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

Table des matières

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

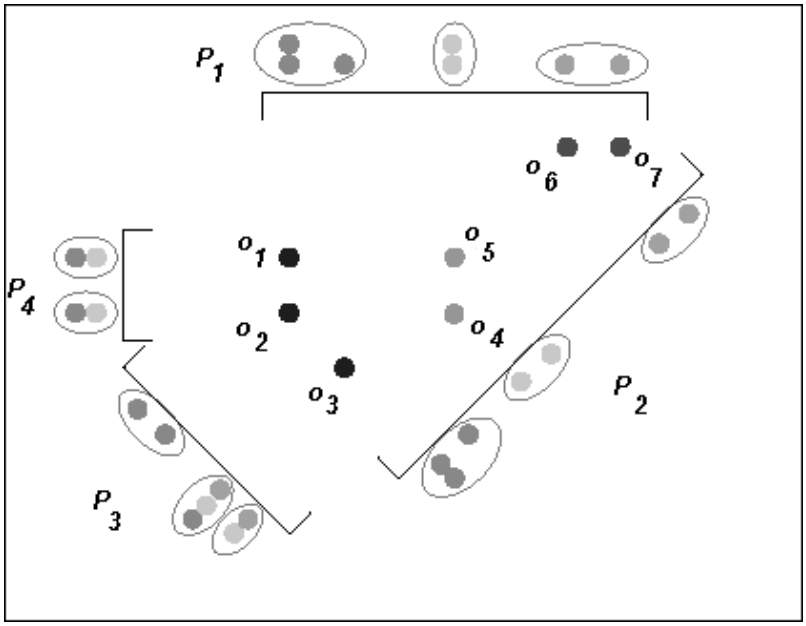

FIG. 6.1 –: *Illustration de la Problématique "Cluster Ensembles"*

(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*

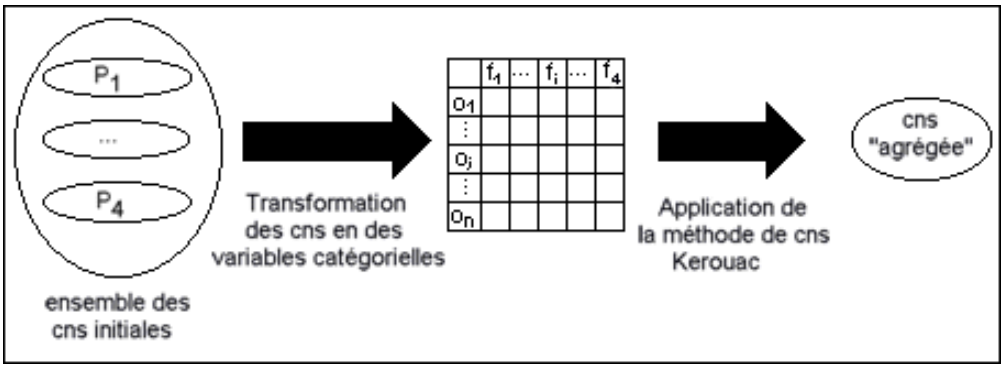

FIG. 6.2 –: *Utilisation de* KEROUAC *pour la problématique "Cluster Ensembles"*

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.

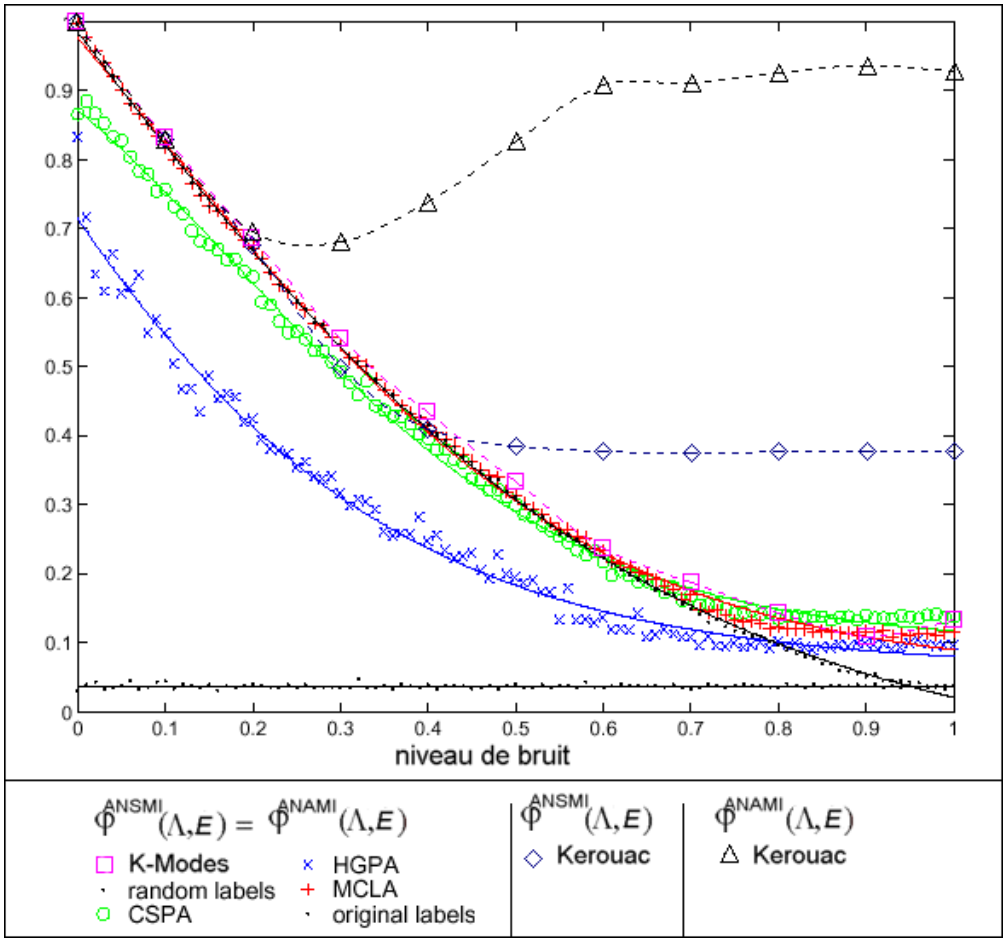

FIG. 6.3 –: *Evaluation de l'information mutuelle symétrique (resp. asymétrique) normalisée moyenne entre chacune des cns résultant d'un processus d'agrégation et l'ensemble des r cns dont elle constitue l'agrégation (évaluation de $\varphi^{ANSMI}(\Lambda,E)$ (resp. $\varphi^{ANAMI}(\Lambda,E)$))*

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

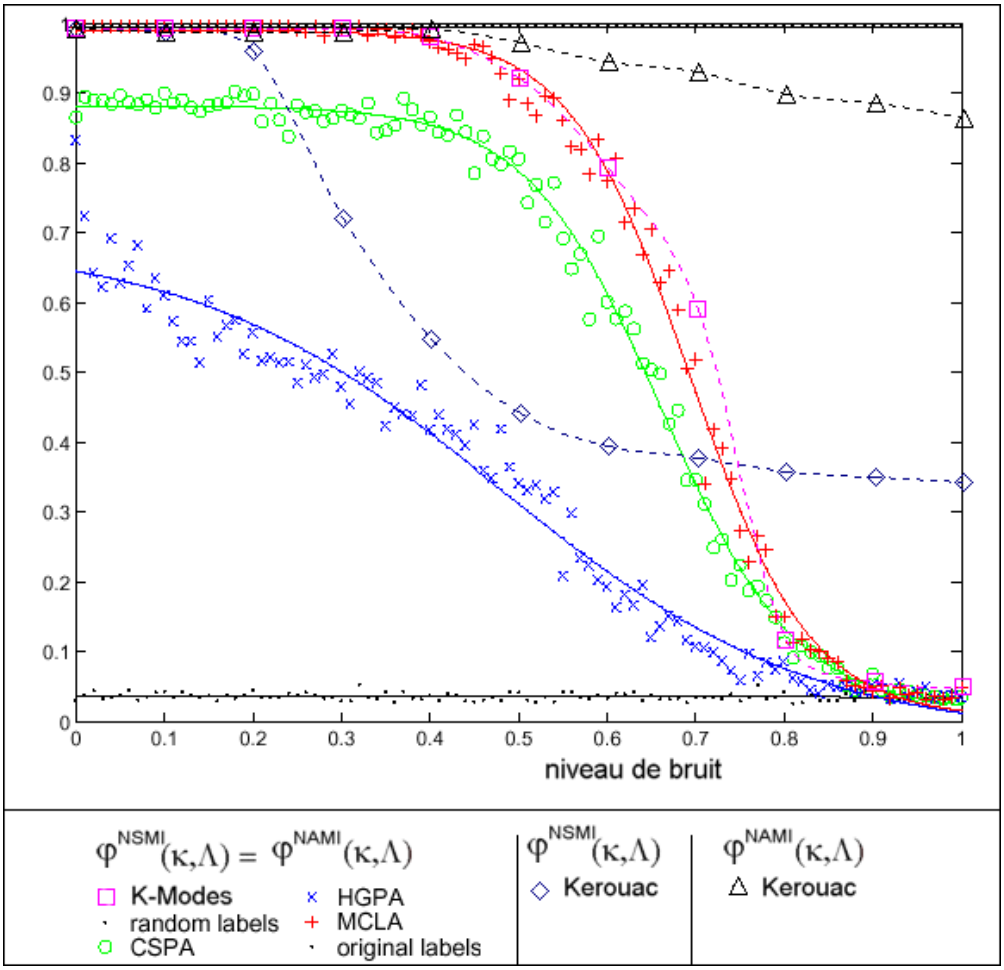

FIG. 6.4 –: *Evaluation de l'information mutuelle symétrique (resp. asymétrique) normalisée entre chacune des cns résultant d'un processus d'agrégation et la cns κ (évaluation de $\varphi^{NSMI}(\kappa,\Lambda)$ (resp. $\varphi^{NAMI}(\kappa,\Lambda)$))*

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

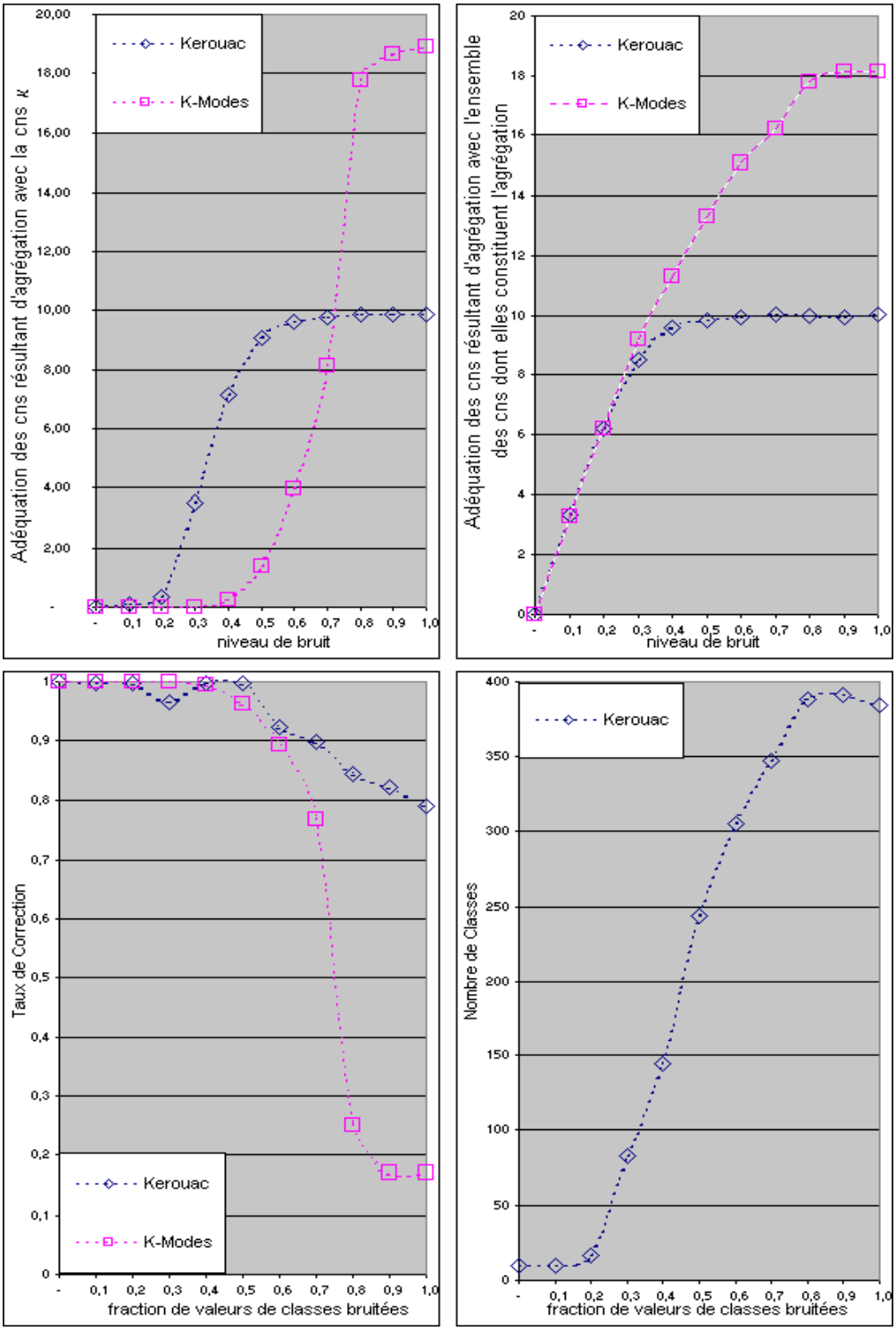

FIG. 6.5 –: *Evaluation de l'adéquation entre chacune des cns résultant d'un processus d'agrégation et l'ensemble des r cns dont elle constitue l'agrégation (évaluation de $Adq(\Lambda,E)$ et $Adq(\kappa,\Lambda)$)*

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

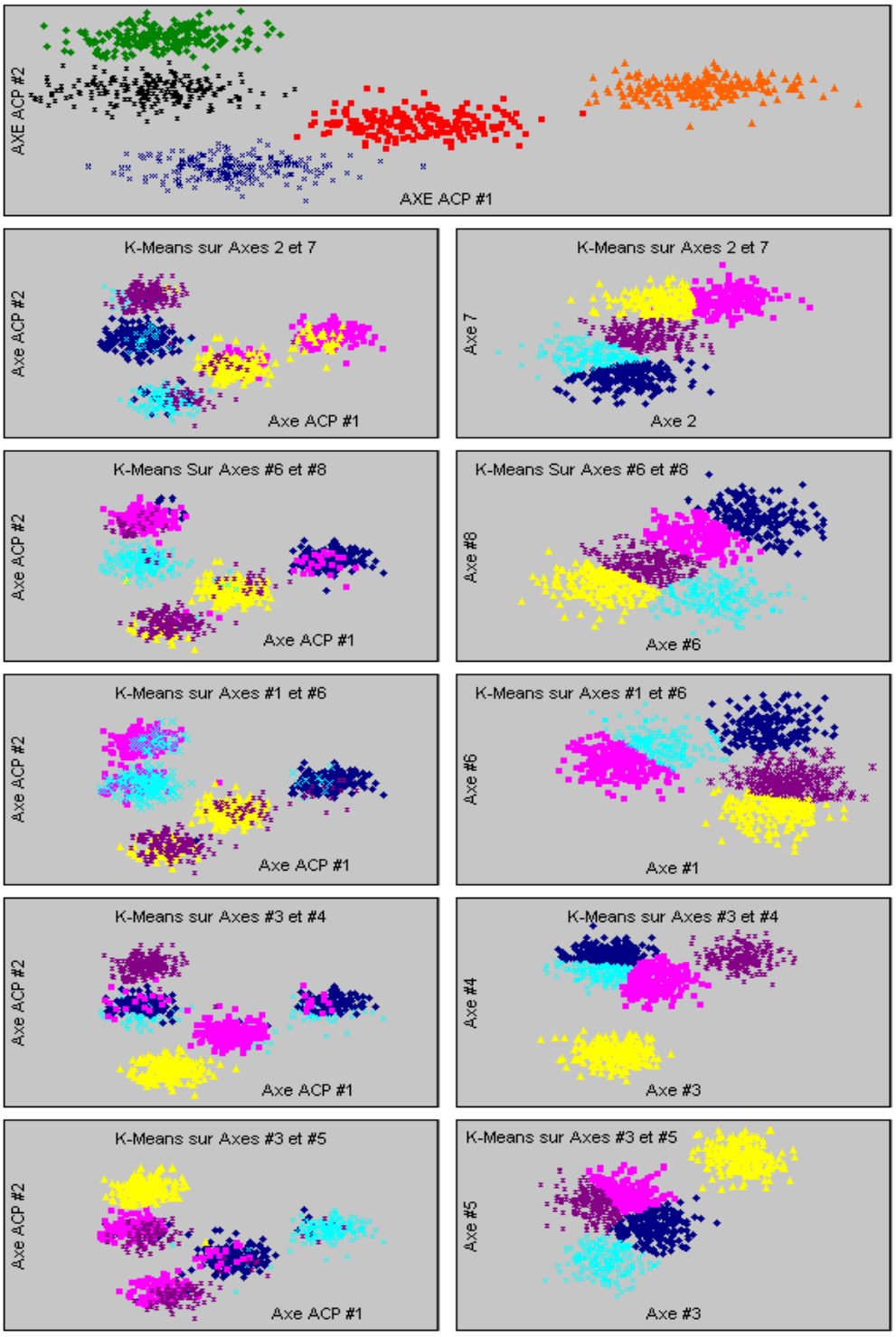

FIG. 6.6 –: *Scénario **DDV** : Expérience sur le jeu de données 8D5K*

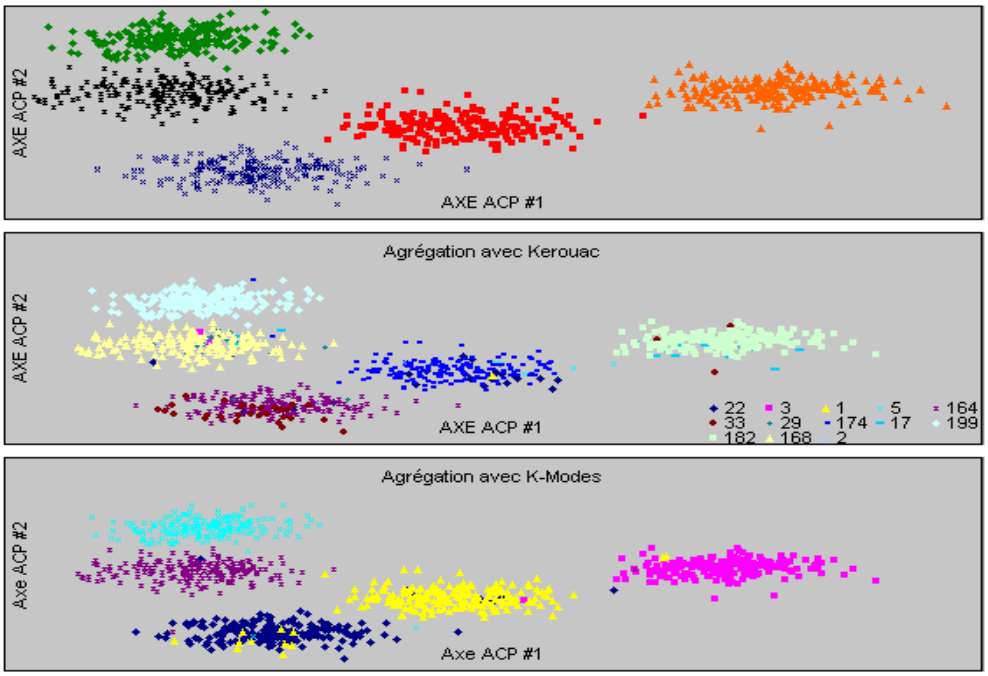

FIG. 6.7 –: *Scénario **DDV** : Expérience sur le jeu de données 8D5K*

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...

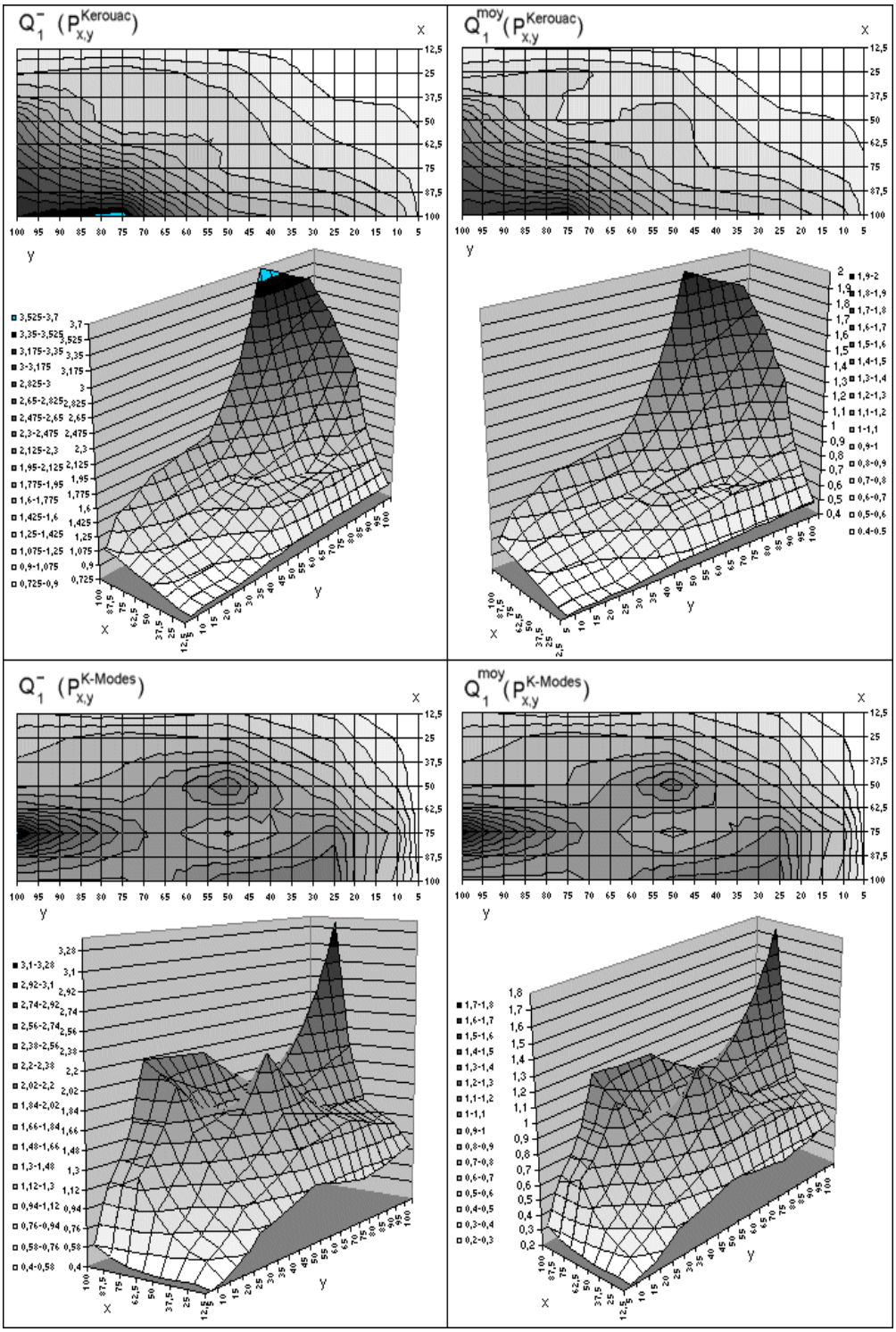

FIG. 6.8 –: *Scénario* ***DDOV****: Evaluation de la qualité des cns issues de l'agrégation, Indice* Q_1

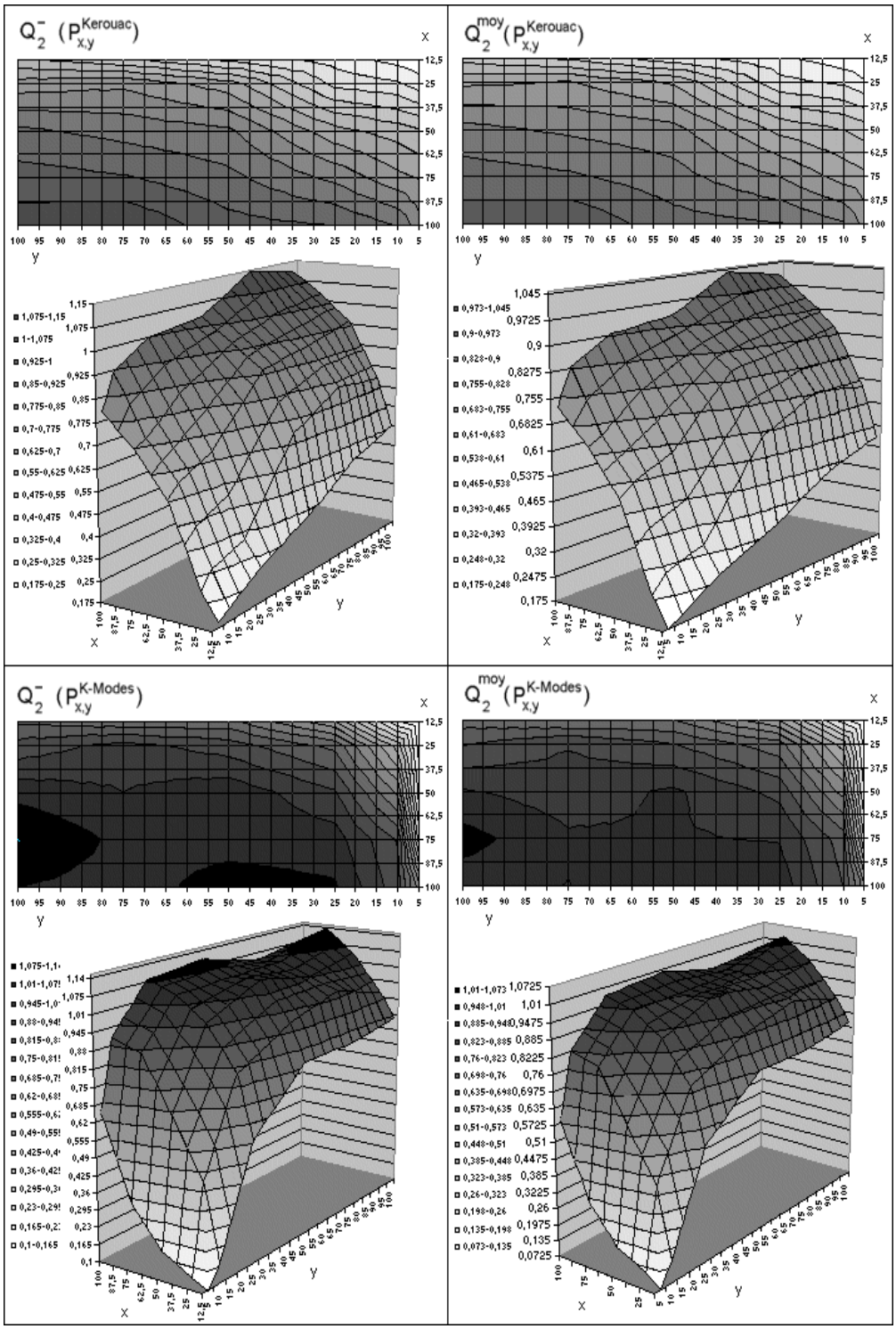

FIG. 6.9 –: *Scénario **DDOV**: Evaluation de la qualité des cns issues de l'agrégation, Indice Q_2*

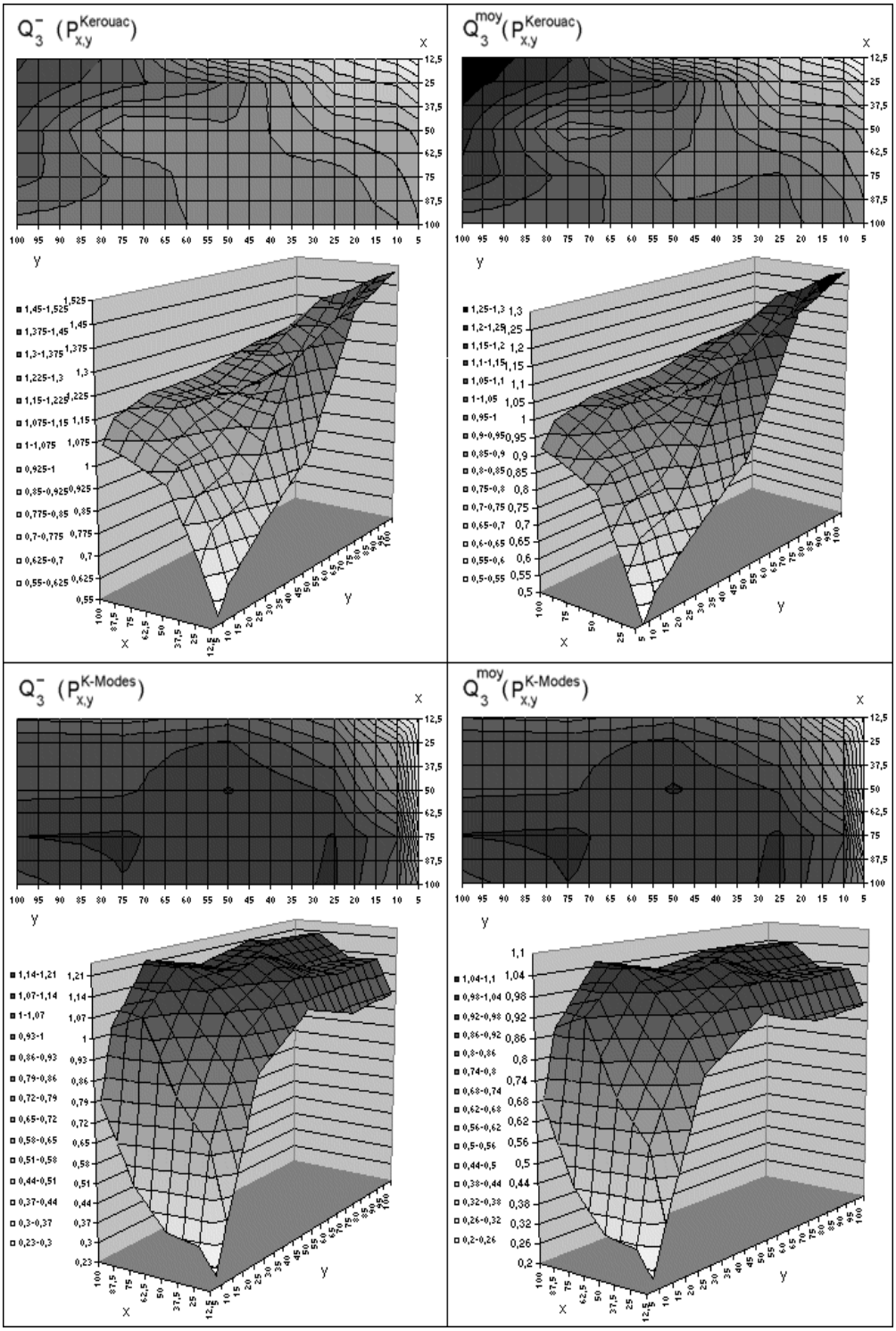

FIG. 6.10 –: *Scénario* ***DDOV****: Evaluation de la qualité des cns issues de l'agrégation, Indice* Q_3

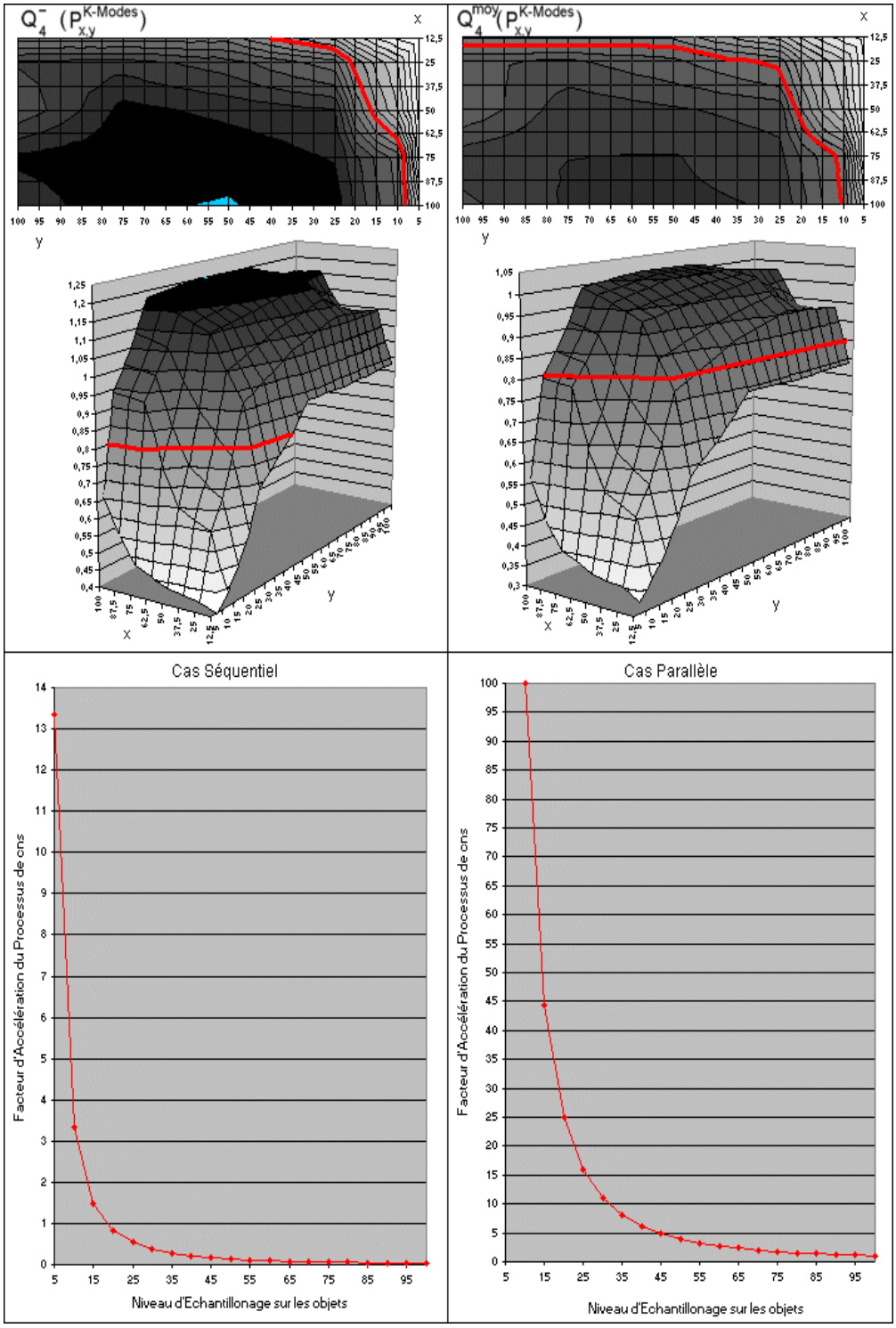

FIG. 6.11 –: *Scénario **DDOV**: Evaluation de la qualité des cns issues de l'agrégation, Indice Q_4; et Facteurs d'accélération du processus de cns*

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

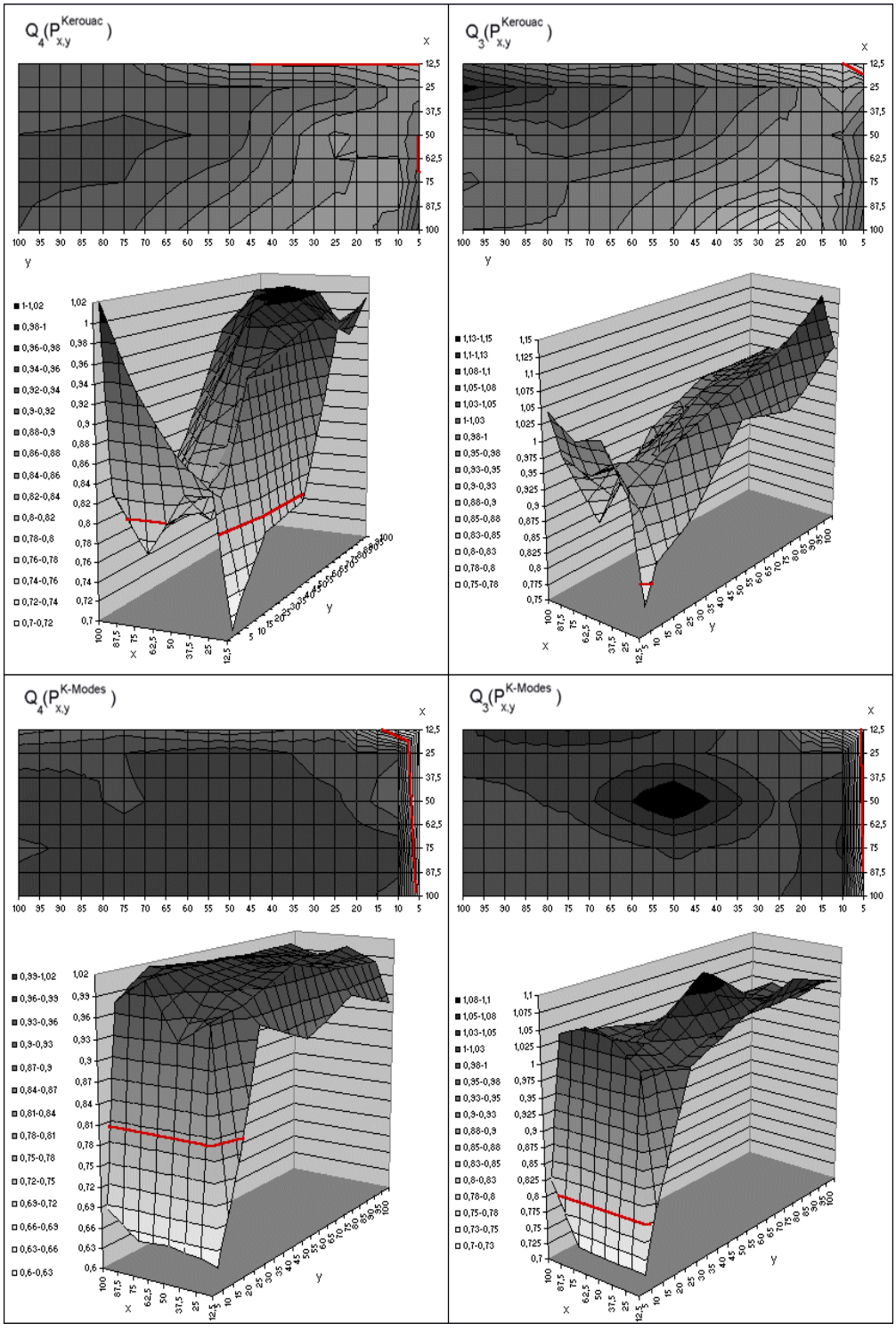

FIG. 6.12 –: *Evaluation de la qualité des cns issues de l'agrégation, Indice Q_4 et Q3 (jeu de données "1984 United States Congressional Voting Records Database")*

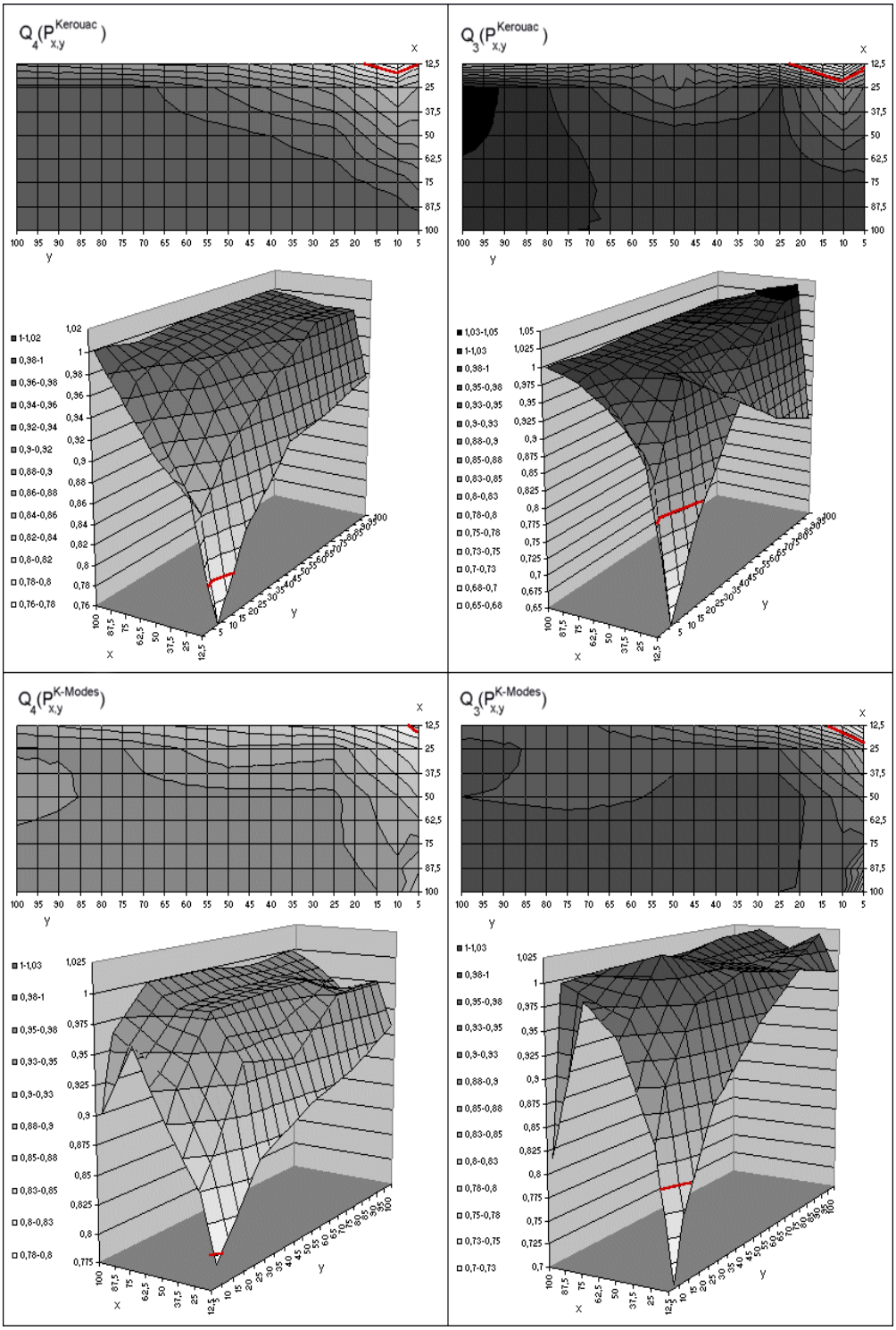

FIG. 6.13 –: *Evaluation de la qualité des cns issues de l'agrégation, Indice Q_4 et $Q3$ (jeu de données "mushrooms")*

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.

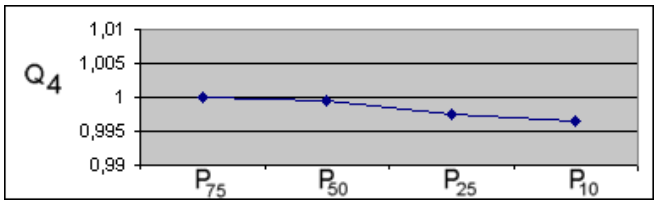

FIG. 6.14 –: *Indice Q_4 pour les 4 cns résultant d'agrégation*

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.

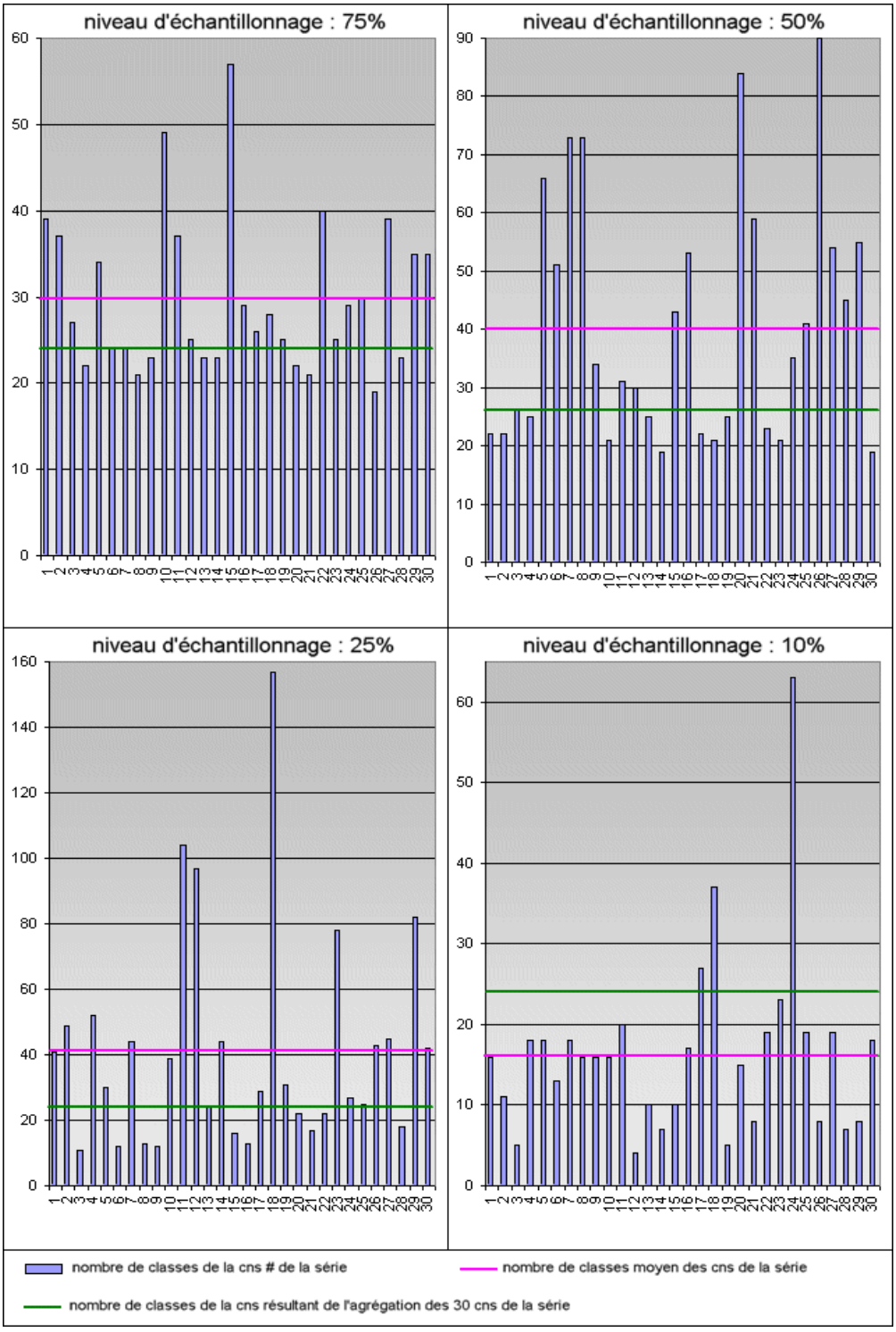

FIG. 6.15 –: *Nombre de classes pour les cns à agréger et les cns résultant d'agrégation*

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)...