le plus souvent supérieur à 10 rendant ainsi impossible l'analyse des valeurs de Q_4^- et Q_4^{moy}.
– Sont indiquées les facteurs d'accélération (théoriques) des processus de cns dans le cas où chaque cns de chaque série a été obtenue par utilisation d'un algorithme en $O(n^2)$ et ce soit dans le cas où les différentes cns de chaque série ont été réalisées séquentiellement ou dans le cas où elles ont été réalisées en parallèle de manière simultanée.

Nous proposons ici une analyse relativement succincte des résultats de ces expériences et laissons au lecteur le soin de l'approfondir. Cette analyse est divisée en deux points :

– On procède dans un premier temps à l'étude des valeurs des indices $Q_1^-(P_{x,y})$, $Q_2^-(P_{x,y})$, $Q_3^-(P_{x,y})$, $Q_4^-(P_{x,y})$.
Leur étude permet de comparer les cns obtenues par agrégation avec la "moins bonne" des cns obtenues par application directe de l'algorithme de cns sur l'intégralité du jeu de données. Les deux méthodes semblent exhiber des comportements relativement similaires : la majorité des cns $P_{x,y}$ possède un niveau de qualité supérieur ou proche de la cns P_-. Cependant si les niveaux d'échantillonnage X et Y sont faibles, les cns obtenues par agrégation exhibent alors une qualité dégradée par rapport à celle de P_-. Il apparaît donc que pour une large gamme de couple de niveaux d'échantillonnage les cns $P_{x,y}$ présentent une qualité au moins équivalente à celle de P_- ce qui valide d'une certaine manière les approches proposées pour l'agrégation. (Notons que la distribution des données peut impliquer une accélération des processus de cns (voir figure 6.11).)
Plus précisément, on observe que (comme on pouvait le prévoir), la qualité des cns issues d'agrégation se dégrade au fur et à mesure que les niveaux d'échantillonnage diminuent, et, que la sensibilité à l'échantillonnage sur les variables est plus importante que la sensibilité à l'échantillonnage sur les objets (là encore, ce comportement semble "normal").
Enfin, l'analyse de l'indice Q_4^-, pour des agrégations réalisées par la méthode K-Modes (l'indice Q_4^- constitue, selon nous, le meilleur indicateur pour la comparaison de la qualité des cns), montrent que la très grande majorité des cns issues de l'agrégation possèdent un niveau de qualité au moins égale à 80% de la cns P_- (voir les zones délimitées par un trait rouge sur la figure 6.11).
– Le premier point a consisté en une comparaison entre les cns $P_{x,y}$ et la "pire" des cns obtenue par application de l'algorithme de cns sur l'intégralité du jeu de données P_-. Nous proposons maintenant une analyse visant à comparer la qualité des cns $P_{x,y}$ et la qualité moyenne des cns obtenues application de l'algorithme de cns sur l'intégralité du jeu

de données. Pour cela nous étudions les valeurs des indices $Q_1^{moy}(P_{x,y})$, $Q_2^{moy}(P_{x,y})$, $Q_3^{moy}(P_{x,y})$, $Q_4^{moy}(P_{x,y})$.
Les résultats sont ici similaires à ceux du point précédent :

- forte similitude pour le comportement des deux méthodes ;
- une large gamme (mais, certes plus restreinte) de cns obtenues par agrégation présentent un niveau de qualité supérieur ou proche à la qualité moyenne des cns de la série Serie100,100 ;
- la qualité des cns issues d'agrégation se dégrade au fur et à mesure que les niveaux d'échantillonnage diminuent ;
- la sensibilité à l'échantillonnage sur les variables est plus importante que la sensibilité à l'échantillonnage sur les objets ;
- l'analyse de l'indice Q_4^{moy}, pour des agrégations réalisées par la méthode K-Modes montrent que la très grande majorité des cns issues de l'agrégation possèdent un niveau de qualité au moins égale à 80% de la cns P_{moy} (voir les zones délimitées par un trait rouge sur la figure 6.11).

En définitive, l'analyse des résultats semble valider les deux approches proposées dans le cas où les niveaux d'échantillonnage ne sont pas trop faibles, cette remarque sur le niveau d'échantillonnage s'appliquant surtout pour l'échantillonnage sur les variables. De manière plus détaillée :

- Les résultats sont extrêmement concluants pour le scénario **DDO** (i.e. pour des niveaux d'échantillonnage quelconque pour les objets et un niveau valant 100% pour les variables), en effet, on peut observer sur la figure 6.11 que l'indice Q_4^{moy} n'est inférieur à 0.8 que dans le cas d'un niveau d'échantillonnage sur les objets strictement inférieur à 10%. Cela signifie donc que pour des niveaux d'échantillonnage supérieurs à 10% la qualité de la cns obtenue par agrégation est au moins égale à 80% de la qualité moyenne des cns de la série Serie100,100. Or, si les différentes cns à agréger ont été réalisées de manière parallèle (res. séquentielle) et si l'algorithme de cns utilisé possède une complexité en $O(n^2)$, le facteur d'accélération du processus de cns est alors, par exemple, 100 (resp. 13.33) pour un niveau d'échantillonnage sur les objets valant 10%...
- Les résultats pour le scénario **DDV** sont eux aussi intéressants puisque, lorsque le niveau d'échantillonnage sur les objets est de 100%, l'ensemble des cns obtenues pour des niveaux d'échantillonnage sur les variables supérieurs à 25% présentent également un niveau de qualité au moins égale à 80% de la moyenne des cns de la série Serie100,100 (on considère ici encore l'indice Q_4^{moy}).
- Concernant le scénario **DDOV** bien que les bornes sur les niveaux d'échantillonnage nécessaires à l'obtention de résultats de bonne qualité s'accroissent, nous pensons que là encore les résultats s'avèrent intéressants...

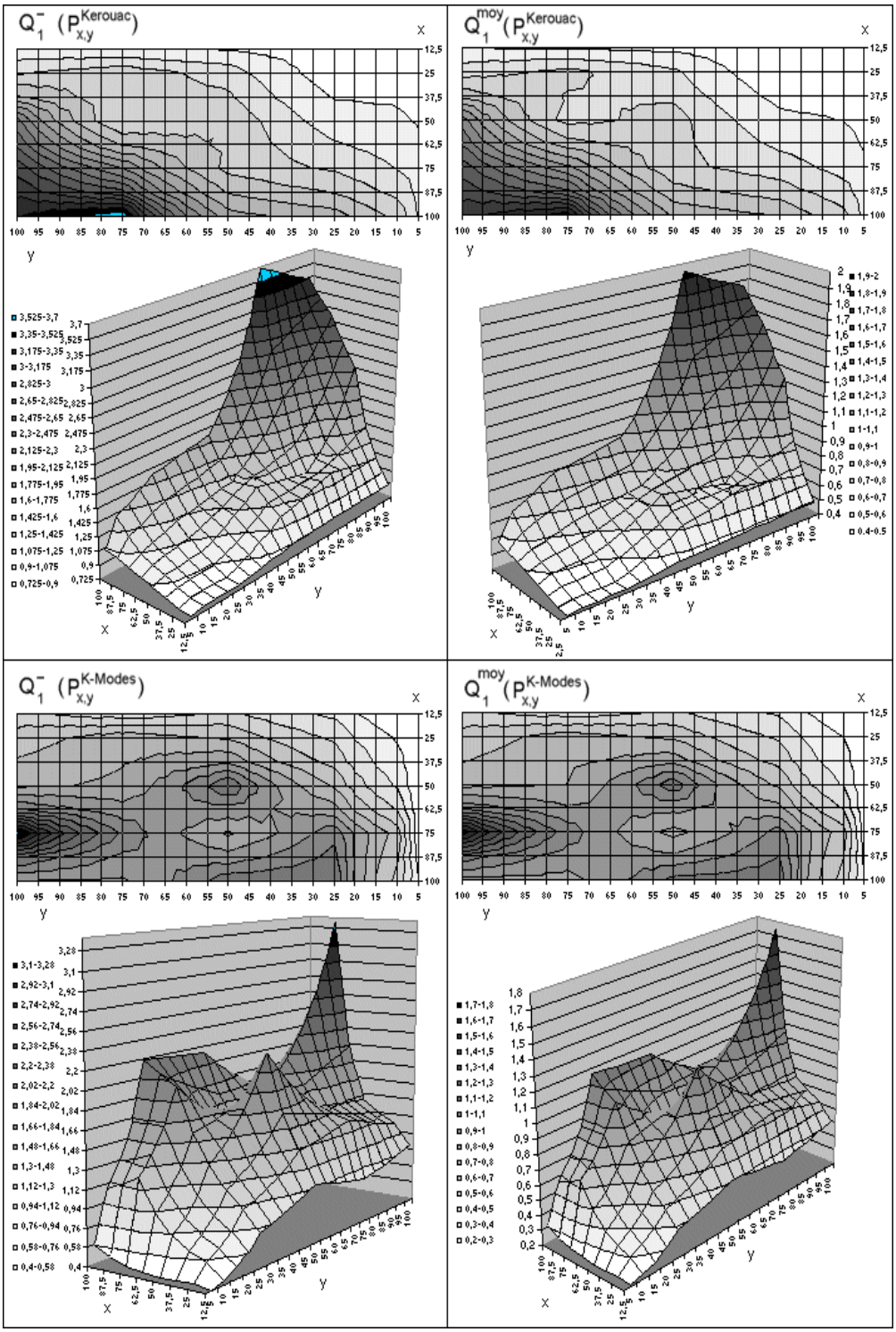


FIG. 6.8 –: *Scénario* ***DDOV****: Evaluation de la qualité des cns issues de l'agrégation, Indice* Q_1

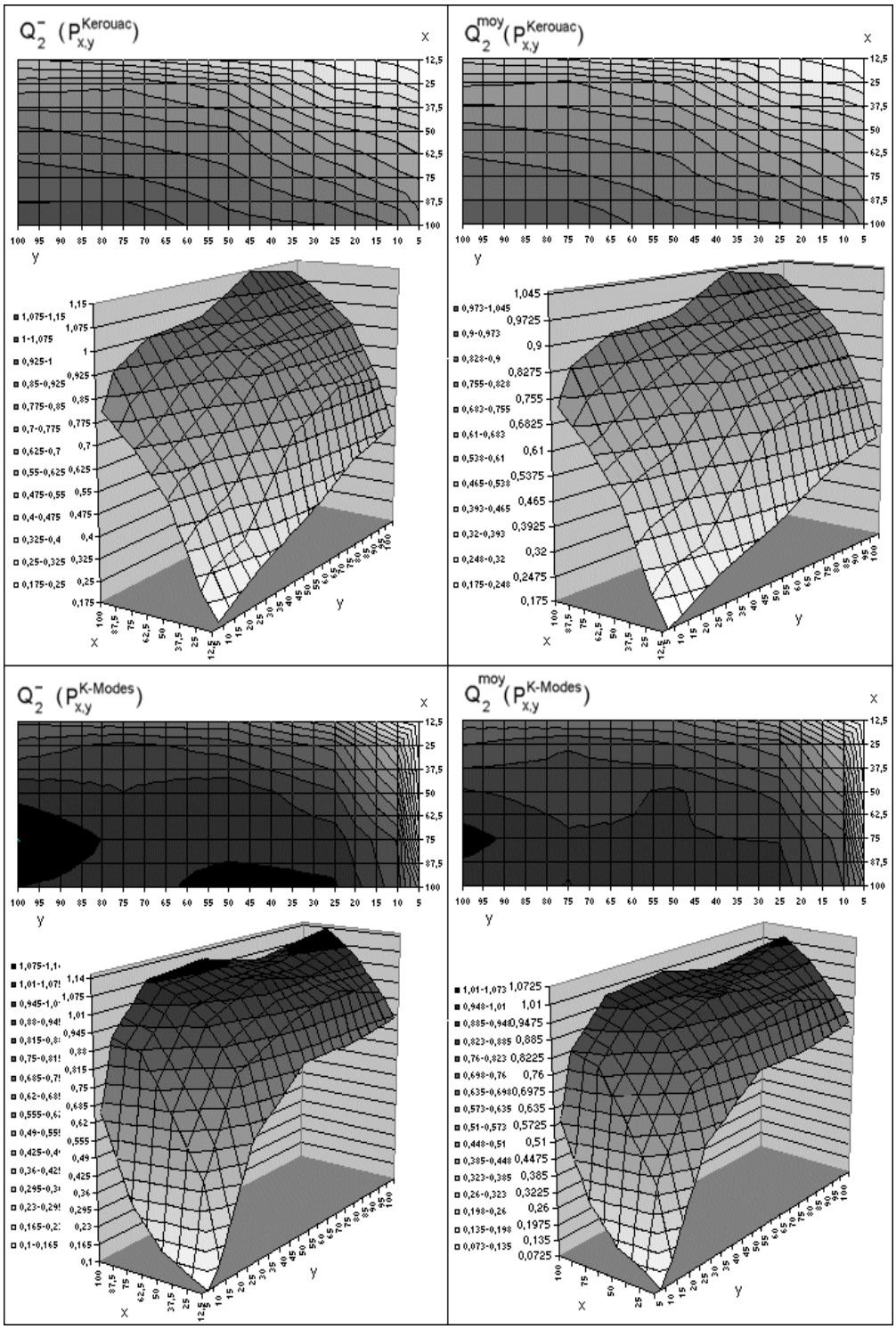


FIG. 6.9 –: *Scénario **DDOV**: Evaluation de la qualité des cns issues de l'agrégation, Indice Q_2*

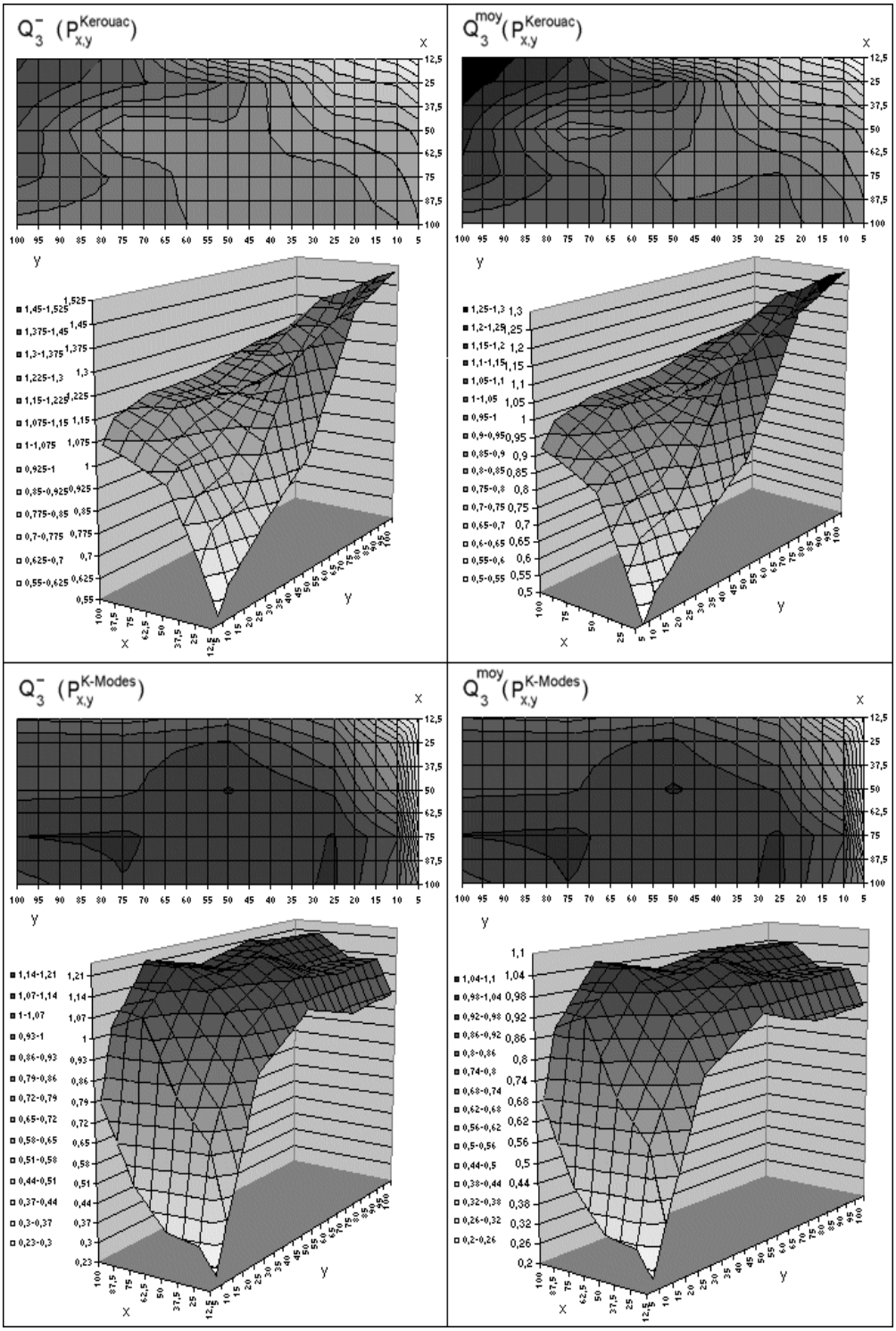


FIG. 6.10 –: *Scénario* ***DDOV****: Evaluation de la qualité des cns issues de l'agrégation, Indice* Q_3

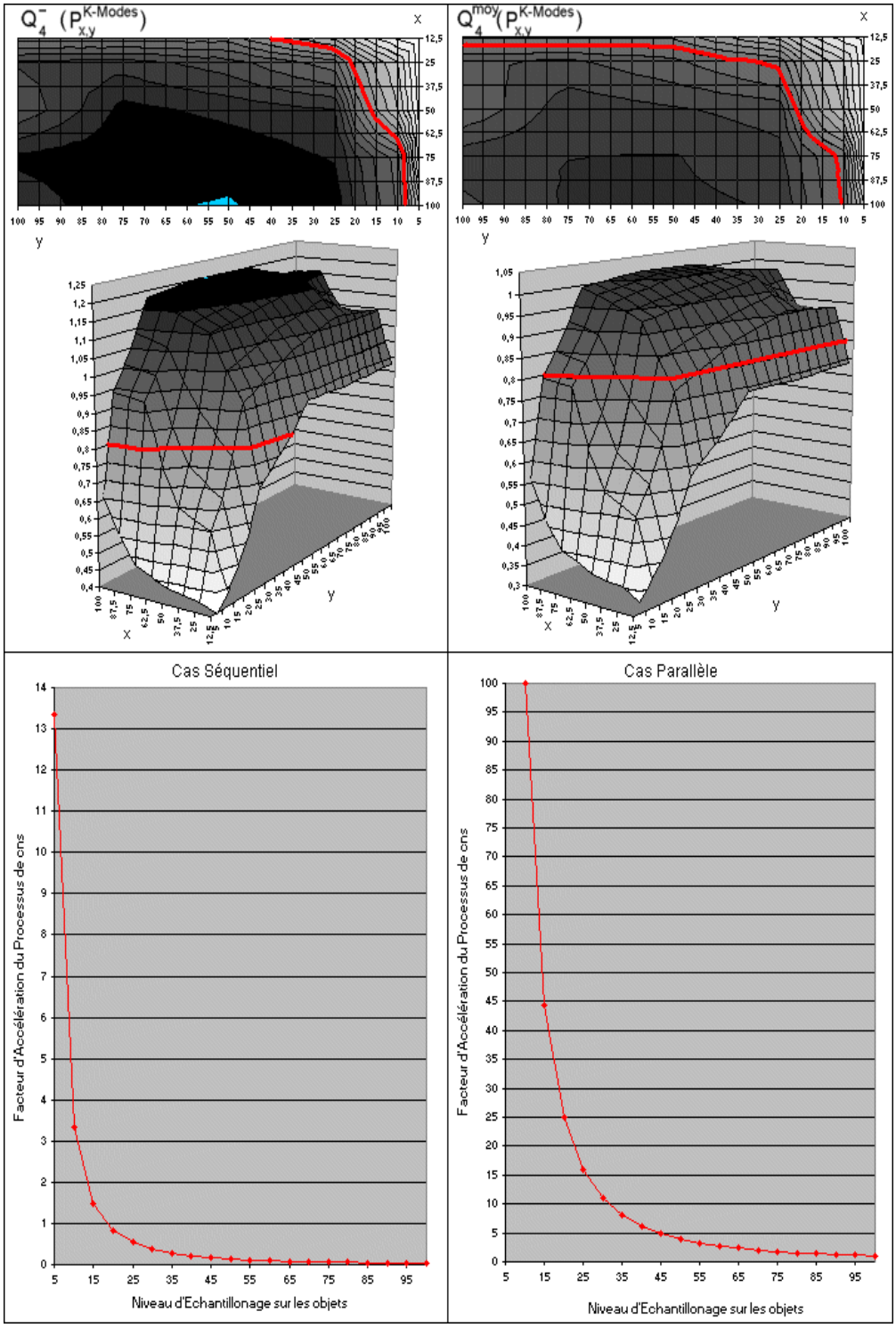


FIG. 6.11 –: *Scénario **DDOV**: Evaluation de la qualité des cns issues de l'agrégation, Indice Q_4; et Facteurs d'accélération du processus de cns*

Expériences supplémentaires

Le même type d'expérimentation a également été mené sur deux autres jeux de données de la collection de l'UCI [MM96] :

- "1984 United States Congressional Voting Records Database" (noté HVOTES), ce jeu de données décrit 16 votes de 435 représentants du congrès des USA (chacun des 435 objets est décrit par 16 variables catégorielles),
- "Mushrooms", ce jeu de données est composé de 8124 objets, chacun de ces objets étant décrit par 22 variables catégorielles.

(Ces jeux de données sont présentés plus en détail plus tard (voir page 217). Cette fois ci, étant donnée la nature catégorielle des données, la méthode KEROUAC est employée pour générer les cns initiales (avec une valeur de 1 pour le facteur de granularité). La cns de référence (P_{ref}) correspond à la cns obtenue par application de la méthode KEROUAC sur l'intégralité du jeu de données. La valeur du critère $NCC^{\star}$ (critère à optimiser sous-jacent à cette méthode) est donc substitué à la valeur du critère $QKMeans$ dans la définition de Q_4, ainsi : $Q_4(P_{ref},P_{x,y}) = Q_4^{-}(P_{ref},P_{x,y}) = Q_4^{moy}(P_{ref},P_{x,y}) = \frac{NCC^{\star}(P_{ref})}{NCC^{\star}(P_{x,y})}$. En effet, comme plusieurs applications de l'algorithme KEROUAC sur l'intégralité du jeu de données mènent au même résultat on a : $P_{Ref} = P_{-}$ et $\forall P \in Serie100{,}100,\ P = P_{Ref}$.

La figure 6.12 (resp. 6.13) décrit les résultats (pour le critère Q_4) de l'expérimentation sur le jeu de données "1984 United States Congressional Voting Records Database" (resp. "Mushrooms"). Les résultats obtenus pour chaque test montrent la grande capacité des deux méthodes à agréger correctement les cns : les valeurs de l'indice Q_4 sont extrêmement proches de 1 (voire supérieures) pour la plupart des couples de niveaux d'échantillonnage et seuls les cas de niveaux d'échantillonnage simultanément très faibles pour les variables et les objets mènent à une relative forte décroissance de la valeur de l'indice Q_4.

6.3.4.3 Comportement de la méthode KEROUAC face à des cns à agréger possédant des nombre de classes très différents

Afin d'évaluer la capacité de la méthode KEROUAC à traiter des cns ayant des nombres de classes largement différents, l'expérience suivante a été menée : 4 séries de 30 cns ont été menées sur le jeu de données "Mushrooms", chaque série correspondant à l'un des 4 niveaux d'échantillonnage suivant pour les objets : 75%, 50%, 25%, 10% et un niveau d'échantillonnage commun de 100% pour les variables. Pour ces expériences nous avons utilisé la méthode KEROUAC pour générer les cns initiales. Le facteur de granularité a été fixé à 3 pour chacun des processus de cns (le facteur de granularité a été fixé à 3 de manière à obtenir des cns possédant des nombres de classes largement différent). Nous avons utilisé la méthode KEROUAC afin de procéder pour chaque

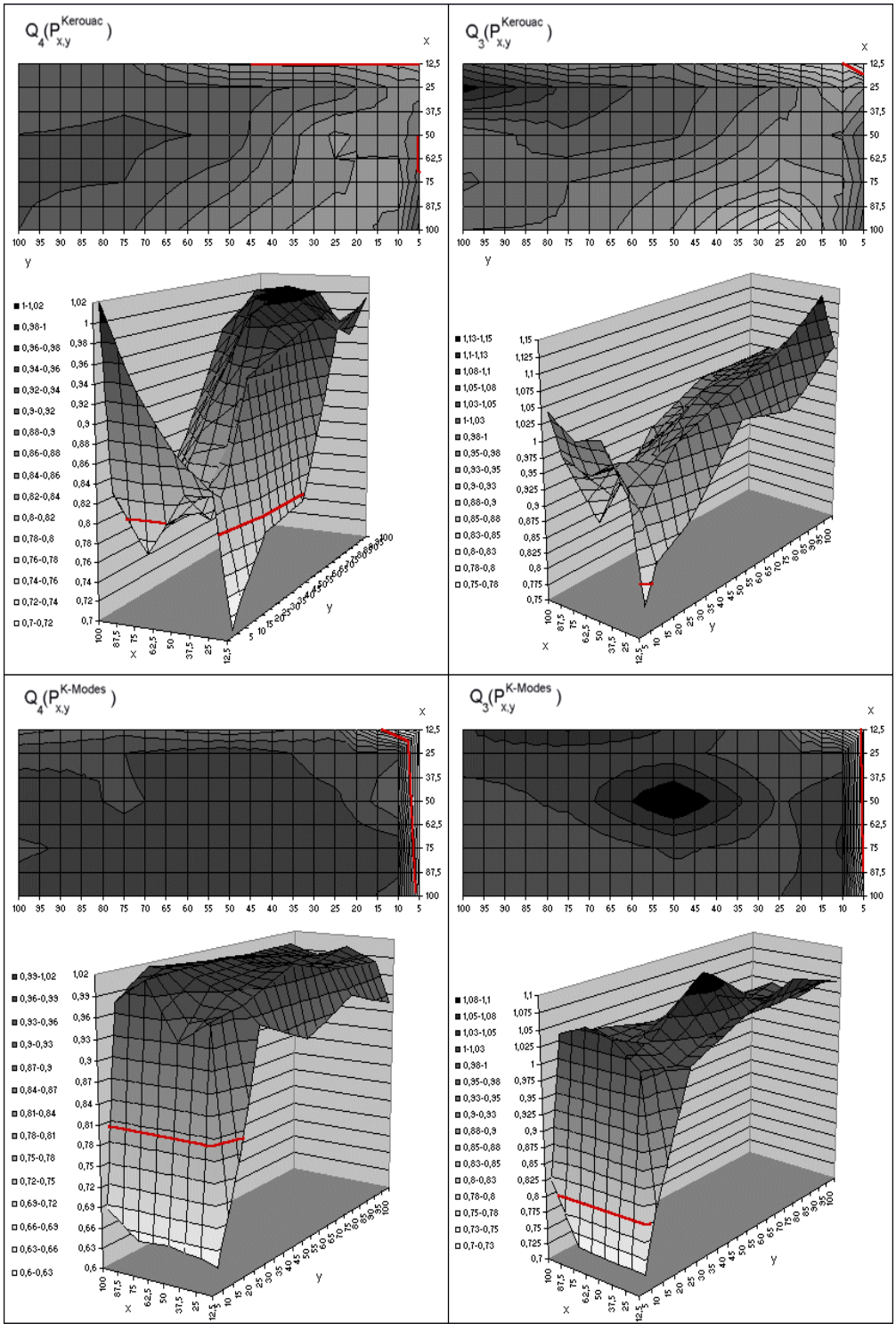


FIG. 6.12 –: *Evaluation de la qualité des cns issues de l'agrégation, Indice Q_4 et Q3 (jeu de données "1984 United States Congressional Voting Records Database")*

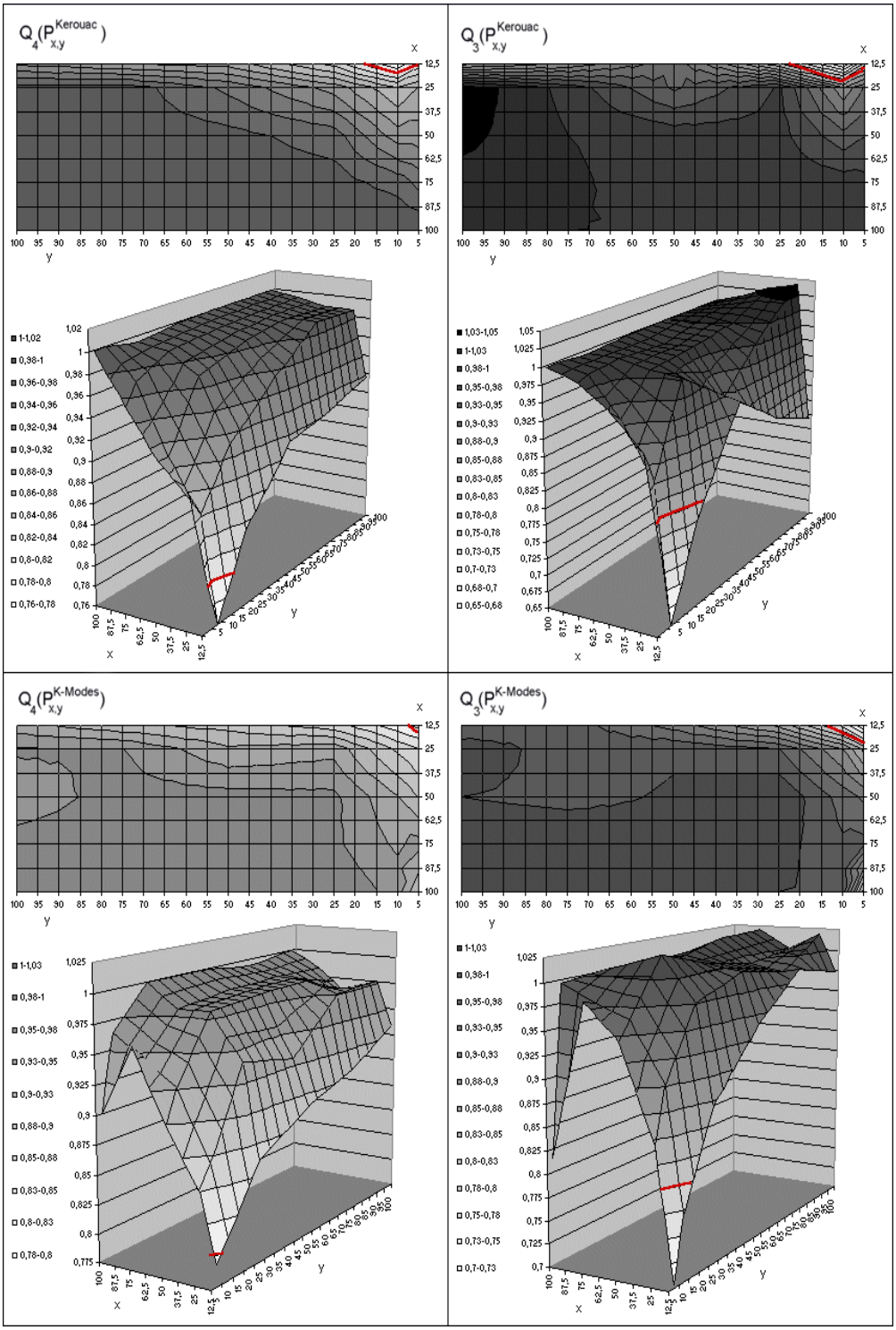


FIG. 6.13 –: *Evaluation de la qualité des cns issues de l'agrégation, Indice Q_4 et $Q3$ (jeu de données "mushrooms")*

série à l'agrégation de leurs 30 cns. La cns obtenue pour un niveau d'échantillonnage spécifique pour les objets ($x\%$) est par la suite notée P_x. Enfin, nous avons utilisé la méthode cns KEROUAC (avec le facteur de granularité fixé à 3) sur l'ensemble des variables et objets du jeu de données afin d'obtenir une cns de référence.

Les résultats concernant le nombre de classes des cns initiales ainsi que le nombre de classes des cns issues d'agrégation et l'indice Q_4 (définit comme précédemment) sont présentés sur les figures 6.14, 6.15.

Ces résultats montrent que la méthode ne semble pas être handicapée par la présence de cns initiales possédant des nombres de classes très différents puisque la qualité des cns résultant d'agrégation est excellente.

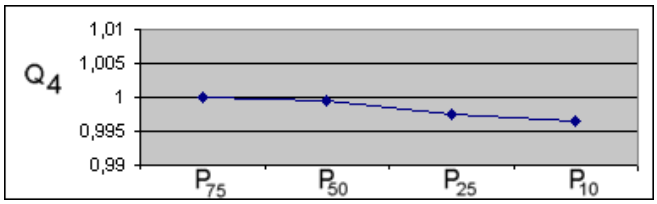


FIG. 6.14 –: *Indice Q_4 pour les 4 cns résultant d'agrégation*

De plus, le nombre de classes des cns issues d'agrégation (respectivement 24, 26, 24 et 26 classes) est relativement stable et proche du nombre de classes de la cns de reference (24 classes), et ce, même si les cns initiales possédaient des nombre de classes largement plus élevés. On peut ainsi conclure que la méthode n'est pas sensiblement affectée par des variations pour le nombre de classes des cns à agréger, et qu'il n'est pas vraiment problématique de ne pas connaître par avance le nombre de classes que doit posséder la cns résultant de l'agrégation.

6.4 Conclusion

Nous venons de présenter 3 méthodes d'agrégation de cns et avons évalué deux d'entre elles dans le cadre de la problématique "Cluster Ensembles". Ces évaluations ont montré que les résultats obtenus par l'intermédiaire de ces méthodes sont très intéressants : dans la plupart des cas la cns résultant de l'agrégation de cns obtenues par application d'un algorithme de cns prenant en compte uniquement un échantillon des objets d'un jeu de données et un échantillon des variables du même jeu de données est proche de la cns obtenue par application du même algorithme de cns sur l'intégralité du jeu de données. En fait, les expérimentations menées montrent que les résultats ne se dégradent que pour des échantillons de tailles relativement faibles : échantillons de tailles inférieures à 50% pour les variables et inférieures à 10% pour les objets.

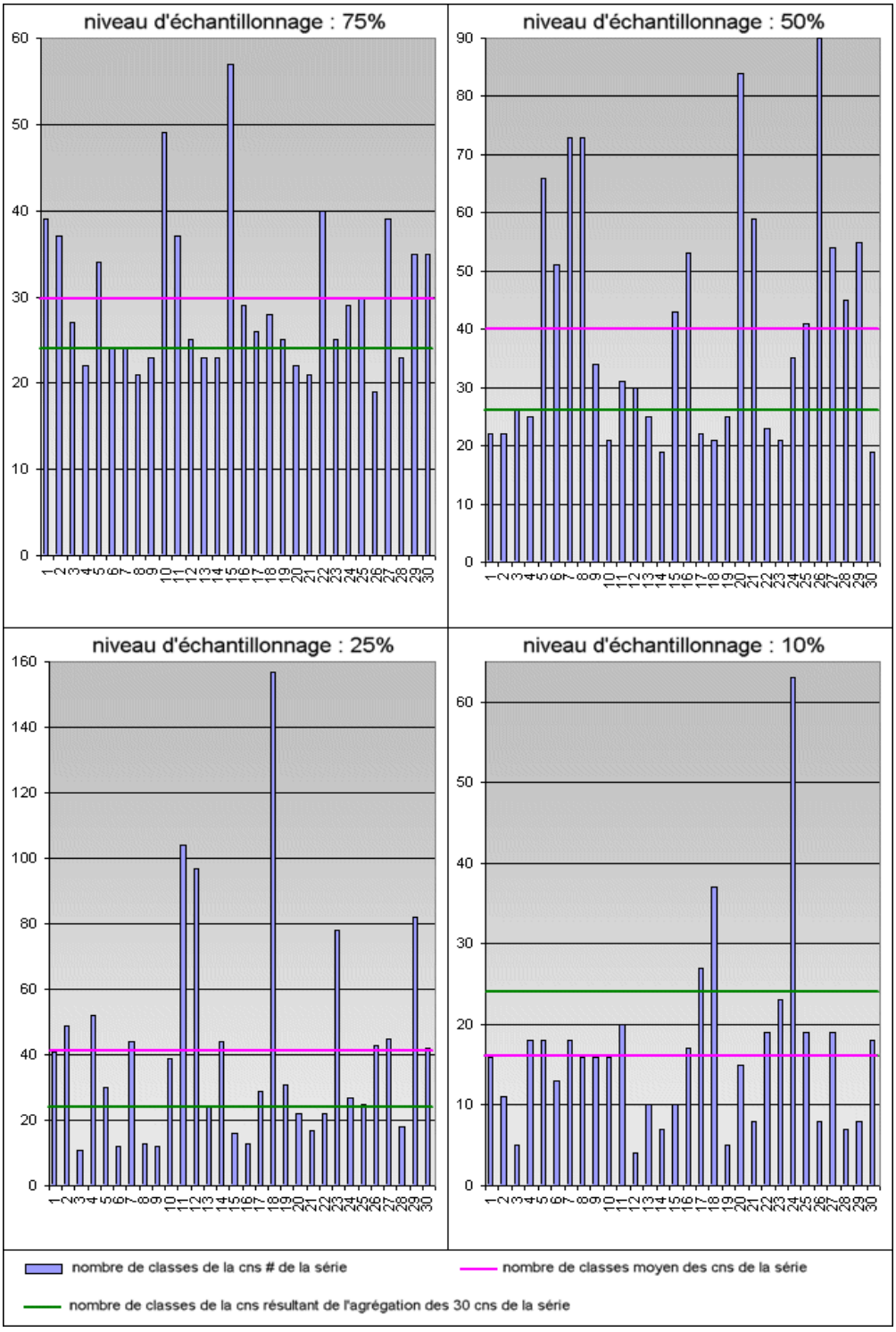


FIG. 6.15 –: *Nombre de classes pour les cns à agréger et les cns résultant d'agrégation*

L'utilisation de ces méthodes dans le cadre de la problématique "Cluster Ensembles" doit permettre de pratiquer la cns sur des bases de données distribuées, d'accroître la robustesse de cns, d'exploiter et d'intégrer des connaissances dans ce processus ou encore d'accélérer ce processus...

Nous désirons dans le cadre de futurs travaux proposer un ensemble d'expérimentations plus complet. Plus spécifiquement nous envisageons d'évaluer l'intérêt de ces méthodes pour le "Robust Centralized Clustering" (i.e. l'agrégation de modèles de cns issus de méthodologie différentes) ainsi que pour le traitement de données très hétérogènes.

Enfin, nous souhaitons également évaluer comment se comporterait une méthode d'agrégation de modèles d'apprentissage supervisé utilisant les méthodes présentées (il s'agit plus précisément de substituer nos méthodes d'agrégation à la technique d'agrégation à la majorité pour les forêts aléatoires)...

7 Conclusion

"On ne fait jamais attention à ce qui a été fait ; on ne voit que ce qui reste à faire."

- Marie Curie -

Les travaux présentés dans cette thèse participent donc en premier lieu de l'intégration de la classification non supervisée au sein d'un processus d'ECD.

Nous nous sommes penchés sur différentes étapes de ce processus (la sélection de variables au chapitre 5, l'application d'algorithmes de classification non supervisée au chapitre 3, l'évaluation de la validité des résultats au chapitre 4) en nous efforçant de proposer de nouvelles solutions originales permettant d'une part la résolution de problèmes ou l'intégration d'exigences issues de la pratique et d'autre part de bien prendre en compte le rôle central de l'utilisateur dans le processus ECD.

Ainsi, la méthode de sélection de variables proposée (pour la classification non supervisée) semble permettre le traitement de jeux de données plus volumineux en limitant le temps de traitement tout en assurant un bon niveau de qualité des résultats. Cette contribution nous apparaît intéressante d'une part car peu de méthodes permettant l'automatisation de ce processus existent, et d'autre part car certaines des méthodes existantes (telles celles fondées sur l'analyse factorielle) nécessitent d'appliquer l'algorithme de classification non supervisée sur un espace de représentation des données modifié (les axes principaux de l'analyse factorielle par exemple), rendant ainsi plus complexe l'interprétation des résultats.

La mise au point de la méthode de classification non supervisée (pour données catégorielles) KEROUAC a, quant à elle, été motivée par le désir de disposer d'une méthode à l'utilisabilité augmentée : sa mise en œuvre ne nécessite pas une expertise particulière tout comme l'analyse de ses résultats ne suppose pas une interprétation complexe à mener. De plus, cette méthode autorise le traitement de données manquantes ou encore l'intégration de contraintes, de connaissances qui constituent autant d'éléments bien utiles en pratique. Enfin, les éléments classiquement définis comme essentiels à une bonne méthode de classification non supervisée (validité et stabilité des résultats fournis, coût calculatoire "allégé"...) constituent autant de points non négligés par la méthode KEROUAC.

Evaluer aisément la qualité/validité des résultats d'un processus de classification non supervisée correspond là encore à une préoccupation importante tant d'un point de vue pratique qu'académique. Aussi avons nous proposé une méthodologie originale d'évaluation/comparaison de la validité de classification non supervisée, méthodologie s'appuyant sur une analyse graphique relativement intuitive. Cette méthodologie semble de plus se démarquer de la majorité des approches existantes de part son coût calculatoire limité combiné à la possibilité de comparer toutes formes de partitions, et de part l'aspect caractérisation visuelle d'un jeu de données qu'elle propose.

Enfin, de manière à intégrer mieux encore des exigences résultant de la pratique (traitement de données distribuées physiquement, traitement de données hétérogènes, réutilisation de connaissances, nécessité d'accélérer le processus de classification non supervisée...) nous avons introduit la problématique "Cluster Ensembles" et proposé l'utilisation de trois méthodes pour sa résolution. Les intérêts principaux de cette partie sont de souligner de manière simultanée les différents apports de l'agrégation de classification non supervisée et de proposer d'utiliser deux méthodes de classification non supervisée efficientes pour procéder à l'agrégation. Ce dernier point étant particulièrement intéressant car il semble démontrer que la simple réutilisation de méthodes existantes permet d'atteindre des résultats qualitativement aussi intéressants que ceux obtenus avec des méthodes spécifiques (en permettant même une accélération du processus d'agrégation pour la méthode K-Modes, ou la limitation du paramétrage dans le cas de la méthode KEROUAC).

Dénominateurs communs à l'ensemble de ces travaux, les concepts de comparaisons par paires, d'agrégation de préférences (concepts sous-jacents au critère de Condorcet) se sont avérés être des outils puissants.

Outre la présentation de solutions concrètes évoquées plus tôt, cette thèse vise également à indiquer que ces concepts véhiculent certainement des éléments de solutions à de nombreux problèmes alors qu'ils apparaissent actuellement sous-exploités.

Assistés des éléments théoriques fournit par la S-Théorie de Michaud [Mic87], [Mic91] (éléments visant notamment à la résolution efficace de problèmes d'optimisation posés dans le cadre précis de l'agrégation de préférences) nous pensons que ces concepts peuvent constituer des bases alternatives pour la mise au point de futures méthodes et méthodologies efficientes dans le cadre de l'ECD.

Des travaux additionnels, sortant du domaine du "Non Supervisé" et touchant au domaine du "Supervisé" ont également été abordés, soit en tant que

travaux aboutis (la méthode de sélection de variables pour l'apprentissage supervisé du chapitre 4), soit sous forme de prototype (la méthode d'apprentissage supervisé par apprentissage non supervisé sous contraintes du chapitre 3), ou simplement évoqués comme travaux à venir (mise au point de méthode d'apprentissage généralisé par le biais de KEROUAC (voir chapitre 3), utilisation des méthodes d'agrégation de classifications non supervisées dans le cadre de l'agrégation de classifieurs supervisés (voir chapitre 6)).

Concernant la méthode de sélection de variables pour l'apprentissage supervisé, les expérimentations menées la place au niveau des méthodes de référence actuelles tant pour la qualité des modèles d'apprentissage qu'elle permet de bâtir, que pour la réduction de l'espace de représentation des données qu'elle implique et le temps d'exécution qui lui est associé. Enfin, nous avons vu que cette méthode était fondée sur un certain nombre d'hypothèses dont certaines apparaissent trop strictes et dont la relaxation (envisagée dans des travaux futures) devrait (peut être ?) permettre l'amélioration de ces performances.

La méthode d'apprentissage non supervisé sous contraintes (qui n'est actuellement qu'un prototype) semble, quant à elle, fournir des modèles de qualité relativement similaire à celle des approches classiques, et son développement devrait permettre de répondre à un certain nombre de questions intéressantes touchant notamment à l'intérêt de l'apprentissage semi-supervisé.

Ces travaux (participant non plus de l'intégration de la classification non supervisée au sein du processus d'ECD mais de l'utilisation de la classification non supervisée pour d'autres éléments constitutifs d'un processus d'ECD) présentent donc un intérêt en tant que tel, mais, leur intégration dans ce document est également motivée par la volonté de souligner l'importance (et peut être la prééminence ?) de la structuration de l'information pour le processus d'ECD (la structuration de l'information étant abordée ici par le biais de la classification non supervisée).

En effet, si la relation entre classification non supervisée et apprentissage supervisé par les méthodes de type plus proche voisin est évidente (le concept "qui se ressemblent s'assemblent" sous-jacent à ce type de méthode d'apprentissage supervisé régit également la classification non supervisée) les méthodes d'apprentissage semi-supervisé mettant à profit la structure interne de l'information dans le cadre d'apprentissage supervisé sont encore peu nombreuses et peu exploitées. Ce constat peut être généralisé à l'ensemble des méthodes d'ECD : un faible nombre d'entre elles exploitent vraiment la structuration de l'information.

Le prototype de méthode d'apprentissage non supervisé sous contraintes, la méthode de sélection de variables pour l'apprentissage supervisé (présentée comme une méthode visant à déterminer l'espace de représentation des données tel que sa structure interne reflète au mieux la réalité à apprendre) doivent ainsi être, en partie, considérés comme autant d'éléments contribuant à montrer l'intérêt que peut revêtir l'apport de la structuration dans le cadre

d'un processus d'ECD. Notons enfin que, à l'opposé, il nous semble parfois utile d'adopter une perspective apprentissage supervisé afin de "féconder" des travaux touchant à la classification non supervisée comme le montre la méthode de sélection de variables pour l'apprentissage non supervisé qui peut être envisagée comme une généralisation de la méthode de sélection de variables proposée dans le cadre supervisé (plus spécifiquement une méthode de sélection de variables dans un cadre supervisé où plusieurs variables endogènes cohabitent).

Nous synthétisons finalement les contributions de cette thèse, les relations unissant les univers "Supervisé" et "Non Supervisé" apparaissant dans cette dissertations, ainsi que les travaux à venir sur les figures 7.1, 7.2.

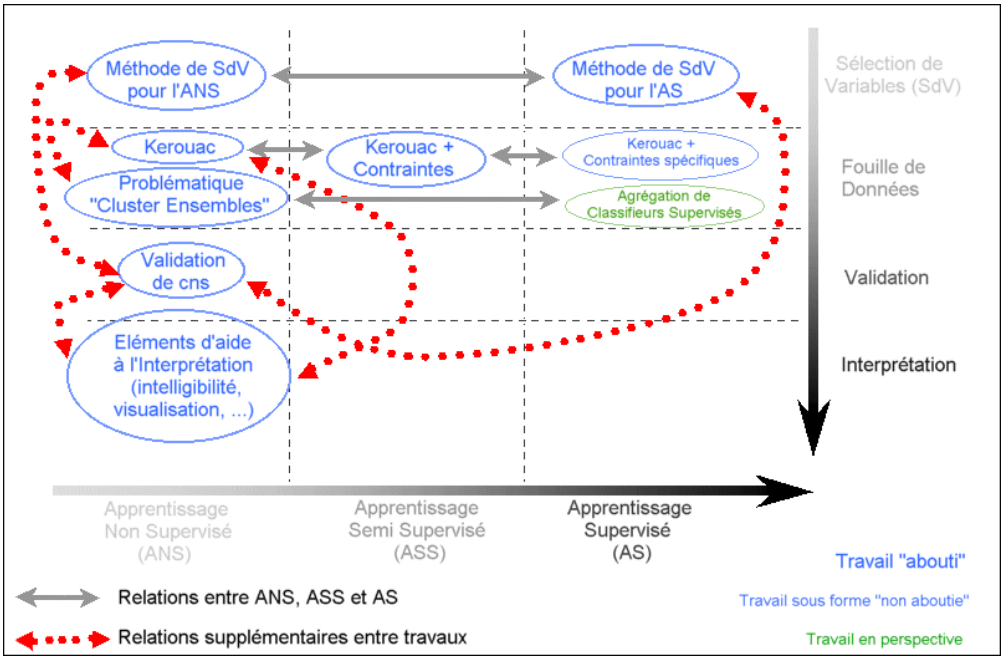


FIG. 7.1 –: *Synthèse des contributions, relations entre contributions et relations unissant entre "Supervisé", "Semi-Supervisé" et "Non Supervisé"*

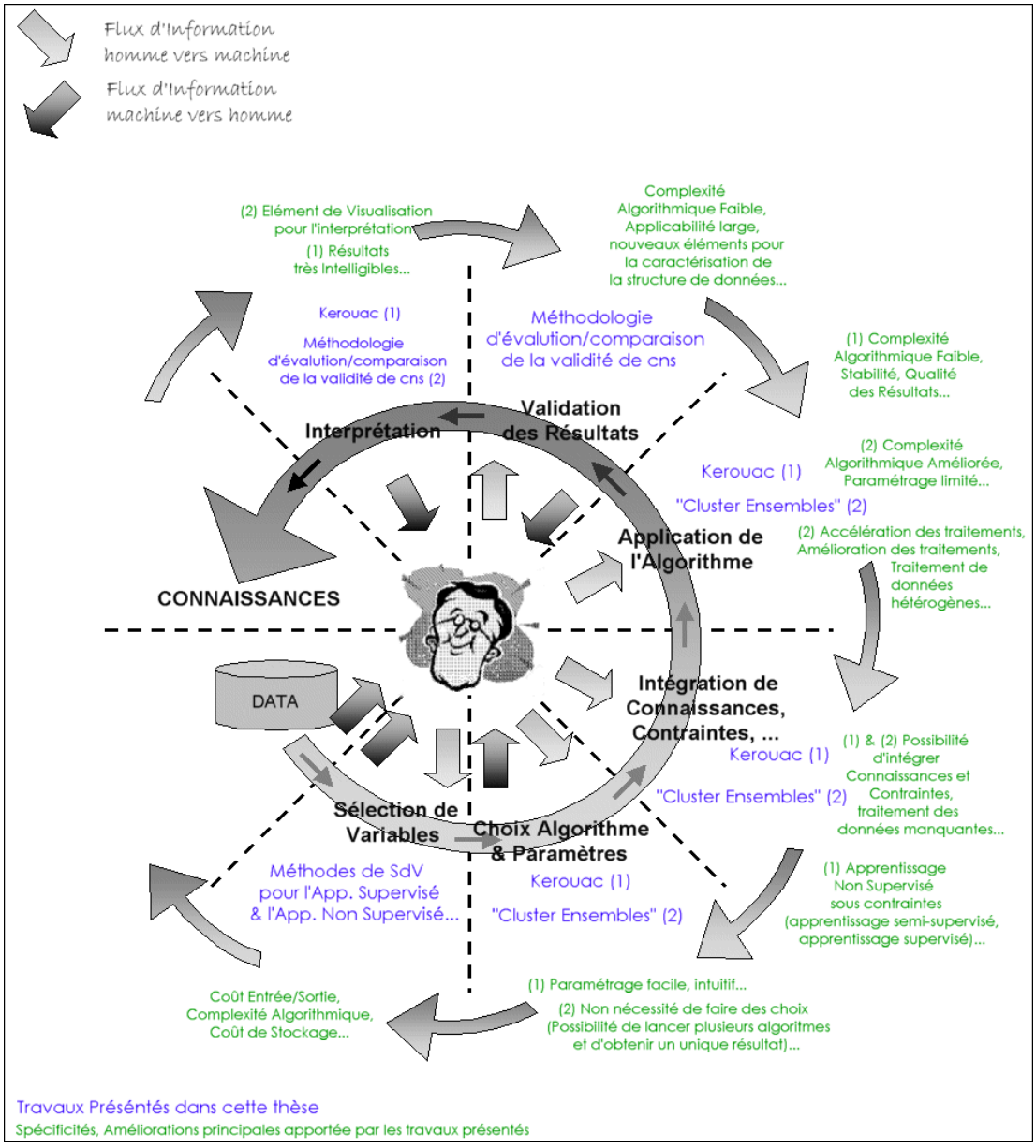


FIG. 7.2 –: *Synthèse des contributions*

8 Données Utilisées pour les Expérimentations

"The usefulness of data repositories like the one at UCI is subject to extreme positions : some people use them blindly without taking into account their limitations, while others simply reject them completely without trying to make adequate use of them. (...) In this paper we argue that, in principle, data repositories can be useful for KDD (...)"

- Carlos Soares -
Is the UCI Repository useful for Data Mining?
First International Workshop on Data Mining Lessons Learned (DMLL-2002)
http://www.hpl.hp.com/personal/Tom Fawcett/DMLL-2002/

Les jeux de données utilisés dans cette dissertation proviennent tous (à l'exception de quelques jeux de données synthétiques) de la collection de l'Université de Californie à IRVINE (http://www.ics.uci.edu/#mlearn/mlrepository.html) [MM96]. Le choix d'utiliser ces jeux de données est motivé ici, non pas par une croyance absolue en une qualité et une représentativité extrême de ces jeux de données, mais par la volonté de proposer des évaluations expérimentales reproductibles par d'autres chercheurs.

Nous reprenons dans les prochaines pages les descriptions des jeux de données telles qu'elles sont données dans le repertoire de l'UCI.

8.1 Jeu de Données ADULT

This data was extracted from the census bureau database found at :
http://www.census.gov/ftp/pub/DES/www/welcome.html

Donor: Ronny Kohavi and Barry Becker, Data Mining and Visualization Silicon Graphics.
e-mail: ronnyk@sgi.com for questions.
Split into train-test using MLC++ GenCV Files (2/3, 1/3 random).
48842 instances, mix of continuous and discrete (train=32561, test=16281)
45222 if instances with unknown values are removed (train=30162, test=15060)
Duplicate or conflicting instances : 6
Class probabilities for adult.all file
Probability for the label '>50K' : 23.93
Probability for the label '<=50K' : 76.07

Extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions:
((AAGE>16) & (AGI>100) & (AFNLWGT>1) & (HRSWK>0))
Prediction task is to determine whether a person makes over 50K a year.

First cited in: [Koh96]

Error Accuracy reported as follows, after removal of unknowns from train/test sets:
C4.5 : 84.46+-0.30
Naive-Bayes: 83.88+-0.30
NBTree : 85.90+-0.28
Following algorithms were later run with the following error rates, all after removal of unknowns and using the original train/test split. All these numbers are straight runs using MLC++ with default values.

8.2 Jeu de Données MUSHROOMS

1. Title: Mushroom Database

2. Sources:
(a) Mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred A. Knopf
(b) Donor: Jeff Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu)
(c) Date: 27 April 1987

3. Past Usage:
1. Schlimmer,J.S. (1987). Concept Acquisition Through Representational Adjustment (Technical Report 87-19). Doctoral disseration, Department of Information and Computer Science, University of California, Irvine. — STAGGER: asymptoted to 951000 instances.

2. Iba,W., Wogulis,J., & Langley,P. (1988). Trading off Simplicity and Coverage in Incremental Concept Learning. In Proceedings of the 5th International Conference on Machine Learning, 73-79. Ann Arbor, Michigan: Morgan Kaufmann. – approximately the same results with their HILLARY algorithm

3. In the following references a set of rules (given below) were learned for this data set which may serve as a point of comparison for other researchers.

Duch W, Adamczak R, Grabczewski K (1996) Extraction of logical rules from training data using backpropagation networks, in: Proc. of the The 1st Online Workshop on Soft Computing, 19-30.Aug.1996, pp. 25-30, available on-line at: http://www.bioele.nuee.nagoya-u.ac.jp/wsc1/

Duch W, Adamczak R, Grabczewski K, Ishikawa M, Ueda H, Extraction of crisp logical rules using constrained backpropagation networks - comparison of two new approaches, in: Proc. of the European Symposium on Artificial Neural Networks (ESANN'97), Bruge, Belgium 16-18.4.1997, pp. xx-xx

Wlodzislaw Duch, Department of Computer Methods, Nicholas Copernicus University, 87-100 Torun, Grudziadzka 5, Poland
e-mail: duch@phys.uni.torun.pl

WWW http://www.phys.uni.torun.pl/kmk/

Date: Mon, 17 Feb 1997 13:47:40 +0100
From: Wlodzislaw Duch <duch@phys.uni.torun.pl>
Organization: Dept. of Computer Methods, UMK

I have attached a file containing logical rules for mushrooms. It should be helpful for other people since only in the last year I have seen about 10 papers analyzing this dataset and obtaining quite complex rules. We will try to contribute other results later.

With best regards, Wlodek Duch

Logical rules for the mushroom data sets
Logical rules given below seem to be the simplest possible for the mushroom dataset and therefore should be treated as benchmark results.

Disjunctive rules for poisonous mushrooms, from most general to most specific:
P1) odor=NOT(almond.OR.anise.OR.none) 120 poisonous cases missed, 98.52% accuracy
P2) spore-print-color=green 48 cases missed, 99.41% accuracy
P3) odor=none.AND.stalk-surface-below-ring=scaly.AND. (stalk-color-above-ring=NOT.brown) 8 cases missed, 99.90% accuracy
P4) habitat=leaves.AND.cap-color=white 100% accuracy
Rule P4) may also be P4') population=clustered.AND.cap-color=white

These rule involve 6 attributes (out of 22). Rules for edible mushrooms are obtained as negation of the rules given above, for example the rule:
odor=(almond.OR.anise.OR.none).AND.spore-print-color=NOT.green
gives 48 errors, or 99.41% accuracy on the whole dataset.

Several slightly more complex variations on these rules exist, involving other attributes, such as gill-size, gill-spacing, stalk-surface-above-ring, but the rules given above are the simplest we have found.

4. Relevant Information:
This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family (pp. 500-525). Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one. The Guide clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like "leaflets three, let it be" for Poisonous Oak and Ivy.

5. Number of Instances: 8124

6. Number of Attributes: 22 (all nominally valued)

7. Attribute Information: (classes: edible=e, poisonous=p)

1. cap-shape: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
2. cap-surface: fibrous=f, grooves=g, scaly=y, smooth=s
3. cap-color: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
4. bruises?: bruises=t, no=f
5. odor: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
6. gill-attachment: attached=a, descending=d, free=f, notched=n
7. gill-spacing: close=c, crowded=w, distant=d
8. gill-size: broad=b, narrow=n
9. gill-color: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
10. stalk-shape: enlarging=e, tapering=t
11. stalk-root: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
12. stalk-surface-above-ring: ibrous=f, scaly=y, silky=k, smooth=s
13. stalk-surface-below-ring: ibrous=f, scaly=y, silky=k, smooth=s
14. stalk-color-above-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
15. stalk-color-below-ring: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
16. veil-type: partial=p, universal=u
17. veil-color: brown=n, orange=o, white=w, yellow=y
18. ring-number: none=n, one=o, two=t
19. ring-type: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
20. spore-print-color: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
21. population: abundant=a, clustered=c, numerous=n, scattered=s, several=v, solitary=y
22. habitat: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

8. Missing Attribute Values:
2480 of them (denoted by "?"), all for attribute #11.

9. Class Distribution:
– edible: 4208 (51.8
– poisonous: 3916 (48.2
– total: 8124 instances

8.3 Jeu de Données BREAST CANCER

Citation Request: This breast cancer databases was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. If you publish results when using this database, then please include this information in your acknowledgements. Also, please cite one or more of:

1. O. L. Mangasarian and W. H. Wolberg: "Cancer diagnosis via linear programming", SIAM News, Volume 23, Number 5, September 1990, pp 1 & 18.

2. William H. Wolberg and O.L. Mangasarian: "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", Proceedings of the National Academy of Sciences, U.S.A., Volume 87, December 1990, pp 9193-9196.

3. O. L. Mangasarian, R. Setiono, and W.H. Wolberg: "Pattern recognition via linear programming: Theory and application to medical diagnosis", in: "Large-scale nume-

rical optimization", Thomas F. Coleman and Yuying Li, editors, SIAM Publications, Philadelphia 1990, pp 22-30.

4. K. P. Bennett & O. L. Mangasarian: "Robust linear programming discrimination of two linearly inseparable sets", Optimization Methods and Software 1, 1992, 23-34 (Gordon & Breach Science Publishers).

1. Title: Wisconsin Breast Cancer Database (January 8, 1991)

2. Sources:
– Dr. WIlliam H. Wolberg (physician)
University of Wisconsin Hospitals
Madison, Wisconsin
USA
– Donor: Olvi Mangasarian (mangasarian@cs.wisc.edu)
Received by David W. Aha (aha@cs.jhu.edu)
– Date: 15 July 1992

3. Past Usage:

Attributes 2 through 10 have been used to represent instances. Each instance has one of 2 possible classes: benign or malignant.

1. Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *Proceedings of the National Academy of Sciences*, *87*, 9193–9196.
– Size of data set: only 369 instances (at that point in time)
– Collected classification results: 1 trial only
– Two pairs of parallel hyperplanes were found to be consistent with 50% of the data
  – Accuracy on remaining 50% of dataset: 93.5%
– Three pairs of parallel hyperplanes were found to be consistent with 67% of data
  – Accuracy on remaining 33% of dataset: 95.9%
2. Zhang, J. (1992). Selecting typical instances in instance-based learning. In *Proceedings of the Ninth International Machine Learning Conference* (pp. 470–479). Aberdeen, Scotland: Morgan Kaufmann.
– Size of data set: only 369 instances (at that point in time)
– Applied 4 instance-based learning algorithms
– Collected classification results averaged over 10 trials
– Best accuracy result:
  – 1-nearest neighbor: 93.7%
  – trained on 200 instances, tested on the other 169
– Also of interest:
  – Using only typical instances: 92.2% (storing only 23.1 instances)
  – trained on 200 instances, tested on the other 169

4. Relevant Information:

Samples arrive periodically as Dr. Wolberg reports his clinical cases. The database therefore reflects this chronological grouping of the data. This grouping information appears immediately below, having been removed from the data itself:

Group 1: 367 instances (January 1989)
Group 2: 70 instances (October 1989)
Group 3: 31 instances (February 1990)

Group 4: 17 instances (April 1990)
Group 5: 48 instances (August 1990)
Group 6: 49 instances (Updated January 1991)
Group 7: 31 instances (June 1991)
Group 8: 86 instances (November 1991)
———————————
Total: 699 points (as of the donated datbase on 15 July 1992)

Note that the results summarized above in Past Usage refer to a dataset of size 369, while Group 1 has only 367 instances. This is because it originally contained 369 instances; 2 were removed. The following statements summarizes changes to the original Group 1's set of data: Group 1 : 367 points: 200B 167M (January 1989)
Revised Jan 10, 1991: Replaced zero bare nuclei in 1080185 & 1187805
Revised Nov 22,1991: Removed 765878,4,5,9,7,10,10,10,3,8,1 no record ; Removed 484201,2,7,8,8,4,3,10,3,4,1 zero epithelial ; Changed 0 to 1 in field 6 of sample 1219406 ; Changed 0 to 1 in field 8 of following sample: 1182404,2,3,1,1,1,2,0,1,1,1

5. Number of Instances: 699 (as of 15 July 1992)

6. Number of Attributes: 10 plus the class attribute

7. Attribute Information:
(class attribute has been moved to last column)

Attribute ; Domain
– ————————————
1. Sample code number ; id number
2. Clump Thickness ; 1 - 10
3. Uniformity of Cell Size ; 1 - 10
4. Uniformity of Cell Shape ; 1 - 10
5. Marginal Adhesion ; 1 - 10
6. Single Epithelial Cell Size ; 1 - 10

Attribute ; Domain
————————————-
7. Bare Nuclei ; 1 - 10
8. Bland Chromatin ; 1 - 10
9. Normal Nucleoli ; 1 - 10
10. Mitoses ; 1 - 10
11. Class: ; 2 for benign, 4 for malignant

8. Missing attribute values: 16
There are 16 instances in Groups 1 to 6 that contain a single missing (i.e., unavailable) attribute value, now denoted by "?".

9. Class distribution:
Benign: 458 (65.5%)
Malignant: 241 (34.5%)

8.4 Jeu de Données CAR

1. Title: Car Evaluation Database

2. Sources: (a) Creator: Marko Bohanec
(b) Donors: Marko Bohanec (marko.bohanec@ijs.si), Blaz Zupan (blaz.zupan@ijs.si)
(c) Date: June, 1997

3. Past Usage:

The hierarchical decision model, from which this dataset is derived, was first presented in : M. Bohanec and V. Rajkovic: Knowledge acquisition and explanation for multi-attribute decision making. In 8th Intl Workshop on Expert Systems and their Applications, Avignon, France. pages 59-78, 1988.

Within machine-learning, this dataset was used for the evaluation of HINT (Hierarchy INduction Tool), which was proved to be able to completely reconstruct the original hierarchical model. This, together with a comparison with C4.5, is presented in : B. Zupan, M. Bohanec, I. Bratko, J. Demsar: Machine learning by function decomposition. ICML-97, Nashville, TN. 1997 (to appear)

4. Relevant Information Paragraph:
Car Evaluation Database was derived from a simple hierarchical decision model originally developed for the demonstration of DEX (M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.). The model evaluates cars according to the following concept structure:

CAR car acceptability . PRICE overall price . . buying buying price . . maint price of the maintenance . TECH technical characteristics . . COMFORT comfort . . . doors number of doors . . . persons capacity in terms of persons to carry . . . lug-boot the size of luggage boot . . safety estimated safety of the car

Input attributes are printed in lowercase. Besides the target concept (CAR), the model includes three intermediate concepts: PRICE, TECH, COMFORT. Every concept is in the original model related to its lower level descendants by a set of examples (for these examples sets see http://www-ai.ijs.si/BlazZupan/car.html).

The Car Evaluation Database contains examples with the structural information removed, i.e., directly relates CAR to the six input attributes: buying, maint, doors, persons, lug-boot, safety.

Because of known underlying concept structure, this database may be particularly useful for testing constructive induction and structure discovery methods.

5. Number of Instances: 1728 (instances completely cover the attribute space)

6. Number of Attributes: 6

7. Attribute Values:

# Attribute ; Domain	# Attribute ; Domain
buying ; v-high, high, med, low	persons ; 2, 4, more
maint ; v-high, high, med, low	lug-boot ; small, med, big
doors ; 2, 3, 4, 5-more	safety ; low, med, high

8. Missing Attribute Values: none

9. Class Distribution (number of instances per class)

class	N	N[%]
unacc	1210	(70.023 %)
acc	384	(22.222 %)
good	69	(3.993 %)
v-good	65	(3.762 %)

8.5 Jeu de Données : ADULT

1. Title of Database: adult

2. Sources:
(a) Original owners of database (name/phone/snail address/email address) US Census Bureau.
(b) Donor of database (name/phone/snail address/email address)
Ronny Kohavi and Barry Becker,
Data Mining and Visualization Silicon Graphics.
e-mail: ronnyk@sgi.com
(c) Date received (databases may change over time without name change!) 05/19/96

3. Past Usage:
(a) Complete reference of article where it was described/used [Koh96]
(b) Indication of what attribute(s) were being predicted Salary greater or less than 50,000.
(b) Indication of study's results (i.e. Is it a good domain to use?) Hard domain with a nice number of records. The following results obtained using MLC++ with default settings for the algorithms mentioned below.

Algorithm ; Error	Algorithm Error
1 C4.5 ; 15.54	10 HOODG ; 14.82
2 C4.5-auto ; 14.46	11 FSS Naive Bayes ; 14.05
3 C4.5 rules ; 14.94	12 IDTM (Decision table) ; 14.46
4 Voted ID3 (0.6) ; 15.64	13 Naive-Bayes ; 16.12
5 Voted ID3 (0.8) ; 16.47	14 Nearest-neighbor (1) ; 21.42
6 T2 ; 16.84	15 Nearest-neighbor (3) ; 20.35
7 1R ; 19.54	16 OC1 ; 15.04
8 NBTree ; 14.10	17 Pebls ; Crashed. Unknown why (bounds WERE increased)
9 CN2 ; 16.00	

4. Relevant Information Paragraph:
Extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) & (AGI>100) & (AFNLWGT>1) & (HRSWK>0))

5. Number of Instances
48842 instances, mix of continuous and discrete (train=32561, test=16281)
45222 if instances with unknown values are removed (train=30162, test=15060)
Split into train-test using MLC++ GenCVFiles (2/3, 1/3 random).

6. Number of Attributes
6 continuous, 8 nominal attributes.

7. Attribute Information:
age: continuous. workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

fnlwgt: continuous. education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num: continuous. marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex: Female, Male.

capital-gain: continuous.

capital-loss: continuous.

hours-per-week: continuous.

native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

class: >50K, <=50K

8. Missing Attribute Values: 7% have missing values.

9. Class Distribution:

Probability for the label '>50K' : 23.93% / 24.78% (without unknowns)

Probability for the label '<=50K' : 76.07% / 75.22% (without unknowns)

8.6 Jeu de Données Contraceptive Method Choice

1. Title: Contraceptive Method Choice

2. Sources:

(a) Origin: This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey

(b) Creator: Tjen-Sien Lim (limt@stat.wisc.edu)

(c) Donor: Tjen-Sien Lim (limt@stat.wisc.edu)

(c) Date: June 7, 1997

3. Past Usage:

Lim, T.-S., Loh, W.-Y. & Shih, Y.-S. (1999). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. Machine Learning. Forthcoming. (ftp://ftp.stat.wisc.edu/pub/loh/treeprogs/quest1.7/mach1317.pdf or (http://www.stat.wisc.edu/ limt/mach1317.pdf)

4. Relevant Information:

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not

know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

5. Number of Instances: 1473

6. Number of Attributes: 10 (including the class attribute)

7. Attribute Information:

1. Wife's age (numerical)
2. Wife's education (categorical) 1=low, 2, 3, 4=high
3. Husband's education (categorical) 1=low, 2, 3, 4=high
4. Number of children ever born (numerical)
5. Wife's religion (binary) 0=Non-Islam, 1=Islam
6. Wife's now working? (binary) 0=Yes, 1=No
7. Husband's occupation (categorical) 1, 2, 3, 4
8. Standard-of-living index (categorical) 1=low, 2, 3, 4=high
9. Media exposure (binary) 0=Good, 1=Not good
10. Contraceptive method used (class attribute) 1=No-use, 2=Long-term, 3=Short-term

8. Missing Attribute Values: None

8.7 Jeu de Données FLAGS

1. Title: Flag database

2. Source Information
– Creators: Collected primarily from the "Collins Gem Guide to Flags": Collins Publishers (1986).
– Donor: Richard S. Forsyth
8 Grosvenor Avenue
Mapperley Park
Nottingham NG3 5DX
0602-621676
– Date: 5/15/1990

3. Past Usage:
– None known other than what is shown in Forsyth's PC/BEAGLE User's Guide.

4. Relevant Information:
– This data file contains details of various nations and their flags. In this file the fields are separated by spaces (not commas). With this data you can try things like predicting the religion of a country from its size and the colours in its flag.
– 10 attributes are numeric-valued. The remainder are either Boolean- or nominal-valued.

5. Number of Instances: 194

6. Number of attributes: 30 (overall)

7. Attribute Information:

1. name Name of the country concerned
2. landmass 1=N.America, 2=S.America, 3=Europe, 4=Africa, 4=Asia, 6=Oceania
3. zone Geographic quadrant, based on Greenwich and the Equator 1=NE, 2=SE, 3=SW, 4=NW
4. area in thousands of square km
5. population in round millions
6. language 1=English, 2=Spanish, 3=French, 4=German, 5=Slavic, 6=Other Indo-European, 7=Chinese, 8=Arabic, 9=Japanese/Turkish/Finnish/Magyar, 10=Others
7. religion 0=Catholic, 1=Other Christian, 2=Muslim, 3=Buddhist, 4=Hindu, 5=Ethnic, 6=Marxist, 7=Others
8. bars Number of vertical bars in the flag
9. stripes Number of horizontal stripes in the flag
10. colours Number of different colours in the flag
11. red 0 if red absent, 1 if red present in the flag
12. green same for green
13. blue same for blue
14. gold same for gold (also yellow)
15. white same for white
16. black same for black
17. orange same for orange (also brown)
18. mainhue predominant colour in the flag (tie-breaks decided by taking the topmost hue, if that fails then the most central hue, and if that fails the leftmost hue)
19. circles Number of circles in the flag
20. crosses Number of (upright) crosses
21. saltires Number of diagonal crosses
22. quarters Number of quartered sections
23. sunstars Number of sun or star symbols
24. crescent 1 if a crescent moon symbol present, else 0
25. triangle 1 if any triangles present, 0 otherwise
26. icon 1 if an inanimate image present (e.g., a boat), otherwise 0
27. animate 1 if an animate image (e.g., an eagle, a tree, a human hand) present, 0 otherwise
28. text 1 if any letters or writing on the flag (e.g., a motto or slogan), 0 otherwise
29. topleft colour in the top-left corner (moving right to decide tie-breaks)
30. botright Colour in the bottom-left corner (moving left to decide tie-breaks)

8. Missing values: None

8.8 Jeu de Données GERMAN

1. Title: German Credit data

2. Source Information

Professor Dr. Hans Hofmann Institut f"ur Statistik und "Okonometrie Universit"at Hamburg FB Wirtschaftswissenschaften Von-Melle-Park 5 2000 Hamburg 13

3. Number of Instances: 1000

Two datasets are provided. the original dataset, in the form provided by Prof. Hofmann, contains categorical/symbolic attributes and is in the file "german.data".

For algorithms that need numerical attributes, Strathclyde University produced the file "german.data-numeric". This file has been edited and several indicator variables added to make it suitable for algorithms which cannot cope with categorical variables.

Several attributes that are ordered categorical (such as attribute 17) have been coded as integer. This was the form used by StatLog.

6. Number of Attributes : 20 (7 numerical, 13 categorical)

7. Attribute description

Attribute 1: (qualitative) Status of existing checking account : A11 : ... < 0 DM , A12 : 0 <= ... < 200 DM , A13 : ... >= 200 DM / salary assignments for at least 1 year , A14 : no checking account
Attribute 2: (numerical) Duration in month
Attribute 3: (qualitative) Credit history : A30 : no credits taken/ all credits paid back duly , A31 : all credits at this bank paid back duly , A32 : existing credits paid back duly till now , A33 : delay in paying off in the past , A34 : critical account/ other credits existing (not at this bank)
Attribute 4: (qualitative) Purpose
A40 : car (new) : A41 : car (used) , A42 : furniture/equipment , A43 : radio/television , A44 : domestic appliances , A45 : repairs , A46 : education , A47 : (vacation - does not exist?) , A48 : retraining , A49 : business , A410 : others
Attribute 5: (numerical) Credit amount
Attribute 6: (qualitative) Savings account/bonds : A61 : ... < 100 DM , A62 : 100 <= ... < 500 DM , A63 : 500 <= ... < 1000 DM , A64 : .. >= 1000 DM , A65 : unknown/ no savings account
Attribute 7: (qualitative) Present employment since : A71 : unemployed , A72 : ... < 1 year , A73 : 1 <= ... < 4 years , A74 : 4 <= ... < 7 years , A75 : .. >= 7 years
Attribute 8: (numerical) Installment rate in percentage of disposable income
Attribute 9: (qualitative) Personal status and sex : A91 : male : divorced/separated , A92 : female : divorced/separated/married , A93 : male : single , A94 : male : married/widowed , A95 : female : single
Attribute 10: (qualitative) Other debtors / guarantors : A101 : none , A102 : co-applicant , A103 : guarantor
Attribute 11: (numerical) Present residence since
Attribute 12: (qualitative) Property : A121 : real estate , A122 : if not A121 : building society savings agreement/ life insurance , A123 : if not A121/A122 : car or other, not in attribute 6 , A124 : unknown / no property
Attribute 13: (numerical) Age in years
Attribute 14: (qualitative) Other installment plans : A141 : bank , A142 : stores , A143 : none
Attribute 15: (qualitative) Housing : A151 : rent , A152 : own , A153 : for free
Attribute 16: (numerical) Number of existing credits at this bank
Attribute 17: (qualitative) Job : A171 : unemployed/ unskilled - non-resident , A172 : unskilled - resident , A173 : skilled employee / official , , A174 : management/ self-employed/ highly qualified employee/ officer
Attribute 18: (numerical) Number of people being liable to provide maintenance for
Attribute 19: (qualitative) Telephone : A191 : none , A192 : yes, registered under the customers name
Attribute 20: (qualitative) foreign worker : A201 : yes , A202 : no

8. Cost Matrix

This dataset requires use of a cost matrix (see below)

	1	2
1	0	1
2	5	0

(1 = Good, 2 = Bad)

the rows represent the actual classification and the columns the predicted classification.

It is worse to class a customer as good when they are bad (5), than it is to class a customer as bad when they are good (1).

8.9 Jeu de Données HOUSE VOTES 84

1. Title: 1984 United States Congressional Voting Records Database

2. Source Information:
(a) Source: Congressional Quarterly Almanac, 98th Congress, 2nd session 1984, Volume XL: Congressional Quarterly Inc. Washington, D.C., 1985.

(b) Donor: Jeff Schlimmer (Jeffrey.Schlimmer@a.gp.cs.cmu.edu)

(c) Date: 27 April 1987

3. Past Usage

- Publications

1. Schlimmer, J. C. (1987). Concept acquisition through representational adjustment. Doctoral dissertation, Department of Information and Computer Science, University of California, Irvine, CA.

– Results: about 90%-95% accuracy appears to be STAGGER's asymptote

- Predicted attribute: party affiliation (2 classes)

4. Relevant Information:

This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of votes: voted for, paired for, and announced for (these three simplified to yea), voted against, paired against, and announced against (these three simplified to nay), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known (these three simplified to an unknown disposition).

5. Number of Instances: 435 (267 democrats, 168) republicans)

6. Number of Attributes: 16 + class name = 17 (all Boolean valued)

7. Attribute Information:
1. Class Name: 2 (democrat, republican)
2. handicapped-infants: 2 (y,n)
3. water-project-cost-sharing: 2 (y,n)
4. adoption-of-the-budget-resolution: 2 (y,n)
5. physician-fee-freeze: 2 (y,n)
6. el-salvador-aid: 2 (y,n)
7. religious-groups-in-schools: 2 (y,n)
8. anti-satellite-test-ban: 2 (y,n)
9. aid-to-nicaraguan-contras: 2 (y,n)
10. mx-missile: 2 (y,n)
11. immigration: 2 (y,n)
12. synfuels-corporation-cutback: 2 (y,n)
13. education-spending: 2 (y,n)
14. superfund-right-to-sue: 2 (y,n)
15. crime: 2 (y,n)
16. duty-free-exports: 2 (y,n)
17. export-administration-act-south-africa: 2 (y,n)

8. Missing Attribute Values: Denoted by "?"

NOTE: It is important to recognize that "?" in this database does not mean that the value of the attribute is unknown. It means simply, that the value is not "yea" or "nay" (see "Relevant Information" section above).

Attribute: Missing Values:

1: 0 ; 2: 0 ; 3: 12 ; 4: 48 ; 5: 11 ; 6: 11 ; 7: 15 ; 8: 11 ; 9: 14 ; 10: 15 ; 11: 22 ; 12: 7 ; 13: 21 ; 14: 31 ; 15: 25 ; 16: 17 ; 17: 28

9. Class Distribution: (2 classes)

1. 45.2 percent are democrat
2. 54.8 percent are republican

8.10 Jeu de Données IONOSPHERE

1. Title: Johns Hopkins University Ionosphere database

2. Source Information:

– Donor: Vince Sigillito (vgs@aplcen.apl.jhu.edu)
– Date: 1989
– Source: Space Physics Group
Applied Physics Laboratory
Johns Hopkins University
Johns Hopkins Road
Laurel, MD 20723

3. Past Usage:

– Sigillito, V. G., Wing, S. P., Hutton, L. V., & Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest, 10, 262-266.

They investigated using backprop and the perceptron training algorithm on this database. Using the first 200 instances for training, which were carefully split almost 50% positive and 50% negative, they found that a "linear" perceptron attained 90.7%, a "non-linear" perceptron attained 92%, and backprop an average of over 96% accuracy on the remaining 150 test instances, consisting of 123 "good" and only 24 "bad" instances. (There was a counting error or some mistake somewhere; there are a total of 351 rather than 350 instances in this domain.) Accuracy on "good" instances was much higher than for "bad" instances. Backprop was tested with several different numbers of hidden units (in [0,15]) and incremental results were also reported (corresponding to how well the different variants of backprop did after a periodic number of epochs).

David Aha (aha@ics.uci.edu) briefly investigated this database. He found that nearest neighbor attains an accuracy of 92.1%, that Ross Quinlan's C4 algorithm attains 94.0% (no windowing), and that IB3 (Aha & Kibler, IJCAI-1989) attained 96.7% (parameter settings: 70% and 80% for acceptance and dropping respectively).

4. Relevant Information:

This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. See the paper for more details. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some

type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere.

Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this databse are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.

5. Number of Instances: 351

6. Number of Attributes: 34 plus the class attribute
– All 34 predictor attributes are continuous

7. Attribute Information:
– All 34 are continuous, as described above
– The 35th attribute is either "good" or "bad" according to the definition summarized above. This is a binary classification task.

8. Missing Values: None

8.11 Jeu de Données MONKS

1. Title: The Monk's Problems

2. Sources:
(a) Donor: Sebastian Thrun
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

E-mail: thrun@cs.cmu.edu
(b) Date: October 1992

3. Past Usage:
- See File: thrun.comparison.ps.Z
- Wnek, J., "Hypothesis-driven Constructive Induction," PhD dissertation, School of Information Technology and Engineering, Reports of Machine Learning and Inference Laboratory, MLI 93-2, Center for Artificial Intelligence, George Mason University, March 1993.
- Wnek, J. and Michalski, R.S., "Comparing Symbolic and Subsymbolic Learning: Three Studies," in Machine Learning: A Multistrategy Approach, Vol. 4., R.S. Michalski and G. Tecuci (Eds.), Morgan Kaufmann, San Mateo, CA, 1993.

4. Relevant Information:
The MONK's problem were the basis of a first international comparison of learning algorithms. The result of this comparison is summarized in "The MONK's Problems - A Performance Comparison of Different Learning algorithms" by S.B. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. De Jong, S. Dzeroski, S.E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R.S. Michalski,

T. Mitchell, P. Pachowicz, Y. Reich H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, and J. Zhang has been published as Technical Report CS-CMU-91-197, Carnegie Mellon University in Dec. 1991.

One significant characteristic of this comparison is that it was performed by a collection of researchers, each of whom was an advocate of the technique they tested (often they were the creators of the various methods). In this sense, the results are less biased than in comparisons performed by a single person advocating a specific learning method, and more accurately reflect the generalization behavior of the learning techniques as applied by knowledgeable users.

There are three MONK's problems. The domains for all MONK's problems are the same (described below). One of the MONK's problems has noise added. For each problem, the domain has been partitioned into a train and test set.

5. Number of Instances: 432

6. Number of Attributes: 8 (including class attribute)

7. Attribute information:

1. class: 0, 1
2. a1: 1, 2, 3
3. a2: 1, 2, 3
4. a3: 1, 2
5. a4: 1, 2, 3
6. a5: 1, 2, 3, 4
7. a6: 1, 2
8. Id: (A unique symbol for each instance)

8. Missing Attribute Values: None

9. Target Concepts associated to the MONK's problem:

MONK-1: (a1 = a2) or (a5 = 1)

MONK-2: EXACTLY TWO of a1 = 1, a2 = 1, a3 = 1, a4 = 1, a5 = 1, a6 = 1

MONK-3: (a5 = 3 and a4 = 1) or (a5 /= 4 and a2 /= 3) (5% class noise added to the training set)

8.12 Jeu de Données NURSERY

1. Title: Nursery Database

2. Sources:

(a) Creator: Vladislav Rajkovic et al. (13 experts)

(b) Donors: Marko Bohanec (marko.bohanec@ijs.si), Blaz Zupan (blaz.zupan@ijs.si)

(c) Date: June, 1997

3. Past Usage:

The hierarchical decision model, from which this dataset is derived, was first presented in

M. Olave, V. Rajkovic, M. Bohanec: An application for admission in public school systems. In (I. Th. M. Snellen and W. B. H. J. van de Donk and J.-P. Baquiast, editors) Expert Systems in Public Administration, pages 145-160. Elsevier Science Publishers (North Holland), 1989.

Within machine-learning, this dataset was used for the evaluation of HINT (Hierarchy INduction Tool), which was proved to be able to completely reconstruct the original hierarchical model. This, together with a comparison with C4.5, is presented in

B. Zupan, M. Bohanec, I. Bratko, J. Demsar: Machine learning by function decomposition. ICML-97, Nashville, TN. 1997 (to appear)

4. Relevant Information Paragraph:

Nursery Database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It was used during several years in 1980's when there was excessive enrollment to these schools in Ljubljana, Slovenia, and the rejected applications frequently needed an objective explanation. The final decision depended on three subproblems: occupation of parents and child's nursery, family structure and financial standing, and social and health picture of the family. The model was developed within expert system shell for decision making DEX (M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.).

The hierarchical model ranks nursery-school applications according to the following concept structure:

NURSERY Evaluation of applications for nursery schools
. EMPLOY Employment of parents and child's nursery
. . parents Parents' occupation
. . has-nurs Child's nursery
. STRUCT-FINAN Family structure and financial standings
. . STRUCTURE Family structure
. . . form Form of the family
. . . children Number of children
. . housing Housing conditions
. . finance Financial standing of the family
. SOC-HEALTH Social and health picture of the family
. . social Social conditions
. . health Health conditions

Input attributes are printed in lowercase. Besides the target concept (NURSERY) the model includes four intermediate concepts: EMPLOY, STRUCT-FINAN, STRUCTURE, SOC-HEALTH. Every concept is in the original model related to its lower level descendants by a set of examples (for these examples sets see http://www-ai.ijs.si/BlazZupan/nursery.html).

The Nursery Database contains examples with the structural information removed, i.e., directly relates NURSERY to the eight input attributes: parents, has-nurs, form, children, housing, finance, social, health.

Because of known underlying concept structure, this database may be particularly useful for testing constructive induction and structure discovery methods.

5. Number of Instances: 12960 (instances completely cover the attribute space)

6. Number of Attributes: 8

7. Attribute Values:
parents : usual, pretentious, great-pret
has-nurs : proper, less-proper, improper, critical, very-crit
form : complete, completed, incomplete, foster
children : 1, 2, 3, more
housing : convenient, less-conv, critical
finance : convenient, inconv
social : non-prob, slightly-prob, problematic
health : recommended, priority, not-recom

8. Missing Attribute Values: none

9. Class Distribution (number of instances per class)
class N N[%]
———————————
not-recom 4320 (33.333 %)
recommend 2 (0.015 %)
very-recom 328 (2.531 %)
priority 4266 (32.917 %)
spec-prior 4044 (31.204 %)

8.13 Jeu de Données PIMA

1. Title: Pima Indians Diabetes Database

2. Sources:
(a) Original owners: National Institute of Diabetes and Digestive and Kidney Diseases
(b) Donor of database: Vincent Sigillito (vgs@aplcen.apl.jhu.edu)
Research Center, RMI Group Leader
Applied Physics Laboratory
The Johns Hopkins University
Johns Hopkins Road
Laurel, MD 20707
(301) 953-6231
(c) Date received: 9 May 1990

3. Past Usage:
1. Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261–265). IEEE Computer Society Press.

The diagnostic, binary-valued variable investigated is whether the patient shows signs of diabetes according to World Health Organization criteria (i.e., if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care). The population lives near Phoenix, Arizona, USA.

Results: Their ADAP algorithm makes a real-valued prediction between 0 and 1. This was transformed into a binary decision using a cutoff of 0.448. Using 576 training instances, the sensitivity and specificity of their algorithm was 76% on the remaining 192 instances.

4. Relevant Information:
Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. ADAP is an adaptive learning routine that generates and executes digital analogs of perceptron-like devices. It is a unique algorithm; see the paper for details.

5. Number of Instances: 768

6. Number of Attributes: 8 plus class

7. For Each Attribute: (all numeric-valued)
1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

8. Missing Attribute Values: None

9. Class Distribution: (class value 1 is interpreted as "tested positive for diabetes")

Class Value	Number of instances
0	500
1	268

10. Brief statistical analysis:

Attribute:	Mean:	S. Deviation:	Attribute:	Mean:	S. Deviation:
1.	3.8	3.4	5.	79.8	115.2
2.	120.9	32.0	6.	32.0	7.9
3.	69.1	19.4	7.	0.5	0.3
4.	20.5	16.0	8.	33.2	11.8

8.14 Jeu de Données SICK

Thyroid disease records supplied by the Garavan Institute and J. Ross Quinlan, New South Wales Institute, Syndney, Australia 1987.

sick, negative. | classes
age: continuous.
sex: M, F. on
thyroxine: f, t.
query on thyroxine: f, t.
on antithyroid medication: f, t.
sick: f, t.
pregnant: f, t.
thyroid surgery: f, t.
I131 treatment: f, t.
query hypothyroid: f, t.
query hyperthyroid: f, t.
lithium: f, t.
goitre: f, t.
tumor: f, t.
hypopituitary: f, t.
psych: f, t.
TSH measured: f, t.
TSH: continuous.
T3 measured: f, t.
T3: continuous.
TT4 measured: f, t.
TT4: continuous.
T4U measured: f, t.
T4U: continuous.
FTI measured: f, t.
FTI: continuous.
TBG measured: f, t.
TBG: continuous.
referral source: WEST, STMW, SVHC, SVI, SVHD, other.

8.15 Jeu de Données SMALL SOYBEAN DISEASES

1. Title: Small Soybean Database

2. Sources:
(a) Michalski,R.S. Learning by being told and learning from examples: an experimental comparison of the two methodes of knowledge acquisition in the context of developing an expert system for soybean desease diagnoiss", International Journal of Policy Analysis and Information Systems, 1980, 4(2), 125-161.
(b) Donor: Doug Fisher (dfisher%vuse@uunet.uucp)
(c) Date: 1987

3. Past Usage:
1. R.S. Michalski and R.L. Chilausky "Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis", International Journal of Policy Analysis and Information Systems, Vol. 4, No. 2, 1980.
2. Tan, M., & Eshelman, L. (1988). Using weighted networks to represent classification knowledge in noisy domains. Proceedings of the Fifth International Conference on Machine Learning (pp. 121-134). Ann Arbor, Michigan: Morgan Kaufmann. – IWN recorded a 97.1% classification accuracy – 290 training and 340 test instances
3. Fisher,D.H. & Schlimmer,J.C. (1988). Concept Simplification and Predictive Accuracy. Proceedings of the Fifth International Conference on Machine Learning (pp. 22-28). Ann Arbor, Michigan: Morgan Kaufmann. – Notes why this database is highly predictable

4. Relevant Information Paragraph:
A small subset of the original soybean database. See the reference for Fisher and Schlimmer in soybean-large.names for more information.
Steven Souders wrote:
> Figure 15 in the Michalski and Stepp paper (PAMI-82) says that the > discriminant values for the attribute CONDITION OF FRUIT PODS for the > classes Rhizoctonia Root Rot and Phytophthora Rot are "few or none" > and "irrelevant" respectively. However, in the SOYBEAN-SMALL dataset > I got from UCI, the value for this attribute is "dna" (does not apply) > for both classes. I show the actual data below for cases D3 > (Rhizoctonia Root Rot) and D4 (Phytophthora Rot). According to the > attribute names given in soybean-large.names, FRUIT-PODS is attribute > #28. If you look at column 28 in the data below (marked with arrows) > you'll notice that all cases of D3 and D4 have the same value. Thus, > the SOYBEAN-SMALL dataset from UCI could NOT have produced the results > in the Michalski and Stepp paper.
I do not have that paper, but have found what is probably a later variation of that figure in Stepp's dissertation, which lists the value "normal" for the first 2 classes and "irrelevant" for the latter 2 classes. I believe that "irrelevant" is used here as a synonym for "not-applicable", "dna", and "does-not-apply". I believe that there is a mis-print in the figure he read in their PAMI-83 article.
I have checked over each attribute value in this database. It corresponds exactly with the copies listed in both Stepp's and Fisher's dissertations.

5. Number of Instances: 47

6. Number of Attributes: 35 (all have been nominalized)
– All attributes here appear with numeric values

7. Attribute Information:
Classes : diaporthe-stem-canker (D1), charcoal-rot(D2), rhizoctonia-root-rot (D3), phytophthora-rot (D4)
1. date: april,may,june,july,august,september,october,?.
2. plant-stand: normal,lt-normal,?.
3. precip: lt-norm,norm,gt-norm,?.
4. temp: lt-norm,norm,gt-norm,?.
5. hail: yes,no,?.
6. crop-hist: diff-lst-year,same-lst-yr,same-lst-two-yrs, same-lst-sev-yrs,?.
7. area-damaged: scattered,low-areas,upper-areas,whole-field,?.
8. severity: minor,pot-severe,severe,?.
9. seed-tmt: none,fungicide,other,?.
10. germination: 90-100
11. plant-growth: norm,abnorm,?.
12. leaves: norm,abnorm.
13. leafspots-halo: absent,yellow-halos,no-yellow-halos,?.
14. leafspots-marg: w-s-marg,no-w-s-marg,dna,?.
15. leafspot-size: lt-1/8,gt-1/8,dna,?.
16. leaf-shread: absent,present,?.
17. leaf-malf: absent,present,?.
18. leaf-mild: absent,upper-surf,lower-surf,?.
19. stem: norm,abnorm,?.
20. lodging: yes,no,?.
21. stem-cankers: absent,below-soil,above-soil,above-sec-nde,?.
22. canker-lesion: dna,brown,dk-brown-blk,tan,?.
23. fruiting-bodies: absent,present,?.
24. external decay: absent,firm-and-dry,watery,?.
25. mycelium: absent,present,?.
26. int-discolor: none,brown,black,?.
27. sclerotia: absent,present,?.
28. fruit-pods: norm,diseased,few-present,dna,?.
29. fruit spots: absent,colored,brown-w/blk-specks,distort,dna,?.
30. seed: norm,abnorm,?.
31. mold-growth: absent,present,?.
32. seed-discolor: absent,present,?.
33. seed-size: norm,lt-norm,?.
34. shriveling: absent,present,?.
35. roots: norm,rotted,galls-cysts,?.

8. Number of Missing Attribute Values: 0

9. Class Distribution:
1. D1: 10
2. D2: 10
3. D3: 10
4. D4: 17

8.16 Jeu de Données VEHICLE

This dataset comes from the Turing Institute, Glasgow, Scotland. If you use this dataset in any publication you must acknowledge this source.

NAME : vehicle silhouettes

PURPOSE :
to classify a given silhouette as one of four types of vehicle, using a set of features extracted from the silhouette. The vehicle may be viewed from one of many different

angles.

PROBLEM TYPE : classification

SOURCE
Drs.Pete Mowforth and Barry Shepherd
Turing Institute
George House
36 North Hanover St.
Glasgow
G1 2AD

CONTACT
Alistair Sutherland
Statistics Dept.
Strathclyde University
Livingstone Tower
26 Richmond St.
GLASGOW G1 1XH
Great Britain
Tel: 041 552 4400 x3033
Fax: 041 552 4711
e-mail: alistair@uk.ac.strathclyde.stams

HISTORY :
This data was originally gathered at the TI in 1986-87 by JP Siebert. It was partially financed by Barr and Stroud Ltd. The original purpose was to find a method of distinguishing 3D objects within a 2D image by application of an ensemble of shape feature extractors to the 2D silhouettes of the objects. Measures of shape features extracted from example silhouettes of objects to be discriminated were used to generate a classification rule tree by means of computer induction. This object recognition strategy was successfully used to discriminate between silhouettes of model cars, vans and buses viewed from constrained elevation but all angles of rotation. The rule tree classification performance compared favourably to MDC (Minimum Distance Classifier) and k-NN (k-Nearest Neigh- bour) statistical classifiers in terms of both error rate and computational efficiency. An investigation of these rule trees generated by example indicated that the tree structure was heavily influenced by the orientation of the objects, and grouped similar object views into single decisions.

DESCRIPTION :
The features were extracted from the silhouettes by the HIPS (Hierarchical Image Processing System) extension BINATTS, which extracts a combination of scale independent features utilising both classical moments based measures such as scaled variance, skewness and kurtosis about the major/minor axes and heuristic measures such as hollows, circularity, rectangularity and compactness. Four "Corgie" model vehicles were used for the experiment: a double decker bus, Cheverolet van, Saab 9000 and an Opel Manta 400. This particular combination of vehicles was chosen with the expectation that the bus, van and either one of the cars would be readily distinguishable, but it would be more difficult to distinguish between the cars. The images were acquired by a camera looking downwards at the model vehicle from a fixed angle of

elevation (34.2 degrees to the horizontal). The vehicles were placed on a diffuse backlit surface (lightbox). The vehicles were painted matte black to minimise highlights. The images were captured using a CRS4000 framestore connected to a vax 750. All images were captured with a spatial resolution of 128x128 pixels quantised to 64 greylevels. These images were thresholded to produce binary vehicle silhouettes, negated (to comply with the processing requirements of BINATTS) and thereafter subjected to shrink-expand-expand-shrink HIPS modules to remove "salt and pepper" image noise. The vehicles were rotated and their angle of orientation was measured using a radial graticule beneath the vehicle. 0 and 180 degrees corresponded to "head on" and "rear" views respectively while 90 and 270 corresponded to profiles in opposite directions. Two sets of 60 images, each set covering a full 360 degree rotation, were captured for each vehicle. The vehicle was rotated by a fixed angle between images. These datasets are known as e2 and e3 respectively. A further two sets of images, e4 and e5, were captured with the camera at elevations of 37.5 degs and 30.8 degs respectively. These sets also contain 60 images per vehicle apart from e4.van which contains only 46 owing to the difficulty of containing the van in the image at some orientations.

ATTRIBUTES :

COMPACTNESS (average perim)**2/area

CIRCULARITY (average radius)**2/area

DISTANCE CIRCULARITY area/(av.distance from border)**2

RADIUS RATIO (max.rad-min.rad)/av.radius

PR.AXIS ASPECT RATIO (minor axis)/(major axis)

MAX.LENGTH ASPECT RATIO (length perp. max length)/(max length)

SCATTER RATIO (inertia about minor axis)/(inertia about major axis)

ELONGATEDNESS area/(shrink width)**2

PR.AXIS RECTANGULARITY area/(pr.axis length*pr.axis width)

MAX.LENGTH RECTANGULARITY area/(max.length*length perp. to this)

SCALED VARIANCE (2nd order moment about minor axis)/area ALONG MAJOR AXIS

SCALED VARIANCE (2nd order moment about major axis)/area ALONG MINOR AXIS

SCALED RADIUS OF GYRATION (mavar+mivar)/area

SKEWNESS ABOUT (3rd order moment about major axis)/sigma|min**3 MAJOR AXIS

SKEWNESS ABOUT (3rd order moment about minor axis)/sigma|maj**3 MINOR AXIS

KURTOSIS ABOUT (4th order moment about major axis)/sigma|min**4 MINOR AXIS

KURTOSIS ABOUT (4th order moment about minor axis)/sigma|maj**4 MAJOR AXIS

HOLLOWS RATIO (area of hollows)/(area of bounding polygon)

Where sigma|maj**2 is the variance along the major axis and sigma|min**2 is the variance along the minor axis, and area of hollows= area of bounding polyarea of object

The area of the bounding polygon is found as a side result of the computation to find the maximum length. Each individual length computation yields a pair of calipers to the object orientated at every 5 degrees. The object is propagated into an image containing the union of these calipers to obtain an image of the bounding polygon.

NUMBER OF CLASSES : 4 OPEL, SAAB, BUS, VAN

NUMBER OF EXAMPLES :

Total no. = 946
No. in each class
opel 240 saab 240 bus 240 van 226
100 examples are being kept by Strathclyde for validation. So StatLog partners will receive 846 examples.

NUMBER OF ATTRIBUTES : No. of atts. = 18

BIBLIOGRAPHY :
Turing Institute Research Memorandum TIRM-87-018 "Vehicle Recognition Using Rule Based Methods" by Siebert,JP (March 1987)

8.17 Jeu de Données WINE

1. Title of Database: Wine recognition data
Updated Sept 21, 1998 by C.Blake : Added attribute information

2. Sources:
(a) Forina, M. et al, PARVUS - An Extendible Package for Data Exploration, Classification and Correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy.
(b) Stefan Aeberhard, email: stefan@coral.cs.jcu.edu.au
(c) July 1991

3. Past Usage:
(1) S. Aeberhard, D. Coomans and O. de Vel, Comparison of Classifiers in High Dimensional Settings, Tech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland. (Also submitted to Technometrics).
The data was used with many others for comparing various classifiers. The classes are separable, though only RDA has achieved 100% correct classification. (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data)) (All results using the leave-one-out technique)
In a classification context, this is a well posed problem with "well behaved" class structures. A good data set for first testing of a new classifier, but not very challenging.
(2) S. Aeberhard, D. Coomans and O. de Vel, "THE CLASSIFICATION PERFORMANCE OF RDA" Tech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland. (Also submitted to Journal of Chemometrics).
Here, the data was used to illustrate the superior performance of the use of a new appreciation function with RDA.

4. Relevant Information:
– These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.
– I think that the initial data set had around 30 variables, but for some reason I only have the 13 dimensional version. I had a list of what the 30 or so variables were, but a.) I lost it, and b.), I would not know which 13 variables are included in the set.

– The attributes are (dontated by Riccardo Leardi, riclea@anchem.unige.it)
1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10)Color intensity
11)Hue
12)OD280/OD315 of diluted wines
13)Proline

5. Number of Instances class 1 59 class 2 71 class 3 48

6. Number of Attributes 13

7. For Each Attribute:

All attributes are continuous

No statistics available, but suggest to standardise variables for certain uses (e.g. for us with classifiers which are NOT scale invariant)

NOTE: 1st attribute is class identifier (1-3)

8. Missing Attribute Values: None

9. Class Distribution: number of instances per class

class 1 59 class 2 71 class 3 48

8.18 Jeu de Données SPAM

SPAM E-MAIL DATABASE ATTRIBUTES (in .names format)

48 continuous real [0,100] attributes of type word-freq-WORD = percentage of words in the e-mail that match WORD, i.e. 100 * (number of times the WORD appears in the e-mail) / total number of words in e-mail. A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.

6 continuous real [0,100] attributes of type char-freq-CHAR = percentage of characters in the e-mail that match CHAR, i.e. 100 * (number of CHAR occurences) / total characters in e-mail

1 continuous real [1,...] attribute of type capital-run-length-average = average length of uninterrupted sequences of capital letters

1 continuous integer [1,...] attribute of type capital-run-length-longest = length of longest uninterrupted sequence of capital letters |

1 continuous integer [1,...] attribute of type capital-run-length-total = sum of length of uninterrupted sequences of capital letters = total number of capital letters in the e-mail

1 nominal 0,1 class attribute of type spam = denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.

For more information, see file 'spambase.DOCUMENTATION' at the UCI Machine Learning Repository: http://www.ics.uci.edu/ mlearn/MLRepository.html

classes 1, 0. (spam, non-spam)
word-freq-make: continuous.
word-freq-address: continuous.
word-freq-all: continuous.
word-freq-3d: continuous.
word-freq-our: continuous.
word-freq-over: continuous.
word-freq-remove: continuous.
word-freq-internet: continuous.
word-freq-order: continuous.
word-freq-mail: continuous.
word-freq-receive: continuous.
word-freq-will: continuous.
word-freq-people: continuous.
word-freq-report: continuous.
word-freq- addresses: continuous.
word-freq-free: continuous.
word-freq-business: continuous.
word-freq-email: continuous.
word-freq- you: continuous.
word-freq-credit: continuous.
word-freq- your: continuous.
word-freq-font: continuous.
word- freq-000: continuous.
word-freq-money: continuous.
word- freq-hp: continuous.
word-freq-hpl: continuous.
word-freq- george: continuous.
word-freq-650: continuous.
word-freq-lab: continuous.
word-freq- labs: continuous.
word-freq-telnet: continuous.
word-freq-857: continuous.
word- freq-data: continuous.
word-freq-415: continuous.
word- freq-85: continuous.
word-freq-technology: continuous.
word-freq-1999: continuous.
word-freq-parts: continuous.
word-freq-pm: continuous.
word-freq-direct: continuous.
word-freq-cs: continuous.
word-freq-meeting: continuous.
word-freq-original: continuous.
word-freq-project: continuous.
word-freq-re: continuous.
word-freq-edu: continuous.
word-freq-table: continuous.
word-freq-conference: continuous.
char-freq-;: continuous.
char-freq-(: continuous.
char-freq-[: continuous.
char-freq-!: continuous.
char-freq-: continuous.
char-freq-#: continuous.
capital-run-length-average: continuous.
capital-run-length- longest: continuous.
capital-run-length-total: continuous.

Bibliographie

[AB95] D.W. Aha and R.L. Bankert. A comparative evaluation of sequential feature selection algorithms, 1995.

[AD91] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, volume 2, pages 547–552, Anaheim, California, 1991. AAAI Press.

[AG92] H. Almuallim and Dietterich T. G. Efficient algorithms for identifying relevant features. Technical Report 92-30-03, 1992.

[AG94] H. Almuallim and Dietterich T. G. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305, 1994.

[Arr63] K.J. Arrow. *Social Choice and Individual Values*. 1963.

[Bat94] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. on Neural Networks*, 5:537–550, July 1994.

[BBMES03] J. Bi, K. P. Bennett, C. M. Breneman M. Embrechts, and M. Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research, Special Issue on Variable/Feature Selection*, 2003.

[BEF84] J.C. Bezdeck, R. Ehrlich, and W. Full. Fcm:fuzzy c-means algorithm. *Computers and Geoscience*, 1984.

[Ber02] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression trees, The Wadsworth Statistics/Probability Series, Wadsworth, Belmont, CA*. 1984.

[BFOS01] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. 2001.

[BGG+99] D. Boley, M. Gini, R. Gross, E. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitionning-based clustering for web document categorization. *Decision Support System*, pages 329–341, 1999.

[BLM86] J.P. Barthelemey, B. Leclerc, and B. Monjardet. On the use of ordered sets in problems of comparison and consensus classification. *Journal of Classification*, 3, 1986.

[Bre96a] L. Breiman. Arcing classifiers. *Annals of Statistics*, 1996.

[Bre96b] L. Breiman. Bagging predictors. *Machine Learning*, 1996.

[Car93] C. Cardie. Using decision trees to improve case–based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32. Morgan Kaufmann Publishers, Inc., 1993.

[CCM00] David Cohn, Rich Caruana, and Andrew McCallum. Semi-supervised clustering with user feedback (draft), 2000.

[CF94] Rich Caruana and Dayne Freitag. Greedy attribute selection. In *International Conference on Machine Learning*, pages 28–36, 1994.

[CS02] D. Cristofor and D. Simovici. An information-theoretical approach to clustering categorical databases using genetic algorithms. In *2nd SIAM ICDM, Workshop on clustering high dimensional data*, 2002.

[Dav96] R.N. Dave. Validating fuzzy partitions obtained through c-shells clustering. *Pattern Recognition Letters*, 17:613–623, 1996.

[DB79] D.L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2), 1979.

[DBE99] A. Demiriz, K. Bennett, and M. Embrechts. Semi-supervised clustering using genetic algorithms, 1999.

[DCSL02] Manoranjan Dash, Kiseok Choi, Peter Scheuermann, and Huan Liu. Feature selection for clustering - a filter solution. In *Proc. of International Conference on Data Mining (ICDM02)*, pages 115–122, 2002.

[DK82] P. A. Devijver and Kittler. *Pattern Recognition: A Statistical Approach, Englewood Cliffs, New Jersey: Prentice-Hall*. 1982.

[DL97] M. Dash and H. Liu. Feature selection for classification, 1997.

[DL03] Manoranjan Dash and Huan Liu. Feature selection for clustering. In *Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD03)*, pages 110–121, 2003.

[DLP82] Diday, Lemaire, and Pouget. Testu : Eléments d'analyse de données, dunod. 1982.

[DM91] R.L. De Mantaras. A distance-based attribute selection measure for decision tree induction. In *Machine Learning*, volume 6, pages 81–92, 6-9 1991.

[Doa92] J. Doak. An evaluation of feature selection methods and their application to computer security. In Department of Computer Science Davis, CA: University of California, editor, *Technical Report CSE-92-18*, 1992.

[Dom01] B. Dom. An information-theoretic external cluster-validity measure. Technical report, IBM, 2001.

[Dun74] J.C. Dunn. Well separated clusters and optimal fuzzy partitions. *J. Cybern.*, 4:95–104, 1974.

[EC02] Vladimir Estivill-Castro. Why so many clustering algorithms - a position paper. *SIGKDD Explorations*, 4:65–75, 2002.

[EKS+98] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Michael Wimmer, and Xiaowei Xu. Incremental clustering for mining in a data warehousing environment. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 323–333, 24–27 1998.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[FI92] U. M. Fayyad and K. B. Irani. The attribute selection problem in decision tree generation. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 104–110, 1992.

[Fis87] D. Fisher. Cobweb: Knowledge acquisition via conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[FPSS96] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *A.I. Magazine*, 17:37–54, 1996.

[FPSSU96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in knowledge discovery and data mining. *AAAI Press*, 1996.

[FRB98] U.M. Fayyad, C. Reina, and P.S. Bradley. Initialization of iterative refinement clustering algorithms. *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD98, AAAI Press*, 1998.

[Fre02] A.A. Freitas. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. 2002.

[Fri94] J.H. Friedman. An overview of predictive learning and function approximation. *From Statistics to Neural Networks, Proc. NATO/ASI Workshop, Springer-Verlag*, pages 1–61, 1994.

[FS96] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. *Machine Learning : proceedings of the Thirteenth International Conference*, 1996.

[FS97] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. *Journal of Computer and System Sciences*, 1997.

[GC85] M. A. Gluck and J. E. Corter. Information, uncertainty, and the utility of categories. *In Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 283–287, 1985.

[GD91] K. C. Gowda and Edwin Diday. Symbolic clustering using a new dissimilarity measure. *Pattern Recognition*, 24(6):567–578, 1991.

[GE03] Isabelle Guyon and Andre Eliseef Eds. *Journal on Machine Learning Research : Special Issue on Variable and Feature Selection*. 2003.

[Geu03] Pierre Geurts. Traitements de données volumineuses par ensembles d'arbres aléatoires. *Session Spéciale Entreposage et Fouille de Données, XXXVème Journées de la Société Francophone De Statistiques*, pages 111–122, 2003.

[Gha00] B. Ghattas. Agrégation d'arbres de classification. *Revue de Statistique Appliquée*, 2000.

[GKR00] David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Clustering categorical data: An approach based on dynamical systems. *VLDB Journal: Very Large Data Bases*, 8(3–4):222–236, 2000.

[Gra89] C. W. J. Granger. Combining forecasts, twenty years later. *Journal of Forecasting*, 1989.

[GRS98] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 73–84, 1998.

[GRS00] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000.

[GSCWS99] César Guerra-Salcedo, Stephen Chen, Darrell Whitley, and Stephen Smith. Fast and accurate feature selection using hybrid genetic strategies. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 177–184, Mayflower Hotel, Washington D.C., USA, 6-9 1999. IEEE Press.

[GSS02] J. Ghosh, A. Strehl, and Merugu S. A consensus framework for integrating distributed clusterings under limited knowledge sharing. *in Proc. of NSF Workshop on Next Generation Data Mining*, 2002.

[GV98] A.D. Gordon and M. Vichi. Partitions of partitions. *Journal of Classification*, (15):265–285, 1998.

[GW89] M. Grötschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45:59–96, 1989.

[Hal00a] Maria Halkidi. Quality assessment and uncertainty handling in data mining process. In *EDBT PhD Workshop*, 2000.

[Hal00b] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th International Conf. on Machine Learning*, pages 359–366. Morgan Kaufmann, San Francisco, CA, 2000.

[Har84] A. Hart. Experience in the use of an inductive system in knoowledge eng. In M. Bramer, editor, *Research and Development in Expert Systems*. Cambridge Univ. Press, Cambridge, MA,, 1984.

[HBV01] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information System*, 17(2–3):107–145, 2001.

[HBV02a] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Cluster validity methods: part i. *ACM SIGMOD Record*, 31(2):40–45, 2002.

[HBV02b] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Clustering validity checking methods: part ii. *ACM SIGMOD Record*, 31(3):19–27, 2002.

[HK01] J. Han and M. Kamber. Data mining : Concepts and techniques, morgan kaufmann publishers, usa. 2001.

[Hon94] S.J. Hong. Use of contextual information to feature ranking and discretization. In *IEEE Trans. On knowledge and Data Engeneering*, 1994.

[HT01] S. Hirano and Tsumoto. Indiscernibility degrees of objects for evaluating simplicity of knowledge in the clustering procedure. *IEEE International Conference on Data Mining (IEEE ICDM01*, 2001.

[HTO+02] S. Hirano, S. Tsumoto, T. Okuzaki, Y. Hata, and K. Tsumoto. Analysis of biochemical data aided by a rough sets-based clustering technique. *International Journal of Fuzzy Systems*, 4, 2002.

[Hua97] Zhexue Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *Research Issues on Data Mining and Knowledge Discovery*, 1997.

[HV01] Maria Halkidi and Michalis Vazirgiannis. Clustering validity assessment : Finding the optimal partioning of a data set. *Proceedings of ICDM01*, pages 187–194, 2001.

[JD88] A.K. Jain and R.C. Dubes. Algorithms for clustering data, englewood cliffs, nj: Prentice hall. 1988.

[JK99] E. Johnson and H. Kargupta. Collective, hierarchical clustering from distributed, heterogeneous data. *Zaki, M., and Ho, C., eds., Large-Scale Parallel KDD Systems, Springer-Verlag LNCS*, 1759, 1999.

[JKP94] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129, 1994. Journal version in AIJ, available at http://citeseer.nj.nec.com/13663.html.

[JKSK97] B.H. Jun, C.S. Kim, H.Y. Song, and J. Kim. A new criterion in selection and discretization of attributes for the generation of decision trees. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 19, pages 1371–1375, 1997.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[JN03a] Pierre Emmanuel Jouve and Nicolas Nicoloyannis. Classification non supervisée pour données catégorielles. In *Actes Session Spéciale des XXXVèmes Journées de Statistiques : Entreposage et Fouille de Données*, 2003.

[JN03b] Pierre Emmanuel Jouve and Nicolas Nicoloyannis. Classification non supervisée pour données catégorielles. In *Actes de XXXVèmes Journées de Statistiques*, 2003.

[JN03c] Pierre Emmanuel Jouve and Nicolas Nicoloyannis. Kerouac : an algorithm for clustering categorical data sets with practical advantages. In *Proc. of International Workshop on Data Mining for Actionable Knowledge (PAKDD03)*, 2003.

[JN03d] Pierre Emmanuel Jouve and Nicolas Nicoloyannis. A method for aggregating partitions, applications in knowledge discovery in databases. In *Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD03)*, 2003.

[JN03e] Pierre Emmanuel Jouve and Nicolas Nicoloyannis. A new method for combining partitions, applications for distributed clustering. In *International Workshop on Paralell and Distributed Machine Learning and Data Mining (ECML/PKDD03)*, 2003.

[JN03f] Pierre Emmanuel Jouve and Nicolas Nicoloyannis. The 'who is it ?' problem, application for customizable web sites. In *Proc. of Atlantic Web Intelligence Conference (AWIC'03)*, 2003.

[KARS97] G. Karypis, Aggarwal, V. R., Kumar, and S. Shekhar. Multilevel hypergraph partitioning : Application in vlsi domain. *Proc. of the Design & Automation Conf*, 1997.

[KC00] H. Kargupta and P. editors Chan. Advances in distributed and parallel knowledge discovery. aaai/mit press, cambridge, ma. 2000.

[Ken39] Kendall. A new measure of rank correlation. In *Biometrika N°30*, 1939.

[KL03] Youngok Kim and Soowon Lee. A clustering validity assessment index. *Proceedings of PAKKD03*, pages 602–608, 2003.

[Koh89] T. Kohonen. Self organizing memory. *3rd ed. Springer information sciences series, Springer Verlag, New-York*, 1989.

[Koh96] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page to appear, 1996.

[Kon94] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182, 1994.

[KR92a] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In MIT Press, editor, *Tenth National Conference on Artificial Intelligence*, pages 129–134, 1992.

[KR92b] K. Kira and L.A. Rendell. A practical approach to feature selection. In Morgan Kaufmann, editor, *Proceedings of the Tenth International Conference on Machine Learning*, 1992.

[KR02] J. Kittler and F. editors Roli. *Multiple Classifier Systems*, volume 2634. 2002.

[KS96] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996.

[KT88] Yves Kodratoff and G. Tecuci. Learning based on conceptual distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):897–909, 1988.

[LD01] Erel Levine and Eytan Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13(11):2573–2593, 2001.

[LEB02] Gaëlle Legrand, Walid Erray, and Marc Boullé. Un survey des méthodes de sélection d'attributs dans le data mining. In *Rencontres de la société française de classification, SFC02*, 2002.

[LJF02] M. Law, A.K. Jain, and M. Figueiredo. Feature selection in mixture-based clustering. In *Proc. of Neural Information Processing Systems - NIPS'2002*, 2002.

[LM98] H. Liu and H. Motoda. *Feature Extraction, Construction, and Selection : A Data Mining Perspective, Kluwer Academic, Boston, MA*. 1998.

[LMD98] Huan Liu, Hiroshi Motoda, and Manoranjan Dash. A monotonic measure for optimal feature selection. In *European Conference on Machine Learning*, pages 101–106, 1998.

[LMF02] Nada Lavrac, Hiroshi Motoda, and Tom Fawcett. First international workshop on data mining lessons learned (dmll-2002). *Nineteenth International Conference on Machine Learning (ICML-2002)*, 2002.

[LR00] S. Lallich and R. Rakotomalala. Fast feature selection using partial correlation for multi-valued attributes. In *Proceedings of the 4th European Conference on Knowledge Discovery in Databases, PKDD 2000*, pages 221–231, 2000.

[LS94] Pat Langley and Stephanie Sage. Oblivious decision trees and abstract cases. In *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning*. AAAI Press, 1994.

[LS95] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes, 1995.

[LS96] Huan Liu and Rudy Setiono. A probabilistic approach to feature selection - a filter solution. In *International Conference on Machine Learning*, pages 319–327, 1996.

[Mac67] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkley Symposium on Mathematical Statistics and Probability*, volume I : Statistics, pages 281–297, 1967.

[Mar84a] F. Marcotochino. Utilisation des comparaisons par paires en statistique des contingences, etude n°f-071. Technical report, Centre Scientifique IBM-France, Février 1984.

[Mar84b] F. Marcotochino. Utilisation des comparaisons par paires en statistique des contingences, partie ii etude n°f-071. Technical report, Centre Scientifique IBM-France, Mai 1984.

[Mic82] P. Michaud. Agrégation à la majorité 1 : Hommage à condorcet, etude n°f-051. Technical report, Centre Scientifique IBM-France, 1982.

[Mic83] Pierre Michaud. Opinions agregations. *New Trends in Data Analysis and Applications, North-Holland, Amsterdam, J. Janssen, J.F. Marcotorchino and J.M. Proth*, pages 5–27, 1983.

[Mic85] P. Michaud. Agrégation à la majorité 2 : Analyse du résultat d'un vote, etude n°f-051. Technical report, Centre Scientifique IBM-France, 1985.

[Mic87] Pierre Michaud. Condorcet - a man of the avant-garde. *Applied Stochastic Models and Data Analysis, Wiley Chichester, J. Janssen, J.F. Marcotorchino, J.M. Proth and P. Purdue*, 3(3), 1987.

[Mic91] Pierre Michaud. Simulated computation in automatic classification. *Proc. 2nd Symp. on High Performance Computing, M. Durand and F. El Dabaghi*, 1991.

[Mic97] Pierre Michaud. Clustering techniques. *Future Generation Computer Systems*, 13(2–3):135–147, November 1997.

[Min87] J. Mingers. Expert systems – rule induction with statistical data. In *Journal of the Operational Research Society*, 1987.

[Mir01] B. Mirkin. Reinterpreting the category utility function. *Machine Learning*, 42:219–228, 2001.

[ML94] A.W. Moore and M.S. Lee. Efficient algorithms for minimizing cross validation error. In *Proceedings of the 11th Interntional Conf. on Machine Learning*, 1994.

[MM81] F. Marcotorchino and P. Michaud. Heuristic approach to the similarity aggregation problem. *Methods of Operations Research*, 43:395–404, 1981.

[MM96] C. Merz and P. Murphy. Uci repository of machine learning databases. *http://www.ics.uci.edu/#mlearn/mlrepository.html*, 1996.

[Mod93] M. Modrzejewski. Feature selection using rough sets theory. In *Proceedings of the European Conference on Machine Learning*, 1993.

[MS83] R. S. Michalski and R. E. Stepp. Automated construction of classifications : Conceptual clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4):396–410, 1983.

[NF77] P.M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. In *IEEE Transactions Computers*, 1977.

[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.

[Nic88] N. Nicoloyannis. *Structures Prétopologiques et Classification Automatique*. PhD thesis, Université Lyon 1, 1988.

[NTT98] N. Nicoloyannis, M. Terrenoire, and D. Tounissoux. An optimisation model for aggregating preferences : A simulated annealing approach. *Health and System Science*, 2(1-2):33–44, 1998.

[PB97] N.R. Pal and J. Biswas. Cluster validation using graph theoretic concepts. *Pattern Recognition*, 30(6), 1997.

[PCS00] A. Prodromidis, P. Chan, and S. Stolfo. Meta-learning in distributed data mining systems: Issues and approaches. *H. Kargupta and P. Chan eds, Advances in Distributed & Parallel Knowledge Discovery. AAAI/MIT Press*, 2000.

[PSCF+89] Gregory Piatetsky-Shapiro, Jaime Carbonell, William Frawley, Kamran Parsaye, J. Ross Quinlan, Michael Siegel, and Ramasamy Uthurusamy. Workshop on knowledge discovery in databases. *Fourth International Joint Conference on Artificial Intelligence (IJCAI-1989)*, 1989.

[Qui86] J.R. Quinlan. Introduction of decision trees. In *Machine Learning*, volume 1, pages 81–106, 1986.

[Rak03] A. Rakotomamonjy. Variable selection using svm-based criteria. *Journal of Machine Learning Research, Special Issue on Variable/Feature Selection*, 2003.

[RLR98a] R. Rezaee, B.P.F. Lelieveldt, and J.H.C. Reiber. A new cluster validity index for the fuzzy c-mean. *Pattern Recognition Letters*, 19:237–246, 1998.

[RLR98b] R. Rezaee, B.P.F. Lelieveldt, and J.H.C. Reiber. A new cluster validity index for the fuzzy c-mean. *Pattern Recognition Letters 19*, (237–246), 1998.

[RRJ03] Riquelme J.C. Ruiz R. and Aguilar-Ruiz J.S. Projection-based measure for efficient feature selection. In *Journal of Intelligent and Fuzzy Systems*, 2003.

[SCZ98] C. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster : A multiresolution clustering approach for very large spatial database. *24th VLDB Conference, New York, USA*, 1998.

[SD03] H. Stoppiglia and G. Dreyfus. Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research, Special Issue on Variable/Feature Selection*, 2003.

[SG02a] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining partitionings. *Proc. of Conference on Artificial Intelligence (AAAI 2002), Edmonton, AAAI/MIT Press*, 2002.

[SG02b] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR), MIT Press*, 3, 2002.

[Sha48] C.E. Shannon. A mathematical theory of communication. In *Bell System Technical Journal*, 1948.

[Sha96] S.C. Sharma. Applied multivariate techniques, john willey & sons. 1996.

[SJL90] Niblack W. Sheinvald J., Dom B. and Rendell L.A. A modeling approach to feature selection. In *10th International Conf. on Pattern Recognition*, 1990.

[Smy96] P. Smyth. Clustering using monte carlo cross-validation. *Proceedings of KDD, Conference*, 1996.

[Str02] A. Strehl. Relationship-based clustering and cluster ensembles for high-dimensional data mining. *Phd Thesis,University of Texas at Austin*, 2002.

[TK99] S. Theodoridis and K. Koutroubas. Pattern recognition. *Academic Press*, 1999.

[VDJ93] H. Vafaie and K. De Jong. Robust feature selection algorithms. In *Proceedings of the Fifth Conference on Tools for Artificial Intelligence*, pages 356–363, 1993.

[VDJ94] H. Vafaie and H. De Jong. Improving a rule induction system using genetic algorithms. 1994.

[VHD03] Michalis Vazirgiannis, Maria Halkidi, and Gunopulos Dimitrious. *Uncertainty Handling and Quality Assessment*, volume IX of *in Data Mining Series: Advanced Information and Knowledge Processing*. SPRINGER VERLAG, 2003.

[VJ92] H. Vafaie and K. De Jong. Genetic algorithms as a tool for feature selection in machine learning. 1992.

[Wat85] S. Watanabe. Pattern recognition : Human and mechanical. *John Wiley and Sons, Inc., New York*, 1985.

[WL59] W.T. Williams and J.M. Lambert. Multivariate methods in plant ecology. *Journal of Ecology*, 47:83–101, 1959.

[WYM97] Wei Wang, Jiong Yang, and Richard 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.

[WYM99] W. Wang, J. Yang, and R Muntz. STING+: An approach to active spatial data mining. In *Fifteenth International Conference on Data Engineering*, pages 116–125, Sydney, Australia, 1999. IEEE Computer Society.

[XB91] X. L Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4), 1991.

[YH98] Jihoon Yang and Vasant Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13:44–49, 1998.

[YPH97] J. Yang, R. Parekh, and V. Honavar. Distal: An inter-pattern distance-based constructive learning algorithm, 1997.

[ZD91] X. Zhou and T.S. Dillon. A statistical–heuristic feature selection criterion for decision tree induction. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13, pages 834–841, 1991.

[Zha64] C.T. Zhan. Approximating symetric relation by equivalence relation. In *SIAM Journal of Applied Mathematics,*, volume 12, 1964.

[ZKV94] D.A. Ziani, Z. Khalil, and R. Vignes. Recherche de sous-ensembles minimaux de variables à partir d'objets symboliques. In *IPMU'94*, volume 2, pages 794–799, 1994.

[ZRL96] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: an efficient data clustering method for very large databases. In *ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, Canada, June 1996.

[ZRR98] A. Zighed, S. Rabaséda, and R. Rakotomalala. Fusinter : a method for discretization of continuous attributes for supervised learning. *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 6(33):307–326, 1998.

Table des figures

Liste des tableaux