HAL open science

Classification non supervisée : de la multiplicité des données à la multiplicité des analyses

Jacques-Henri Sublemontier

To cite this version:

Jacques-Henri Sublemontier. Classification non supervisée : de la multiplicité des données à la multiplicité des analyses. Apprentissage [cs.LG]. Université d'Orléans, 2012. Français. NNT : . tel-00801555v1

HAL Id: tel-00801555

https://theses.hal.science/tel-00801555v1

Submitted on 17 Mar 2013 (v1), last revised 11 Jul 2013 (v2)

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

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

UNIVERSITÉ D'ORLÉANS

ÉCOLE DOCTORALE MIPTIS

Laboratoire d'Informatique Fondamentale d'Orléans

THÈSE

présentée par :

Jacques-Henri SUBLEMONTIER

soutenue le : **vendredi 07 décembre 2012**

pour obtenir le grade de : **Docteur de l'Université d'Orléans**

Discipline/Spécialité : **Informatique**

Classification non supervisée : de la multiplicité des données à la multiplicité des analyses

THÈSE DIRIGÉE PAR :

Christel VRAIN	Professeur des Universités, Université d'Orléans

RAPPORTEURS :

Younès BENNANI	Professeur des Universités, Université de Paris XIII
Yves LECHEVALLIER	Directeur de Recherche INRIA, Rocquencourt

JURY :

Pierre GANÇARSKI	Professeur des Universités, Université de Strasbourg
Stéphane LALLICH	Professeur des Universités, Université de Lyon II
Younès BENNANI	Professeur des Universités, Université de Paris XIII
Yves LECHEVALLIER	Directeur de Recherche INRIA, Rocquencourt
Christel VRAIN	Professeur des Universités, Université d'Orléans
Guillaume CLEUZIOU	Maître de conférence, Université d'Orléans (encadrant)
Lionel MARTIN	Maître de conférence, Université d'Orléans (encadrant)

Remerciements

Un petit pas pour l'homme, un grand pas pour l'humanité.

Que de pédanterie si je tenais là un tel discours :), mais il n'est pas question de cela ici. Mais quelle sensation agréable que d'achever l'œuvre et atteindre l'aboutissement de toute une vie d'étudiant de l'enseignement supérieur. Le manuscrit de thèse est l'apport si modeste soit-il d'une âme conquérante ayant voulu défier les démons de la Science et persistant encore à les affronter.

un grand pas pour l'homme, un petit pas pour l'humanité !

J'adresse à travers ces quelques paragraphes mes plus grands remerciements envers toutes les personnes : rencontrées, retrouvées, proches, amis, collègues de travail, qui ont contribué à mon épanouissement durant toutes ces années de travail.

Tout d'abord, un grand remerciement à mes rapporteurs Younès Bennani et Yves Lechevallier, pour le temps et l'effort qu'ils ont soigneusement consacré à l'évaluation de mon travail. Les commentaires émis ont contribué à l'amélioration du manuscrit ainsi qu'au maintien voire au renforcement de ma motivation. Je remercie également les examinateurs de mon jury Pierre Gançarski et Stéphane Lallich pour les remarques pertinentes et constructives adressées lors de la soutenance. Elles auront également permis de soigner la qualité de la présente étude.

Tout doctorant, «thésard» tentant de maintenir sa barque dans les remous de la Recherche nécessite de ne jamais perdre l'Étoile du Nord. Je remercie mes encadrants, Guillaume Cleuziou et Lionel Martin ainsi que ma directrice de thèse Christel Vrain, d'avoir tenu ce rôle. Sans vous, le petit œuf n'aurait peut-être pas vu le jour, et n'aurait sans doute jamais été conçu !

Je remercie sincèrement également l'ensemble de mes collègues de travail, extérieurs ou locaux et parmi eux notamment les directeurs successifs du LIFO Christel Vrain encore une fois, puis Jérôme Durand-Lose, pour leurs accompagnements, administratifs et humains à la recherche scientifique. Merci également aux responsables pédagogiques Ali Ed-Dbali et Catherine Julié-Bonnet, pour l'accompagnement cette fois pédagogique d'un humble moniteur et ATER. Pour finir, merci également à vous mes co-bureaux Sylvie, Yannick, André et Irénée. Que vous puissiez retrouver en ces quelques lignes l'entente amicale et chaleureuse que nous avons pu avoir tout au long de ces années.

Quant aux plus jeunes, compagnons de croisade scientifique ou de jeux, futurs-présents-ex doctorants, ATER et post-docs, merci à vous pour les années passées ensemble ;) Les souvenirs les plus importants, ceux qui restent, ne sont jamais les souvenirs du boulot. À vous tous : Jérémie, Matthieu, Julien, Mathieu, Maxime, Joeffrey, Ahmed, Hélène, Simon, Romain, Nicolas, Bastien, Thomas, Simon 2, Mouhamadou, Abderrahim, Sylvain, Irénée, Anthony, Vincent, Claire, Abir, Pierre, Julie, Thang, Chuong, Do, Tung. Pfiou... que les involontaires oubliés me pardonnent ! merci pour toutes les soirées BSS (nan, les verres de saké ne sont que pour ceux qui gagnent les parties de Bomberman : trouver l'équilibre de Nash il faut ;)), Bar (où ils servent de la Guiness en

pression hein ?), jeux (plateaux, société), sorties ciné (ce soir ?) et séances sportives (ping-pong, tennis, foot, basket, badminton, volley...) que nous avons fait ensemble et pour les sentiments que nous avons échangés tout au long de ces années.

À vous aussi les sportifs, ex-coéquipiers du Tennis de Table et du Tennis, à vos soutiens moraux et extérieurs réconfortants.

Enfin, les survivants ayant la possibilité d'atteindre les dernières lignes auront les plus gros câlins !!! c'est ainsi tout naturellement que mes sentiments les plus chers vont aux plus proches... À ma famille et ma belle-famille, à vous les soutiens sans failles Laurent, Marie-Aure, Céline, Arnaud, aux anciens du début Issa, Damien et aux retrouvés (vous vous reconnaîtrez).

À toi Marion, véritable pilier indestructible de notre couple, puissent ces derniers mots te témoigner mes plus hauts sentiments, au réconfort que tu m'as toujours apporté, aux sacrifices que tu as fait, au profond respect que j'ai pour toi.

Sommaire

Introduction **9**

Chapitre 1 : Classification non supervisée **25**

1.1 Introduction . . . 26
1.2 Approches hiérarchiques . . . 27
1.2.1 DIANA : *DIvisive ANAlysis* . . . 27
1.2.2 AGNES : *AGglomerative NESted clustering* . . . 28
1.3 Approches partitives . . . 30
1.3.1 Approches basées sur les prototypes . . . 30
1.3.1.1 KM : les K-moyennes . . . 30
1.3.1.2 SC : *clustering* spectral . . . 32
1.3.2 Approches basées sur la densité . . . 34
1.3.2.1 DBSCAN : *clustering* basé sur la densité . . . 34
1.3.2.2 SOM : les cartes auto-organisatrices . . . 35
1.4 Approches floues et probabilistes . . . 37
1.4.1 FKM : les K-moyennes floues . . . 37
1.4.2 EM : estimation d'un mélange de modèles par Espérance-Maximisation . . . 39
1.5 Bilan . . . 41
1.5.1 Les liens entre familles d'algorithmes de *clustering* . . . 41
1.5.2 Le problème du nombre de groupes . . . 42
1.5.3 Le problème de l'évaluation . . . 43
1.5.3.1 Mesures basées sur l'énumération . . . 44
1.5.3.2 Mesures statistiques basées sur l'entropie. . . . 45
1.5.4 Le choix de la proximité . . . 46
1.5.5 Le choix de l'algorithme . . . 47

Chapitre 2 : Classification non supervisée multi-vues centralisée **49**

2.1 Introduction . . . 50
2.2 Contexte . . . 50
2.3 Approches centralisées . . . 52
2.3.1 MVDBSCAN : DBSCAN multi-vues . . . 53
2.3.2 COFC : *clustering* flou collaboratif . . . 54
2.3.3 FCPU : *clustering* flou dans les univers parallèles . . . 56
2.3.4 MVADASOM : SOM multi-vues *via* les distances adaptatives . . . 58
2.3.5 COMRAF*: champs aléatoires combinatoires de markov . . . 61
2.3.6 COEM : estimation d'un modèle de mélange pour données multi-vues . . . 63
2.4 Contributions . . . 66
2.4.1 Motivation . . . 66
2.4.2 COFKM : *clustering* flou multi-vues . . . 66

2.4.3 CoKFKM : *clustering* flou multi-vues à noyaux . . . 73
2.5 Évaluation . . . 77
2.5.1 Données . . . 78
2.5.2 Protocole expérimental . . . 78
2.5.3 Évaluation interne . . . 79
2.5.4 Évaluation externe . . . 80
2.6 Discussion . . . 87
2.7 Conclusion . . . 87

Chapitre 3 : Classification non supervisée et intégration de connaissances 89
3.1 Introduction . . . 90
3.2 Contexte . . . 90
3.3 Approches par satisfaction des contraintes . . . 92
3.3.1 Cop-KMeans : les K-moyennes sous contraintes . . . 92
3.3.2 CCHC : *clustering* semi-supervisé hiérarchique en lien complet . . . 94
3.3.3 SSEM : estimation d'un mélange de modèle semi-supervisé . . . 95
3.4 Approches par objectif pénalisé . . . 98
3.4.1 PCKM : les K-moyennes contraintes pénalisées . . . 98
3.4.2 SSKM : les K-moyennes semi-supervisées . . . 100
3.5 Approches par altération de la proximité . . . 101
3.5.1 LLMA : adaptation localement linéaire de la métrique . . . 101
3.6 Approches indépendantes de l'algorithme de *clustering* . . . 104
3.6.1 BC : *BoostCluster* . . . 104
3.7 Contributions . . . 106
3.7.1 Motivation . . . 106
3.7.2 BoC : *boosting* de *clustering* . . . 109
3.7.3 UzaBoC et AdaUzaBoC : *boosting* simple et adaptatif de *clustering* par optimisation . . . 117
3.8 Évaluation . . . 123
3.8.1 Données . . . 123
3.8.2 Protocole expérimental . . . 124
3.8.3 Évaluation interne . . . 126
3.8.4 Évaluation externe . . . 134
3.9 Discussion . . . 142
3.10 Conclusion . . . 143

Chapitre 4 : Classification non supervisée collaborative 145
4.1 Introduction . . . 146
4.2 Contexte . . . 147
4.3 Approches de type ensemble de *clusterings* . . . 149
4.3.1 *Clustering* consensus par ensemble de *clusterings* . . . 149
4.3.2 Consensus de partitions . . . 151
4.4 Approches collaboratives . . . 154
4.4.1 SAMARAH : système d'apprentissage multi-agents de raffinement automatique de hiérarchies . . . 154
4.4.2 MOCLE : *clustering* d'ensemble multi-objectif . . . 156
4.5 Approches alternatives . . . 158
4.5.1 COALA : *clustering* hiérarchique alternatif . . . 158

4.5.2 ADFT : apprentissage de distance alternative 160
4.5.3 CAMI : estimation d'un mélange de modèles alternatifs 161
4.6 Contributions 163
4.6.1 Motivation 163
4.6.2 CoBoC : *boosting* collectif et collaboratif pour la recherche de consensus . 166
4.6.3 AlterBoC : *boosting* collectif et collaboratif pour la recherche d'alternatives 170
4.7 Évaluation 172
4.7.1 Protocole expérimental 173
4.7.2 Évaluation interne 174
4.7.3 Évaluation externe 183
4.8 Discussion 205
4.9 Conclusion 207

Conclusion et perspectives **209**

Liste des tableaux **213**

Table des figures **216**

Liste des algorithmes **218**

Bibliographie **221**

Introduction

La classification non supervisée, ou *clustering*, est un thème de recherche majeur en apprentissage automatique, en analyse et en fouille de données ainsi qu'en reconnaissance de formes. Il fait partie intégrante de tout un processus d'analyse exploratoire de données permettant de produire des outils de synthétisation, de prédiction, de visualisation et d'interprétation d'un ensemble d'individus (personnes, objets, processus, *etc.*). L'objectif est, à partir de données constituées d'un ensemble d'individus ou objets et d'une relation de proximité entre ceux-ci, de construire des groupes d'individus homogènes dans le sens où :

- deux individus proches doivent appartenir à un même groupe ;
- deux individus éloignés doivent appartenir à des groupes différents.

Cette introduction a pour but de présenter de manière informelle la problématique à travers différents problèmes survenant lors de l'élaboration de techniques visant à en apporter des solutions. Ainsi seront présentés les fondements de toute approche de *clustering*, les données, la proximité ainsi que les différents moyens de construire des groupes. Dans un second temps, par l'intermédiaire d'applications, des problématiques spécifiques seront développées, problématiques pour lesquelles ce travail de thèse propose des éléments de réponse.

Classification non supervisée

Les données

Le *clustering*, dans sa forme la plus classique, repose essentiellement et tout d'abord sur des données. Ces données sont constituées d'un ensemble d'*individus* associées à une *représentation*.

FIGURE 0.1 — Données désordonnées avant *clustering* (à gauche) et ordonnées après *clustering* (à droite).

Dans l'exemple présenté en figure 0.1 correspondant à la photo du bureau légèrement désordonné d'un doctorant en fin de thèse, plusieurs objets ou individus y sont disposés : livres, articles de recherche, cours, fournitures de bureau, *etc.* Chacun de ces individus peut être muni d'une représentation liée à notre perception visuelle de ceux-ci. On peut ainsi associer des attributs de : largeur, longueur, épaisseur, couleur, présence ou non de texte, présence de dessins, forme primitive (parallélépipède rectangle, ellipse, cylindre, *etc.*). Ainsi, nous pouvons dresser une table non exhaustive des différents objets et de leurs propriétés.

identifiant	largeur	longueur	épaisseur	couleur	texte	dessins	forme
Livre 1	19	23	2,5	Noir	1	1	Rect.
Livre 2	16	24	2	Rouge	1	1	Rect.
Livre 3	14	22	1,3	Blanc	1	1	Rect.
Article 1	29,8	21	0,2	Blanc	1	0	Rect.
Article 2	29,8	21	0,2	Blanc	1	1	Rect.
Peluche 1	18	20	13	Roux	0	0	Ellipt.
Peluche 2	14	25	40	Camel	0	0	Ellipt.
Ciseaux	8	22	1	Argent	0	0	Triang.
Tasse	11	11	7,5	Multi.	0	1	Cylind.
Crayon	0,7	17	0,7	Vert	0	0	Cylind.

Nous pouvons noter que de premiers problèmes interviennent, notamment en ce qui concerne l'intervalle des valeurs que peut prendre une propriété donnée pour un individu :

- celui-ci est-il fini (ensemble des dénominations de couleurs, ensemble des valeurs flottantes mesurables exactement avec une règle) ou infini (ensemble des valeurs réelles) ? un individu est-il rouge, ou à 85,67% rouge ? on parle de domaine de valeurs discret dénombrable et de domaine indénombrable ou continu ;
- un individu est-il noir, ou noir et orange, mais à dominante noir ?, à 80% noir et 20% orange ? on parle de mono-valuation ou de multi-valuation.

La proximité

La représentation permet de comparer différents profils d'individus. Cette idée de comparer les individus correspond au second fondement clé de tout algorithme destiné à une tâche de classification automatique : la *mesure de proximité*. À partir de la représentation construite dans l'exemple, nos pouvons réfléchir à une notion de proximité entre deux individus.

Dans la grande majorité des cas, nous considérons que deux individus sont proches si, pour chaque propriété présente dans la représentation, les valeurs de cette propriété pour ces individus sont proches. Par exemple, ici, nous pouvons considérer que les trois livres sont proches au sens de leur forme primitive, et que les ciseaux et le livre 3 sont proches au sens de l'épaisseur. Nous pouvons faire l'hypothèse que l'ensemble des propriétés présentes dans la représentation ne sont pas toutes utiles, ou certaines, moins que les autres. Cette hypothèse est présente dans de nombreux travaux actuels en classification non supervisée, et vise à sélectionner les propriétés les plus pertinentes, ou bien à réduire l'importance de celles qui sont les moins utiles ou informatives. Quoi qu'il en soit, certaines combinaisons particulières de propriétés permettent d'identifier plus facilement un groupe d'individus relativement aux autres, et peuvent avoir une sémantique bien particulière ; ils portent le nom de concept. Ici un concept serait celui de volume comprenant à la fois les propriétés de longueur, de largeur, et d'épaisseur.

La mesure de proximité peut prendre diverses formes mais elle permet toujours de quantifier la ressemblance ou dissemblance entre les individus. Ainsi, dans l'exemple, nous pouvons attribuer une valeur numérique de proximité entre deux individus, en réalisant la somme des écarts en valeur absolue des valeurs numériques pour les propriétés du concept volume :

$$\begin{aligned} proximité(\text{Objet 1}, \text{Objet 2}) &= |val_1(longueur) - val_2(longueur)| \\ &+ |val_1(largeur) - val_2(largeur)| \\ &+ |val_1(épaisseur) - val_2(épaisseur)| \end{aligned}$$

où $val_i(P)$ correspond à la valeur prise par l'objet i pour la propriété P. De cette mesure, on peut déduire l'application aux données suivante :

$$\begin{aligned} proximité(\text{Livre 3}, \text{Ciseaux}) &= |14-8| + |22-22| + |1,3-1| = 6,3 \\ proximité(\text{Livre 2}, \text{Livre 3}) &= |16-14| + |24-22| + |2-1,3| = 4,7 \end{aligned}$$

Ainsi, on peut déduire des valeurs de proximité, que le livre 2 est plus proche du livre 3, que celui-ci ne l'est des ciseaux. Cette conclusion est dressée à partir du fait que le choix du calcul de la somme des différences en valeur absolue des valeurs de propriétés correspond à une distance. Ainsi deux objets sont proches si ils sont à distance faible l'un de l'autre. D'autre choix sont possibles pour définir une proximité, et peuvent avoir des comportements différents c'est le cas des mesures de similarité. Les différents types de mesures de proximité sont catégorisées en distances, dissimilarités, similarités, et écarts avec pour chacune, des propriétés mathématiques spécifiques.

En général, le calcul de la mesure de proximité se fait en tenant compte de toutes les propriétés des individus. Plusieurs problèmes se posent alors :
- comment établir une distance entre deux valeurs non numériques pour une propriété ? la distance entre Roux et Noir est-elle la même qu'entre Roux et Camel ?
- il existe plusieurs façons de calculer une proximité, quelle mesure choisir ? laquelle est la plus adaptée ?

Enfin, il arrive parfois que la représentation des individus soit inconnue ou non accessible. Dans ce cas, la mesure de proximité est connue pour chaque paire d'individus. On parle alors de données relationnelles, par opposition aux données de l'exemple, dites vectorielles, car chaque individu est décrit par un vecteur de valeurs correspondant aux propriétés. Une distinction existe également au sein d'une représentation vectorielle selon les types associés aux valeurs de propriétés (booléen, entier, flottant, chaîne de caractères, *etc.*). On parle alors de données numériques (flottant, entier) ou symboliques (entier, chaîne de caractères, booléen).

La construction des groupes

Se fondant ainsi sur des données et une mesure de proximité quantifiant la ressemblance ou la dissemblance entre les individus, nous pouvons désormais définir de manière informelle la tâche du *clustering* comme le développement d'algorithmes capables de construire un ensemble fini de groupes disjoints d'individus, ou *clusters*, de telle sorte que deux individus proches (respectivement éloignés) soient dans un même groupe (respectivement dans des groupes différents). La figure 0.1 présente ce à quoi ressemble un résultat de *clustering*, un ensemble de groupes contenant des individus proches.

Les groupes sont représentés ici par les piles d'individus (livres, articles, *etc.*) présentes sur le bureau. Dans l'exemple, on trouve des piles ou paquets de livres, d'ustensiles de bureau, ou de peluches, *etc.* Ainsi, le *clustering* revient à déplacer les individus de manière à ranger ceux-ci en catégories typiques. Ceci est équivalent *in fine* à ne pas déplacer les individus mais à leur attribuer à chacun l'étiquette de la catégorie leur correspondant. Ce *clustering*, ou processus de construction des groupes, peut prendre diverses formes. Une approche peut être de séparer en plusieurs tas dissemblables d'individus l'ensemble de tous les individus, ou bien de prendre chaque individu comme un tas et de rapprocher les tas les plus similaires, et ce jusqu'à atteindre un nombre de tas satisfaisant. Une autre manière consiste à identifier immédiatement un ensemble de paquets homogènes d'individus de taille fixée, dans le sens où, dans chaque paquet, les individus partagent un même ensemble de propriétés. Une correction peut alors être effectuée en changeant l'étiquette de certains individus si l'on s'aperçoit que ceux-ci partagent plus de propriétés avec les individus d'autres paquets. Ces éléments prennent régulièrement la forme de paramètres associés au processus de construction des groupes.

Notons qu'après avoir choisi la mesure de proximité entre les individus, et la manière de procéder à la construction des groupes, un praticien peut ne pas être satisfait du résultat produit. Ainsi, si les groupes obtenus ne correspondent à ses attentes, il convient de remettre en cause le choix de la mesure de proximité, ou les données en considérant :

- une incertitude et/ou une imprécision relative à la description, représentation des individus. On parle de bruit dans les données, que l'on peut prendre en compte en conservant la même mesure de proximité, mais en réduisant l'importance de certaines propriétés relativement aux autres.
- l'existence de certains liens entre propriétés tels que les valeurs de ces propriétés soient partagées de manière équivalente (ou au contraire différentes) par les individus d'un même groupe. On parle ainsi de propriétés corrélées positivement, négativement, ou non corrélées.
- l'existence d'individus éventuellement atypiques qui peuvent perturber le processus de construction des groupes. Dans ce cas on peut envisager de les écarter dans un premier temps, et de les réintégrer ou non ultérieurement.

Ainsi le processus d'analyse exploratoire de données est un processus non linéaire qui nécessite d'introduire des boucles de rétroaction afin d'orienter le choix de la représentation des individus et de la mesure de proximité entre ceux-ci (figure 0.2). Cette première analyse à visée plutôt pédagogique avait pour but de présenter de manière informelle les différentes étapes intervenant lors de l'analyse exploratoire de données classiques, ainsi que le cadre dans lequel s'inscrit le processus de construction des groupes, central dans ce travail de thèse. Je vais maintenant m'exercer à présenter le deuxième objet central dans ce travail de thèse, et qui concerne la nature des données à analyser ou regrouper : la multiplicité des sources et des représentations des données.

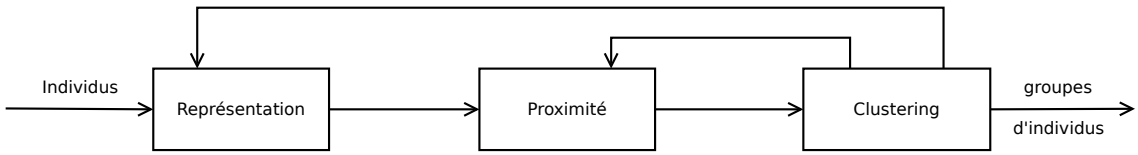

FIGURE 0.2 — Schéma du processus d'analyse exploratoire des données concernant le *clustering*.

La multiplicité des sources d'informations

Les données et les sources de données

Il est désormais possible, et fréquent, de disposer de plusieurs représentations pour un même ensemble d'individus [Bickel and Scheffer, 2004]. Ainsi de nouveaux défis interviennent autour de la construction de groupes d'individus désormais signifiés comme multi-représentés. La disponibilité de plusieurs représentations offre naturellement la possibilité de multiples regroupements, propres à chacune d'entre elles. Ce résultat peut permettre à un utilisateur d'avoir plusieurs groupements utiles pour une interprétation variée des données, mais cette multiplicité d'interprétations nécessite d'être contrôlée. Ainsi, selon la volonté pour un utilisateur d'avoir un seul ou plusieurs regroupements des individus, l'ensemble des représentations de ceux-ci devront être prises en compte, pour enrichir le processus de construction des groupes. Dans l'exemple précédent, nous avons construit une représentation des individus présents sur le bureau selon les propriétés que l'on pouvait déterminer à partir de notre perception visuelle de ceux-ci. Or les mêmes individus peuvent naturellement être appréhender *via* d'autres modes d'intéraction, comme le toucher. Ainsi nous pouvons construire une nouvelle représentation des individus, en considérant ce nouveau capteur sensoriel ou sens :

identifiant	masse	épaisseur	texture	chaleur	forme
Livre 1	Lourd	Moyen	Lisse	Froid	Rect.
Livre 2	Moyen	Moyen	Lisse	Froid	Rect.
Livre 3	Léger	Moyen	Lisse	Froid	Rect.
Article 1	Très léger	Très mince	Lisse	Moyen	Rect.
Article 2	Très léger	Très mince	Lisse	Moyen	Rect.
Peluche 1	Léger	Épais	Poilu	Moyen	Ellipt.
Peluche 2	Léger	Très épais	Poilu	Moyen	Ellipt.
Ciseaux	Léger	Mince	Lisse	Très froid	Triang.
Tasse	Moyen	Épais	Rugueux	Très froid	Cylind.
Crayon	Très léger	Très mince	Lisse	Froid	Cylind.

Nous pouvons définir une nouvelle mesure de proximité se fondant sur cette nouvelle représentation des individus et déduire un regroupement supplémentaire, alternatif au premier. Nous pouvons également envisager de regrouper les deux représentations en une seule et définir une mesure de proximité générale, ou bien encore considérer une telle mesure de proximité générale comme un mélange des deux mesures. Ainsi nous pouvons obtenir un seul regroupement des individus déterminé par le choix d'une proximité définie à partir de toutes les représentations. Nous pouvons également envisager d'obtenir ce regroupement en construisant localement (grâce à chaque représentation ou vue des données) un ensemble de groupes en observant ceux en construction dans les autres vues.

Notons que la définition d'une proximité mélangeant les différentes représentations, conjointe à l'identification de concepts, ouvre quelques problèmes. Certains concepts peuvent être transversaux à plusieurs représentation. Par exemple, un concept transversal ici pourrait être celui de la masse volumique associées aux différents individus, car il fait intervenir le concept de volume (identifié dans la première représentation) et la propriété de masse (présente dans la deuxième représentation). Un autre problème peut concerner l'existence de propriétés communes et ainsi excessivement corrélées comme la propriété d'épaisseur. On pourrait néanmoins envisager que les valeurs diffèrent selon le capteur ou sens utilisé pour appréhender les individus. C'est le cas

ici, pour l'attribut d'épaisseur. Ainsi, un problème majeur est celui de savoir quelle représentation apporte le moins d'imprécision sur la mesure de cette propriété. Voilà quelques problèmes complexes qui peuvent survenir lors de la prise en compte de plusieurs représentations simultanément.

Cette problématique est le thème majeur de ce travail de thèse qui concerne le développement de méthodes de classification non supervisée adaptées aux données dans un contexte de multiplicité des représentations. Avant de donner de plus amples détails sur les problématiques qui interviennent, nous nous éloignerons de l'exemple pédagogique pour présenter concrètement quelles sont les données types concernées par la multiplicité des sources et des représentations.

Les données multi-vues se retrouvent dans les diverses disciplines produisant de gigantesques quantités de résultats d'analyses concernant des objets d'études particuliers : des gènes pour les biologistes, des molécules pour les chimistes, des patients pour les médecins, *etc.*

Les données de la biologie. Dans le cadre de l'analyse du transcriptome, afin d'identifier des gènes qui interviennent dans les mêmes processus biologiques, gènes dits co-régulés, les bio-informaticiens analysent l'activité de ces gènes selon différentes conditions expérimentales (correspondant chacune à une représentation des gènes). De plus, des informations supplémentaires peuvent être extraites d'autres sources pour enrichir ces représentations dans un but par exemple de reconstruction d'un réseau de régulation génétique [Yamanishi et al., 2004] :

- les localisations de différentes protéines (encodées par des gènes particuliers) dans des régions intracellulaires ;
- les profils phylogénétiques d'organismes, qui contiennent chacun un ensemble de protéines. Ainsi pour chaque protéine on peut obtenir l'information de présence/absence de cette protéine dans chaque organisme ;
- les informations de compatibilités chimiques entre enzymes. L'hypothèse admise est qu'un lien existe entre de telles enzymes si elles partagent au moins un de leurs composés.

On peut également joindre aux mesures d'expressions l'analyse de documents de la littérature concernant ces gènes et constituer pour chacun d'eux un vecteur de termes apparaissant dans les documents scientifiques [Zeng et al., 2010].

Les données de la médecine. Dans le domaine de la médecine, différentes sources de données peuvent être intégrées dans un processus d'analyse complexe [Martin et al., 2006] :

- des données cliniques contiennent l'âge, le poids, le sexe pour un ensemble de patients, ainsi que la taille et le stade de la tumeur, ainsi que diverses informations sur les ganglions lymphatiques ou des résultats d'analyses de coupes histologiques ;
- des données catégorielles correspondent à une classification de la tumeur selon sa malignité ;
- des données issues de l'analyse de puces à ADN afin d'identifier les relations entres gènes à partir de leurs expressions dans différentes tumeurs.

Les données du *marketing*. Dans le contexte du *marketing*, des informations sur un même ensemble de clients sont disponibles à partir de différentes bases de données (banque, magasin, administration, *etc.*). On considère ici qu'une compagnie puisse collecter des informations sur un groupe de clients à partir de ces différentes bases pour constituer sa propre base de données. Les différentes entités ne pouvant ainsi pas échanger directement des informations pour des raisons de sécurité ou de confidentialité, les différentes données disponibles sont désignées comme multi-vues, chaque vue correspondant à une source différente [Pedrycz, 2002].

Les données multimédia. Les documents *web*, par leur nature, sont également des données multi-vues [Bekkerman and Jeon, 2007]. Chaque page *web* peut être décrite selon :
- le vocabulaire textuel *i.e.* l'ensemble des mots pertinents apparaissant dans cette page ;
- le vocabulaire graphique *i.e.* l'ensemble des images présentes dans le document *web* ;
- le vocabulaire hypertextuel correspondant aux liens sortant de la page. Notons que l'on peut obtenir également les liens entrant vers la page à partir d'un corpus de documents web en étudiant les liens sortant de chacun.

Une analyse d'un ensemble de pages peut permettre de constituer des groupes de documents à différents niveaux, afin d'offrir une organisation thématique de ces derniers. Chacun de ces vocabulaires est alors une vue différente de l'ensemble des pages *web*.

Les données de reconnaissance de caractères. Les techniques d'apprentissage sont également appliquées dans le domaine de la reconnaissance automatique de caractères manuscrits, comme la reconnaissance de code postal sur une adresse. Ces techniques imposent la définition d'une représentation des chiffres manuscrits. Une telle représentation peut être obtenue par des techniques de traitement du signal comme la transformée de Fourier, mais elle peut être complétée également par d'autres approches différentes capables de capturer une transformation naturelle des individus (une rotation ou une translation) [van Breukelen et al., 1998]. L'utilisation conjointe des différentes représentations permettent également de réduire globalement le bruit.

De la forme des données

Dans la réalité portée par les applications, les données multi-représentées peuvent se présenter de multiples manières. En effet, nous avons vu que parmi les cadres applicatifs majeurs dans lesquels on trouve de telles données, l'aspect décentralisé ou distribué des données est très présent et forme une caractéristique prégnante. Ces données multi-vues peuvent à la fois être distribuées :
- selon des groupes de variables, une vue correspondant à un groupe de variables décrivant un aspect nouveau sur l'ensemble des individus.
- selon des groupes d'individus, une vue est alors un échantillon particulier de l'ensemble des individus.
- dans le cas général, à la fois selon les individus et selon les variables. Des recouvrements peuvent exister entre les individus et les variables dans des vues différentes.

Ces différents cas sont présentés de manière schématique dans la figure 0.3. Dans cette thèse les différentes techniques présentées et approches proposées s'attachent à traiter le premier cas. C'est à dire où l'ensemble des individus (identique dans toutes les vues) est distribué selon des groupes de variables.

De l'organisation intrinsèque des données multi-vues. Parmi les caractéristiques des données multi-vues, relatives à l'existence d'une organisation en groupes des individus, on peut se questionner sur la définition des groupes parmi l'ensemble d'individus, ainsi que les concepts associés. Dans un cadre complètement non supervisé, on ne peut formuler que des hypothèses sur la relative étendue des concepts présents parmi les différentes vues. Ainsi :
- chaque vue peut correspondre à un ensemble de descripteurs visant à exprimer un ensemble de concepts semblables aux descripteurs des autres vues. En d'autres termes, les organisations induites par chaque vue sont naturellement proches et on cherchera à faire émerger la meilleur organisation globale des individus.

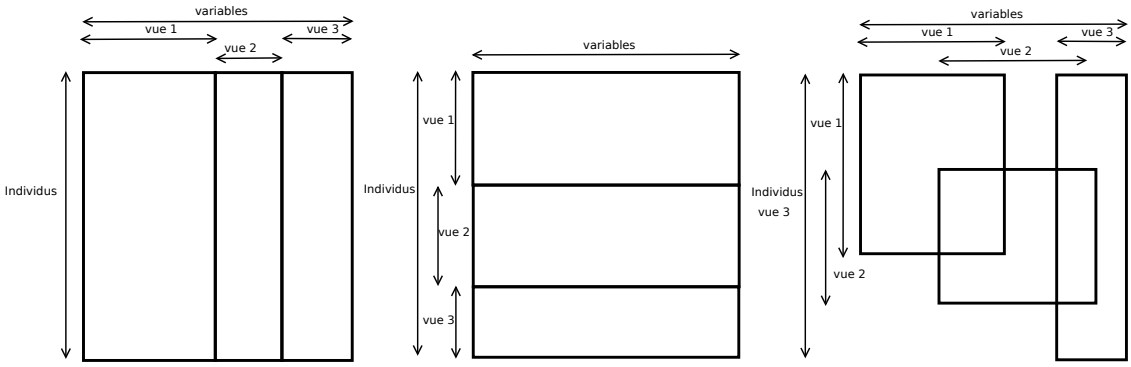

FIGURE 0.3 — Les types de données multi-vues. Dans l'ordre, ci-dessus, les données multi-vues décentralisées selon les variables, selon les individus et selon les variables et les individus simultanément.

- chaque vue est constituée d'un ensemble de descripteurs visant à représenter un ensemble de concepts indépendants des concepts existants dans les autres vues. Dans ce cas les organisations locales naturelles sont différentes, et chaque groupe devrait n'être défini uniquement dans une vue des données.
- dans le cas hybride : chaque vue exprime un ensemble de concepts et une partie des concepts de chaque vue correspond à une partie des concepts présents dans toutes ou partie des autres vues. Dans ce cas général les groupes sont identifiables en tenant compte simultanément de l'ensemble des vues mais également de l'importance relative de chacune d'entre elles dans la définition de chaque groupe.

Problématique et problématiques

La multiplicité est désormais très présente dans les communautés liées à la classification automatique . Cette multiplicité concerne autant les données auxquelles les approches présentées dans cette thèse tentent d'apporter des éléments d'analyse et de structuration, que les expertises pouvant être établies sur ces même données. Dans ce contexte, différents paradigmes ont été développés en apprentissage non-supervisé, en fouille de données ou en reconnaissance de formes. Ils permettent l'intégration de plusieurs sources d'informations afin d'établir un ensemble ou plusieurs ensembles d'hypothèses de classification sur des données non étiquettées.

***Clustering* semi-supervisé.** Le paradigme du *clustering* semi-supervisé concerne l'utilisation d'un ensemble de connaissances *a priori* sur l'appartenance ou non de paires d'individus à un même groupe. Cette information peut se présenter de deux façons différentes :
- on dispose pour chaque individu pris parmi les connaissances *a priori*, de son étiquette de classe ou de l'hypothèse d'un expert sur l'individu ;
- on dispose pour chaque paire d'individus pris parmi les connaissances *a priori*, d'une contrainte indiquant si les deux individus doivent, ou ne doivent pas appartenir à un même groupe.

On requiert en général que les méthodes de *clustering* semi-supervisées soient capables de satisfaire au mieux les connaissances disponibles et que l'utilisation de celles-ci permettent d'améliorer la production des hypothèses sur les appartenances de tous les individus. Il existe de plus d'autres formes de contraintes qui peuvent concerner notamment : les contraintes sur la taille

des groupes si nous disposons d'une connaissance sur l'homogénéité de ceux-ci, les contraintes de densité où l'on peut imposer une distance minimum entre les groupes, ainsi qu'une distance maximum entre deux individus d'un même groupe.

***Clustering* multi-vues.** Le paradigme du *clustering* multi-vues consiste à obtenir un unique *clustering* d'un ensemble d'individus décrits par de multiples représentations (données multi-représentées) que l'on appellera vues :

- on dispose pour chaque individu de plusieurs espaces de représentations, ou ensembles de variables. Il s'agit dans ce cas de données vectorielles multidimensionnelles ;
- on dispose pour chaque paire d'individus d'une information relationnelle sur leur proximité (distance, dissimilarité, similarité ou écart). Il s'agit dans ce cas de données relationnelles multidimensionnelles.

Notons que dans le cas des données relationnelles multidimensionnelles ou des données vectorielles multidimensionnelles de même dimensions, on peut parler de tenseurs d'ordre 3 ou de cube de données, pour représenter de telles données.

Dans le cadre des recherches dans le domaine du *clustering* de données multi-vues, on émet l'hypothèse suivante que l'on va chercher à réaliser : chaque vue apporte suffisamment d'informations pour réaliser un bon *clustering* (mais perfectible) de l'ensemble des individus, mais un meilleur *clustering* peut être obtenu par une utilisation conjointe de l'ensemble des vues.

***Clustering* d'ensemble.** Le paradigme du *clustering* d'ensemble consiste à obtenir un *clustering* d'un ensemble d'individus à partir d'un ensemble de résultats de *clustering* différents, obtenus par de multiples expertises apportées sur une même vue des données :

- les expertises apportées peuvent être certaines, auquel cas les *clusterings* correspondants seront dit durs ou stricts ;
- les expertises apportées peuvent être incertaines, auquel cas les *clusterings* correspondants seront dit flous, ou probabilistes.

Dans les travaux sur le *clustering* d'ensemble, et dans le même esprit que le *clustering* multi-vues, on émet l'hypothèse qu'un *clustering* consensus construit à partir de l'ensemble des *clusterings* disponibles sera meilleur que chaque expertise ou jugement pris isolément. Ces différents résultats de *clustering* peuvent avoir été obtenus par application de plusieurs algorithmes ou l'application du même algorithme mais avec des paramétrages différents, ou bien encore, par sous-échantillonage des données (individus ou descripteurs).

***Clustering* alternatif.** Le paradigme du *clustering* alternatif consiste à obtenir plusieurs *clusterings* à partir d'un ensemble d'individus décrits par une unique vue, et tels que :

- les *clusterings* obtenus soient de bonnes qualités ;
- les *clusterings* obtenus soient dissimilaires entre eux.

L'objectif sous-jacent aux techniques de *clustering* alternatif est de proposer à l'utilisateur un choix plus vaste de *clusterings* possibles pour une meilleure interprétation lors de l'analyse exploratoire. L'idée générale existe depuis longtemps et concerne notamment l'application d'un même algorithme de *clustering* sur un même jeu de données, mais avec des paramétrages différents. Cependant, elle a pris la forme d'un objectif contrôlé de recherche de dissimilarité entre les différents résultats de *clusterings*, introduisant de ce fait une dépendance entre les *clusterings* résultats à construire. Elle peut s'appliquer également dans le cas où l'on a à disposition des données multi-représentées et où l'on cherche localement un *clustering* optimal, mais que les *clusterings* locaux optimaux restent dissimilaires entre eux, ceci afin de maintenir de la diversité. Cette problématique a également été formalisée dans le cas où l'on a à disposition un jeu de

données mono-vue et une partition de cet ensemble d'individus. La tâche est alors de trouver (au moins) un nouveau *clustering* tel que celui-ci soit optimal pour l'ensemble des individus à disposition et différent de la partition donnée.

Liens entre les problématiques. L'ensemble de ces problématiques sont en réalité assez proches, et c'est pour cette raison que cette thèse propose de les mettre en commun et de les explorer. Les problématiques du *clustering* multi-vues et du *clustering* d'ensemble sont connexes de par les propriétés du *clustering* objectif (le consensus), mais diffèrent par l'entrée donnée aux méthodes répondant à ces problématiques (plusieurs descriptions d'un ensemble d'individus d'une part, plusieurs matrices de partitions d'autre part). Au second problème s'ajoute alors la prise en compte des descriptions des données et leur exploitation pour répondre au premier problème, donnant lieu ainsi au *clustering* collaboratif. Le *clustering* alternatif peut être vu comme une problématique duale aux deux précédentes, dans la mesure où, plutôt que de chercher un *clustering* unique consensus à partir de plusieurs *clusterings* provenant d'une ou plusieurs vues des données, on recherche un ensemble de *clusterings* dissimilaires, à partir d'une seule vue des données. Cependant, et dans la mesure où l'efficacité des méthodes d'ensemble repose sur une forme de diversité de l'ensemble des hypothèses à unifier, la recherche de *clusterings* alternatifs peut alors être le préalable adéquat à la recherche de consensus par *clustering* d'ensemble. Le *clustering* semi-supervisé, quant à lui, pourrait être considéré comme l'utilisation d'une matrice de partition partielle (peu de paires d'individus sont réellement identifiées comme étant dans un même groupe ou non), dans un *clustering* classique. On dispose ainsi d'une vue dans lesquels les individus sont décrits, et d'une autre vue correspondant à une partition de ce même ensemble d'individus, avec des valeurs manquantes. Enfin, dans la lignée du développement autour du *clustering* alternatif, une famille d'approches considère une matrice de partition complète des individus à utiliser dans un *clustering* classique mais pour trouver une partition alternative différente. Les travaux concernant ce type de *clustering* alternatif suivent directement les travaux du *clustering* semi-supervisé, où cette fois la matrice de partition sert à générer des contraintes opposées, dans l'esprit, à ce qui est connu dans la partition pour forcer l'algorithme de *clustering* à découvrir des groupes différents. Les différentes problématiques sont schématisées dans la figure 0.4.

Contributions et organisation

L'objectif de la thèse est de dresser un état de l'art des différentes techniques dédiées au traitement de données multi-vues et d'offrir à la communauté des approches nouvelles et/ou innovantes pour l'analyse exploratoire de telles données. Le développement d'approches permettant cette analyse conduit à en étudier les avantages, les cas d'applications et les limites et, dans le cas de ces dernières, à proposer de les outrepasser. Les limites inhérentes aux premières approches de *clustering* multi-vues, constatées après une étude préalable de l'état de l'art, concerne l'imposition trop arbitraire de paramètres nécessaires à l'expression d'un modèle intuitif, effectivement et aisément réalisable et implémentable, et offrant de bonnes propriétés de convergence. Ce constat a alors entraîné l'exploration de nouveaux moyens d'obtenir des solutions plus flexibles au regard de ces limites, mais moins élégantes et fondamentalement moins maîtrisées et contrôlées. Cela a conduit à l'étude des multiples paradigmes associés à la multiplicité des sources de données et/ou d'expertises sur ces données.

Faisant ainsi face aux diverses problématiques posées par l'analyse des données issues de plusieurs sources d'informations, les contributions apportées concernent à la fois le *clustering* multi-vues, le *clustering* sous contraintes ou *clustering* semi-supervisé, et enfin le *clustering* d'ensemble et le *clustering* alternatif.

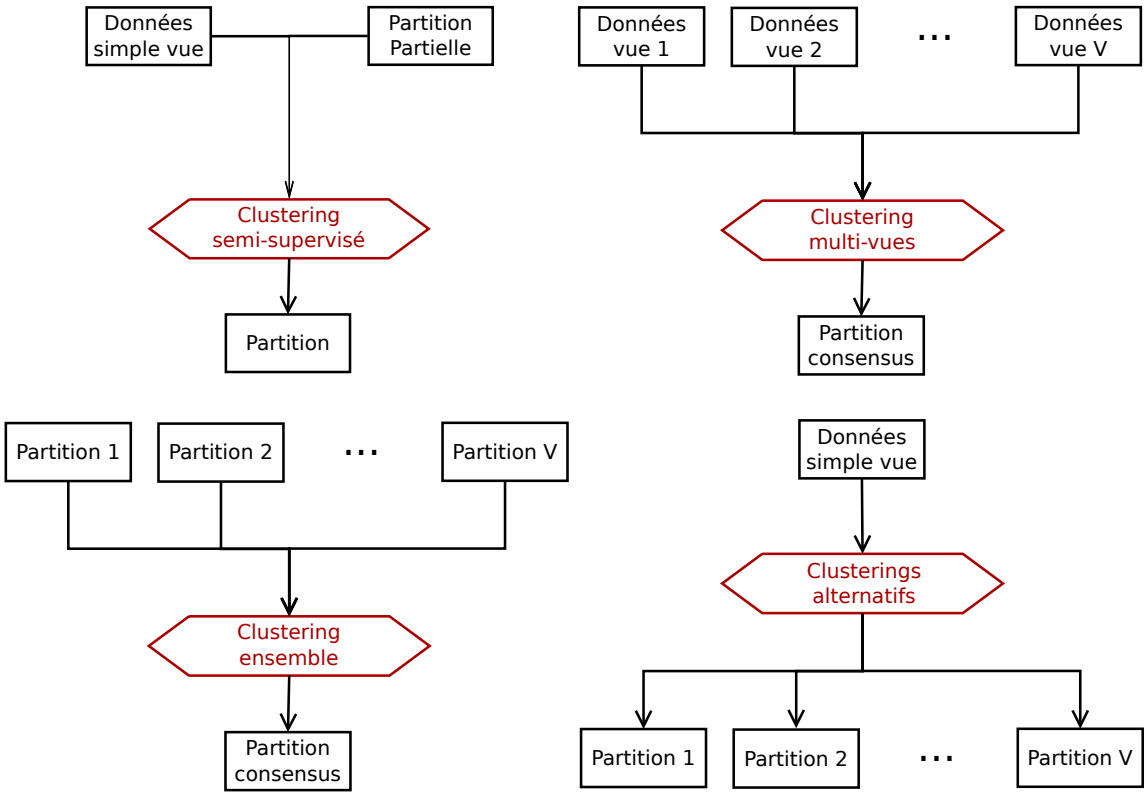

FIGURE 0.4 — Problématiques liées à la multiplicité dans les données. Dans l'ordre, ci-dessus, les problématiques du *clustering* semi-supervisé, puis le *clustering* multi-vues, le *clustering* d'ensemble et enfin le *clustering* alternatif.

Contributions au *clustering* multi-vues. Notre contribution au *clustering* multi-vues se concrétise par une approche permettant d'obtenir une unique partition à partir d'un ensemble d'individus multi-représentés, ainsi qu'une extension permettant le traitement d'un ensemble de relations sur ce même ensemble d'individus. Les méthodes usuelles consistent à chercher à réduire un désaccord entre les partitions obtenables dans chaque vue. Néanmoins l'étude de ces approches dresse le bilan d'une nécessité d'imposer des choix arbitraires sur les modèles pour obtenir des solutions analytiques élégantes, intuitives et convergentes. L'approche CoFKM repose sur l'optimisation d'une fonction objectif permettant simultanément de découvrir les groupes naturels présents dans chaque vue des données et d'assurer la réalisation d'un accord entre les différentes vues, au sens où les groupes obtenus dans chacune des vues doivent être similaires. L'utilisation de CoFKM nécessite d'avoir à disposition la description sous forme vectorielle de chaque individu dans chaque vue, cette limite est dépassée par la seconde approche : CoKFKM. CoKFKM permet non seulement de s'abstraire du type de représentation vectoriel contraignant, puisqu'elle repose sur la donnée pour chaque vue d'une information relationnelle sur l'ensemble des individus, mais offre également la possibilité d'utiliser différentes mesures de proximité pour chaque vue. Ceci permet de mieux correspondre avec la distribution naturelle locale (dans chaque vue) des individus. Des limitations d'ordre plus général sont apparues et se sont révélées *a priori* insolubles dans le modèle proposé. Nous nous sommes ainsi orientés vers une autre méthode pour réaliser un *clustering* consensus d'un ensemble d'individus. Celle-ci propose de

réaliser un consensus par échange de messages entre différents algorithmes de *clustering* appliqués dans chaque vue.

Ouverture aux autres problématiques liées à la multiplicité. Afin d'expliciter au mieux l'intuition sous-jacente à la suite des premières contributions, nous pouvons « humaniser » le principe de l'approche envisagée. Chaque vue des données induit naturellement une ou plusieurs organisations *naturelles*. Tout algorithme de *clustering* permet de retrouver une de ces organisations naturelles en groupes. On peut alors envisager d'appliquer un algorithme de *clustering* différent par vue pour obtenir une partition des individus qui peut s'avérer bien différente de celles obtenues dans les autres vues. Par une analogie grossière, mais pédagogique, on peut envisager que chacun de ces algorithmes de *clustering* soit un agent. L'objectif de cet agent est de produire le *clustering* qu'il jugera le meilleur selon son critère (sa fonction objectif) et ses *a priori* (ses paramètres) à partir de la distribution naturelle des individus. Le *clustering* produit *via* un raisonnement (le déroulement de l'algorithme) est une décision pour chaque paire d'individus (regroupés ou non regroupés) et prend la forme d'un ensemble d'hypothèses si l'on considère un principe d'incertitude relatif à l'agent et à sa décision. Une fois ses hypothèses émises, il peut associer à chacune un degré de confiance (principe d'incertitude). Les deux hypothèses possibles deviennent pour chaque paire : les deux individus présents dans la paire sont ensemble ou ne sont pas ensemble. Ainsi, à l'issue du raisonnement, l'agent peut, si il le souhaite, transmettre aux autres agents des messages, concernant certaines paires d'individus, du type : « je suggère de ne pas mettre ses individus dans un même groupe » ou « je propose de regrouper ces deux individus ». Selon cette perspective, nous pouvons ainsi suggérer la recherche d'un *clustering* consensus, en cherchant à faire collaborer les divers agents pour que les suggestions (hypothèses) émises par chacun permettent à tous de produire une décision communément acceptable, dans le sens où les décisions finales (ou l' ensemble des hypothèses) issues du raisonnement de chacun tendent vers une même solution. Ce principe relève du sens commun et du bien fondé de l'acceptation des divergences d'opinions, de remises en cause de ses propres points de vues pour arriver à des réponses consensuelles à des questions complexes.

Pour revenir à une terminologie plus usuelle dans les communautés scientifiques s'attaquant au *clustering* multi-vues, nous parlerons de contraintes sur les paires d'individus, qui seront transmises d'une vue vers les autres et que celles-ci devront satisfaire pour aller vers une solution globale souhaitée, par exemple, un ensemble de solutions locales proches entre elles, qui revient à réaliser un consensus. Le principe de cette approche a conduit dans un premier temps à étudier des méthodes apportant des solutions à la problématique du *clustering* sous contraintes et, dans un second temps à la proposition d'une approche répondant à cette problématique.

Contributions au *clustering* semi-supervisé. Les contributions au *clustering* semi-supervisé ont été réalisées au travers d'approches permettant l'intégration de connaissances externes sous forme de contraintes sur certaines paires d'individus, devant (ou ne devant pas) appartenir à un même groupe. Parmi les constats majeurs admis concernant cette problématique, deux points sont à observer avec attention :

- Les approches sont, à de rares exceptions près, limitées par la nécessité d'imposer un algorithme de *clustering* particulier. Historiquement, les algorithmes de la littérature étaient dédiées à la satisfaction absolue des contraintes données, faisant face ainsi directement dans le cas général, au problème bien connu de la NP-complétude du problème de la satisfiabilité auquel on peut, dans ce contexte, se réduire. L'intégration de procédures de tests de satisfiabilité des contraintes s'intégrant plus facilement lorsque l'algorithme de *clustering* est fixé, la limitation apparaît alors clairement. Pour «échapper» au problème de satisfiabilité, de la même manière que pour tenir compte d'incertitudes sur les contraintes

données, la satisfaction absolue de celles-ci a été relâchée. Les approches suivantes sont restées malgré cela dépendantes d'un algorithme en particulier, dans le but de satisfaire à une rigueur mathématique et esthétique (propriétés de convergence, contrôle optimal, *etc.*).
– Lorsqu'elles ne sont pas dépendantes d'un algorithme en particulier, les propositions de l'état de l'art consistent essentiellement à modifier la relation de proximité émanant des données dans le but de préparer en amont une meilleure satisfaction des contraintes par un algorithme de *clustering* quelconque. Néanmoins, ces méthodes souffrent de l'absence d'un dialogue (et d'un contrôle sur ce dialogue) entre l'hypothèse réalisée par le calcul de la nouvelle proximité et l'impact de celle-ci sur l'algorithme de *clustering*.

Partant de là, les contributions BoC, UzaBoC et AdaUzaBoC permettent l'intégration de contraintes et leur satisfaction par un algorithme de *clustering* quelconque. Ces approches sont basées notamment sur un contrôle de l'adéquation entre la modélisation de l'intégration des contraintes, et la satisfaction de celles-ci. BoC présente le défaut de ne bénéficier que d'une convergence programmé et non prouvée, ainsi que d'un manque de contrôle sur l'optimalité de la solution. UzaBoC outrepasse ces limitations en utilisant un algorithme d'optimisation numérique permettant de caractériser l'optimal en atteignant une convergence numérique. AdaUzaBoC est une version adaptative de la précédente, pour laquelle notamment la convergence est atteinte plus rapidement. Suivant l'objectif d'un modèle générique dont le principe est l'échange de contraintes entre algorithmes de *clustering*, nous avons choisi d'étendre l'algorithme AdaUzaBoC en vue de satisfaire l'objectif initial de *clustering* dans un contexte de multiplicité des données.

De la généricité d'un modèle face à de multiples contextes d'application. Pour finir, la dernière approche est en réalité une plateforme générique instanciable en deux variantes, permettant d'attaquer les différents problèmes de recherche de consensus et d'alternatives. L'approche CoBoC est complètement générique et répond à la problématique du *clustering* multi-vues. Elle propose dans chaque vue de produire un *clustering* local naturel, mais conscient des organisations naturelles existantes dans les autres vues, par l'intermédiaire de contraintes. La problématique de génération de bonnes contraintes est alors centrale dans le but d'obtenir des solutions locales proches dans toutes les vues. CoBoC est une approche fondée sur des heuristiques frugales de génération de contraintes qui permettent de moduler d'une part, entre recherche de solutions locales similaires ou dissimilaires (alternatives) entre elles, et d'autre part, de produire un *clustering* local final dans chaque vue alternatif au *clustering* naturel sans intégration de contraintes émanant des autres vues. La volonté d'obtenir un consensus par génération de contraintes externes est à rapprocher des techniques visant à obtenir un consensus de partitions comme la recherche de partitions médianes ou le *clustering* d'ensemble, sources d'inspiration et offrant des points de comparaison ainsi qu'une possibilité d'intégration. Le choix des contraintes à générer pour atteindre l'objectif nécessite l'exploration de l'apprentissage actif, afin de produire des heuristiques non frugales. Enfin, obtenir un consensus peut ne pas être l'objectif escompté, si l'on cherche à obtenir de la diversité dans les productions d'hypothèses comme dans le cas de l'alternative *clustering*. Ainsi, les heuristiques de génération de contraintes peuvent être conçues pour tendre vers cet objectif.

Organisation de la thèse.

Dans le but de présenter de la manière la plus complète ces approches, un panorama de la classification non supervisée et des techniques fondamentales utilisées pour le traitement de données classiques (mono-représentées et mono-sources) sera dressé (Chapitre 1). Dans la suite

sont présentées les différentes contributions apportées au *clustering* multi-vues (Chapitre 2), au *clustering* semi-supervisé (Chapitre 3) et à la collaboration entre algorithmes de *clustering* (Chapitre 4) pour des objectifs de recherche de consensus, dans le contexte de la combinaison de modèles adaptés aux données mono-vue/mono-source et le contexte multi-vues, ou de recherche de *clusterings* alternatifs de données mono-vue/mono-source. Ces contributions sont introduites à chaque fois par le contexte scientifique et l'analyse de l'état de l'art répondant à la problématique posée, puis sont discutées en présentant les perspectives d'évolution. Pour finir, la conclusion clôturera le mémoire par une synthèse de l'ensemble des travaux réalisés.

Ce manuscrit de thèse présente l'ensemble des contributions apportées et liées au *clustering* dans un contexte de multiplicité de données. Comme présenté précédemment, cette multiplicité peut concerner :

- la **multiplicité des représentations** des individus ou des informations relationnelles entre ceux-ci ;
- la **multiplicité des sources d'informations** et la nature de ces informations ;
- la **multiplicité des traitements** possibles pour produire des résultats de *clusterings* alternatifs et intéressants.

Le manuscrit est composé, hormis la précédente introduction, de quatre chapitres dont les lignes directrices sont esquissées ci-après.

Chapitre 1 : Classification non supervisée.

Le chapitre 1 présente les algorithmes classiques de classification non supervisée. Celles-ci s'adressent aux données mono-vue et mono-source. Loin de dresser un état de l'art exhaustif de ces approches, le panorama proposé constitue un socle adapté pour la compréhension des différents algorithmes des prochains chapitres, ainsi que celle des contributions proposées. Les principales familles d'approches sont présentées, ainsi que les principaux problèmes associés à l'utilisation de celles-ci et ceux liés au *clustering* en général.

Chapitre 2 : Classification non supervisée centralisée de données multi-vues.

Le chapitre 2 présente l'élaboration d'une approche de *clustering* multi-vues dont le but est de produire une unique partition résultant du traitement d'un ensemble d'individus décrits par plusieurs représentations ou plusieurs tableaux relationnels, répondant ainsi au premier problème sur la multiplicité des données. Cette contribution se fonde sur un algorithme classique présenté dans l'état de l'art: les K-moyennes floues [Bezdek, 1981], et sur un principe régissant le développement des approches de l'état de l'art : la minimisation d'un désaccord entre les *clusterings* naturels des différentes vues.

Chapitre 3 : Classification non supervisée et intégration de connaissances externes.

Le chapitre 3 présente deux approches de *clustering* semi-supervisées. L'objectif est d'améliorer un algorithme de *clustering* en incorporant des connaissances issues de sources externes, et spécifiant, pour certaines paires d'individus, la relation d'appartenance ou non de ces individus à un même groupe. Les approches proposées permettent l'amélioration de n'importe quel algorithme de *clustering*, ce qui les rend plus flexibles. Ainsi, les approches proposées peuvent encapsuler les différents algorithmes présentés dans l'état de l'art sur le *clustering*. Elles se fondent sur le principe de réduction de dimensions contrôlé dans le but d'être en adéquation avec les connaissances externes. Celles-ci se différencient selon la méthode de résolution employée. La première est basée sur un principe de *boosting* adaptatif, technique très efficace d'amélioration

de performance dans un cadre d'apprentissage supervisé, et porté ici dans le cadre du *clustering* semi-supervisé. La seconde est basée sur une méthode d'optimisation numérique offrant des garanties de convergence numérique vers une solution que l'on peut caractériser.

Chapitre 4 : Classification non supervisée et collaboration.

Le chapitre 4 présente un algorithme flexible et paramétré permettant d'attaquer à la fois les problématiques de multiplicité des représentations des individus, et en même temps d'offrir une multiplicité de *clusterings* alternatifs à partir d'individus décrits par de multiples représentations ou non. L'algorithme proposé permet de s'abstraire des algorithmes de *clustering* et permet également la collaboration entre ces algorithmes. Cette collaboration permet d'atteindre les objectifs de (**1**) recherche de consensus, dans le contexte de la combinaison de modèles adaptés aux données mono-vue et mono-source et le contexte multi-vues, ou de (**2**) recherche d'alternatives de *clustering* pour données mono-vue et mono-source. Elle est fondée sur une des approches développées précédemment pour le *clustering* semi-supervisé, couplée à un ensemble d'heuristiques caractérisant l'objectif recherché.

Conclusion.

Pour finir, la conclusion clôturera le mémoire par une synthèse de l'ensemble des travaux réalisés et permettra de dresser les perspectives à court et moyen terme du développement des approches proposées.

Classification non supervisée 1

Sommaire

1.1 Introduction **26**

1.2 Approches hiérarchiques **27**

1.2.1 DIANA : *DIvisive ANAlysis* 27

1.2.2 AGNES : *AGglomerative NESted clustering* 28

1.3 Approches partitives **30**

1.3.1 Approches basées sur les prototypes 30

1.3.1.1 KM : les K-moyennes 30

1.3.1.2 SC : *clustering* spectral 32

1.3.2 Approches basées sur la densité 34

1.3.2.1 DBSCAN : *clustering* basé sur la densité 34

1.3.2.2 SOM : les cartes auto-organisatrices 35

1.4 Approches floues et probabilistes **37**

1.4.1 FKM : les K-moyennes floues 37

1.4.2 EM : estimation d'un mélange de modèles par Espérance-Maximisation 39

1.5 Bilan **41**

1.5.1 Les liens entre familles d'algorithmes de *clustering* 41

1.5.2 Le problème du nombre de groupes 42

1.5.3 Le problème de l'évaluation 43

1.5.3.1 Mesures basées sur l'énumération 44

1.5.3.2 Mesures statistiques basées sur l'entropie. 45

1.5.4 Le choix de la proximité 46

1.5.5 Le choix de l'algorithme 47

1.1 Introduction

Dans ce chapitre introductif sont présentées les grandes familles d'algorithmes de *clustering*. Il existe de multiples critères pour les différencier et les présenter en une typologie cohérente. La présentation de ces approches de classification suit une trame classique selon le type de résultat produit. Les différents algorithmes classiques seront organisés selon les approches :

- **hiérarchiques** produisant un ensemble de partitions imbriquées appelé *dendrogramme*;
- **partitives**, dont le résultat est une partition en un nombre de groupes fixé, donné, ou découvert par l'algorithme;
- **génératives** ou **floues**, permettant d'obtenir une partition floue des individus, ou d'attribuer des valeurs de probabilités d'appartenance des individus à chaque groupe.

Il sera tenu compte également de la philosophie de l'approche et spécifié pour chacune, si elle est de type :

- **discriminative**, c'est à dire que l'approche vise à déterminer géométriquement des frontières de décision en séparant les individus dans un espace donné. Elles se présentent sous la forme d'un programme d'optimisation d'un critère objectif, éventuellement avec contraintes. La solution est ainsi caractérisée par l'optimum global de ce critère objectif, qui est en général approché par un algorithme directement dérivé du critère ;
- **générative**, lorsqu'elle est basée sur un modèle probabiliste. L'approche permet ainsi de définir à la fois la solution optimale après émission d'une hypothèse sur la nature des lois censées régir l'ensemble d'individus, et en même temps propose une explication sur la façon dont a été généré cet ensemble. Cette dernière information offre l'intérêt notamment de pouvoir re-générer automatiquement un nouvel ensemble d'individus, semblable à celui d'origine;
- purement **algorithmique**, si elle n'est fondée sur aucun critère objectif (ou celui-ci n'est pas connu), mais les groupes sont obtenus par pure recherche heuristique durant l'application de l'algorithme.

Avant d'analyser en détail ces approches et les liens existant entre elles, il convient de rappeler l'hypothèse centrale qui régit chacune d'entre elles ainsi que leur utilisation : l'hypothèse de l'existence de groupes. Cette hypothèse établit que des échantillons d'individus très proches entre eux doivent appartenir au même groupe et partager alors la même étiquette. De manière équivalente la frontière de décision entre deux groupes doit correspondre à une zone de faible densité *i.e.* zone dans laquelle peu d'individus sont présents. En prendre connaissance est important, étant donné que l'application d'un algorithme de *clustering* n'a de sens que si l'on peut confirmer cette hypothèse, par exemple, en s'assurant que les individus ne sont pas distribuées selon une loi uniforme. Ce dernier cas peut relever d'un problème de représentation des individus ou de choix de la mesure de proximité, quoiqu'il en soit, d'un problème intervenant en amont de la procédure de découverte des groupes.

Au delà de l'utilité d'appliquer un algorithme de *clustering* et du cœur constituant celui-ci, figure le problème de l'évaluation. Pouvoir déterminer l'apport d'un algorithme de *clustering* particulier est un problème en soi. Un cadre favorable d'évaluation se présente lorsque l'étiquette des individus est connue *i.e.* lorsque l'on a à disposition des groupes cibles appelés *classes*, à retrouver. L'évaluation est alors dans ce contexte une vérification de la ressemblance entre les groupes produits et les classes données. Le problème de l'évaluation sera adressé plus en détail en fin de chapitre.

L'objectif des approches de *clustering* est de produire une structure permettant d'organiser les données. Celle-ci peut être un dendrogramme, ou une partition de taille fixée éventuellement représentée par un ensemble d'éléments représentatifs appelés *prototypes*. La forme et la

manière d'obtenir une telle structure sera explicitée en temps voulu lors de la présentation des différentes familles d'algorithmes. L'ensemble des méthodes classiques présentées sont formalisées selon la notation suivante :

NOTATION	
n :	le nombre d'individus à regrouper.
n_p :	le nombre d'attributs décrivant les individus.
n_k :	le nombre de groupes à identifier.
n_c :	le nombre de classes associé aux données.
$\mathcal{X} = \{x_1, ..., x_n\}$:	l'ensemble des n individus à partitionner.
$X \in \mathbb{R}^{n \times n_p}$:	la représentation matricielle de $\mathcal{X}$.
$x_i \in \mathbb{R}^{n_p}$:	la représentation vectorielle de l'individu x_i.
$C = \{C_1, ..., C_{n_k}\}$:	la structure de *clustering* en n_k groupes à construire.
$c = \{c_1, ..., c_{n_k}\}$:	l'ensemble des n_k prototypes des groupes.
$\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{n_c}\}$:	l'ensemble des n_c classes d'individus à retrouver.
$\mathcal{D} = \{\mathcal{D}_0, ..., \mathcal{D}_n\}$:	la structure de dendrogramme associée aux données.
$d(x_i, x_j)$:	la distance au sens général entre deux individus x_i et x_j.
$\|\|x_i - x_j\|\|_p$:	la distance de Minkowski entre deux individus x_i et x_j.

1.2 Approches hiérarchiques

Les approches de *clustering* hiérarchiques sont des approches non paramétriques et purement algorithmiques qui proposent de construire une structure hiérarchique appelée *dendrogramme*. Il s'agit d'un arbre dans lequel chaque niveau correspond à une partition de l'ensemble des individus. Chaque noeud, appelé aussi *amas*, est une partie de la partition correspondante (un groupe) et l'ensemble de ses fils constitue une partition de ce noeud. La figure 1.1 illustre cette structure. Les approches permettant de construire un dendrogramme de ce type se décomposent en deux familles :

- les approches agglomératives qui construisent le dendrogramme par la base, en regroupant à chaque étape les amas d'individus les plus similaires ;
- les approches divisives qui construisent le dendrogramme par le haut, en partitionnant à chaque étape un amas en sous amas.

1.2.1 DIANA : *DIvisive ANAlysis*

Algorithme

L'approche DIANA pour *DIvisive ANAlysis clustering* suggère une construction descendante du dendrogramme. Partant d'un amas A non singleton et de plus grand diamètre (contenant initialement l'ensemble des individus $x_i \in \mathcal{X}$), l'algorithme procède par division successive et itérative en deux parties A' et $\overline{A'}$ équilibrées. Le diamètre d'un amas A est défini par :

$$Diam(A) = \max_{x_i \in A, x_j \in A} d(x_i, x_j) \tag{1.1}$$

Partant de $A' = A$ et $\overline{A'} = \emptyset$, l'approche consiste alors à transférer un ensemble d'individus de A' vers $\overline{A'}$ de telle sorte à conserver un équilibre entre ces deux ensembles. On choisit de

transférer à chaque étape l'individu $x_i \in \mathcal{X}$ qui maximise

$$D(x_i, A' \setminus \{x_i\}) = \frac{1}{|A'| - 1} \sum_{\substack{x_j \in A' \\ x_j \neq x_i}} d(x_i, x_j) \tag{1.2}$$

correspondant à une distance moyenne de l'individu x_i aux individus de $A' \setminus \{x_i\}$. Lorsque la quantité $\Delta(A', \overline{A'}, x_i) = D(x_i, A' \setminus \{x_i\}) - D(x_i, \overline{A'})$ devient négative, l'individu x_i n'est alors pas transféré et le processus de division de A s'arrête. Une nouvelle subdivision peut alors recommencer en choisissant à nouveau l'amas de plus grand diamètre entre A' et $\overline{A'}$. L'algorithme est présenté en détail dans l'algorithme 1.

Algorithme 1 DIANA

ENTRÉES : $\mathcal{X}$, $d(.,.)$
SORTIES : $\mathcal{D}$
1 : $A = \mathcal{X}$, $\mathcal{D}_0 = \{A\}$ et $i = 1$
2 : $A' = \underset{A \in \mathcal{D}_{i-1}}{arg\ max}\, Diam(A)$, $\overline{A'} = \emptyset$ et $\mathcal{D}_i = \mathcal{D}_{i-1} \setminus A'$
3 : choisir $x_i^* = \underset{x_i \in A'}{arg\ max}\, D(x_i, A' \setminus \{x_i\})$
4 : si $\Delta(A', \overline{A'}, x_i^*) \geq 0$ alors $A' = A' \setminus \{x_i^*\}$, $\overline{A'} = \overline{A'} \cup \{x_i^*\}$ et aller en **3**
5 : si $i < |\mathcal{X}|$ alors $i = i + 1$, $\mathcal{D}_i = \mathcal{D}_i \cup A' \cup \overline{A}$ et aller en **2**

L'un des problèmes majeurs de l'approche DIANA est sa sensibilité aux *outliers*, qui sont des individus isolés dont on peut considérer qu'ils proviennent d'une erreur de mesure ou d'un comportement anormal selon le cadre applicatif. En effet ceux-ci biaisent la définition du diamètre d'un amas et perturbe le processus de subdivision. Les approches divisives restent moins présentes et utilisées que les approches agglomératives pour des raisons de complexité, le problème de trouver une bipartition optimale pour tout critère étant lui même NP-difficile.

1.2.2 AGNES : *AGglomerative NESted clustering*

Algorithme

Les méthodes agglomératives de type AGNES pour *AGglomerative NESted clustering* consistent à partir d'autant d'amas singletons que d'individus, puis à fusionner dans un processus itératif les amas les moins dissimilaires, ou de manière équivalente, les plus similaires (cf. algorithme 2). La dissimilarité inter-amas peut être calculée de multiples façons, et le choix de la mesure influence grandement le résultat de l'algorithme agglomératif. Les différentes mesures mènent naturellement à différentes déclinaisons de la méthode de construction des partitions imbriquées. Parmi elles, nous trouvons :

SLINK ou le simple lien qui consiste à utiliser la mesure :

$$D(A_i, A_j) = \underset{x_i \in A_i, x_j \in A_j}{min} d(x_i, x_j)$$

La distance entre deux amas est alors la distance la plus courte entre individus de ces amas ;

ALINK ou la méthode en lien moyen utilisant la mesure de dissimilarité inter-amas suivante :

$$D(A_i, A_j) = d(c_i, c_j)$$

où $c_i = \frac{1}{|A_i|} \sum_{x_i \in A_i} x_i$ et $c_j = \frac{1}{|A_j|} \sum_{x_j \in A_j} x_j$ sont les moyennes respectives des amas A_i et A_j. La distance entre deux amas correspond dans ce cas à la distance entre les barycentres respectifs de ceux-ci ;

CLINK ou la méthode en lien complet qui est basée sur la définition de la mesure suivante :

$$D(A_i, A_j) = \max_{x_i \in A_i, x_j \in A_j} d(x_i, x_j)$$

La distance entre deux amas devient la distance la plus grande entre individus présents dans ces amas.

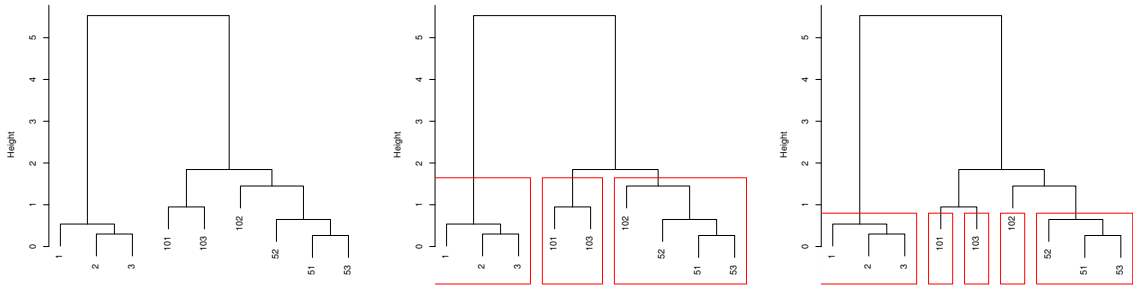

FIGURE 1.1 — Dendrogramme obtenu après application d'un *clustering* hiérarchique. Les deux dernières images correspondent à une coupure du dendrogramme afin d'obtenir une partition "à plat" du nombre de groupes désiré ($n_k = 3$ et $n_k = 5$ respectivement).

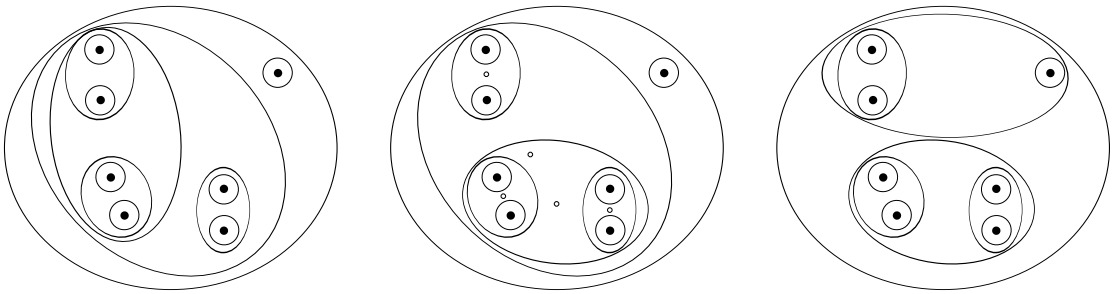

FIGURE 1.2 — . Les différents résultats issus de l'application d'un algorithme de *clustering* hiérarchique agglomératif selon différentes mesures de dissimilarité inter-amas. Dans l'ordre, le *clustering* SLINK, le *clustering* ALINK et le *clustering* CLINK.

D'autres approches divisives et agglomératives ont été explorées et sont présentes dans la littérature [Kaufman and Rousseeuw, 1990]. Elles ne seront pas présentées ici car elle ne servent pas de socle pour les algorithmes développés par la suite pour les problématiques spécifiques. Ces méthodes sont utiles lorsque l'on souhaite une analyse de l'ensemble d'individus à plusieurs niveaux de granularité, au travers de plusieurs partitions imbriquées de 1 à n groupes. Elles

Algorithme 2 AGNES

ENTRÉES : $\mathcal{X}$, $d(.,.)$
SORTIES : $\mathcal{D}$
1 : $\forall x_i \in \mathcal{X}, A_i = \{x_i\}$, $\mathcal{D}_0 = \{A_i\}_{1 \leq i \leq |\mathcal{X}|}$ et $i' = 1$
2 : $(A_1^*, A_2^*) = \underset{(A_1,A_2) \in \mathcal{D}_{i'-1}^2}{arg\ min} \quad D(A_1, A_2)$
3 : $\mathcal{D}_{i'} = \mathcal{D}_{i'-1} \setminus \{A_1^*\} \cup \{A_2^*\}$ et $\mathcal{D}_{i'} = \mathcal{D}_{i'-1} \cup \{A_1^* \cup A_2^*\}$
4 : si $i' < |\mathcal{X}|$ alors $i' = i' + 1$ et aller en **2**

peuvent être implémentées par des algorithmes de complexité acceptable à l'aide de structures de données adéquats et peuvent alors être adaptées à de grandes bases. Néanmoins, elles sont limitées par le fait que lorsque deux amas deviennent agglomérés, une réaffectation des individus présents dans ces amas n'est plus possible. Ainsi, le meilleur *clustering* des individus en n_k amas ne peut espérer être réellement atteint, et on préférera utiliser des méthodes par partitionnement en n_k groupes, plus adaptées.

1.3 Approches partitives

Cette sous section détaille les algorithmes de partitionnement selon deux familles d'approches :

- les approches basées sur les prototypes qui consistent à définir un ensemble de centres ou moyennes de départ qui caractériseront chacun un groupe d'individus ;
- les approches basées sur le voisinage qui émettent des hypothèses topologiques sur la distribution de l'ensemble d'individus $\mathcal{X}$.

Ces approches sont désormais plus courantes dans les communautés dédiées au développement de nouveaux algorithmes de *clustering*. Dans la suite sont présentés des exemples de telles familles ainsi que les liens que l'on peut trouver entre elles.

1.3.1 Approches basées sur les prototypes

1.3.1.1 KM : les K-moyennes

La méthode discriminative des K-moyennes [MacQueen, 1967], notée KM, est l'approche la plus connue, utilisée et étendue dans les différentes communautés dédiées au *clustering*. Le principe est «naturel», étant données la distribution des individus de $\mathcal{X}$ dans l'espace de description et un nombre n_k de groupes fixé, l'objectif est de minimiser la dispersion des individus relativement à un ensemble de prototypes représentatifs de ces groupes.

Objectif

Les individus $x_i \in \mathcal{X}$ doivent nécessairement être représentés par un vecteur de $\mathbb{R}^p$, et l'ensemble $\mathcal{X}$ est alors décrit par une matrice $X \in \mathbb{R}^{n \times p}$. Du point de vue du modèle, KM est basé sur la minimisation d'une erreur quadratique relativement à ces prototypes qui se formalise par:

$$\underset{c,C}{min}\ Q_{\text{KM}}(c, C) = \underset{c,C}{min} \sum_{k=1}^{n_k} \sum_{x_i \in C_k} ||x_i - c_k||_2^2$$

où c_k est le prototype du groupe C_k.

Algorithme

Du point de vue de l'algorithme (cf. algorithme 3), il s'agit d'un processus itératif qui alterne, à chaque étape:

1. une phase d'affectation des individus à leur groupe le plus proche :

$$C_k^* = \{x_i \in \mathcal{X} \mid c_k = \underset{c \in \{c_1,\ldots,c_{n_k}\}}{arg\ min}\ ||x_i - c||_2^2\} \tag{1.3}$$

2. une phase de mise à jour des centres de groupe :

$$\begin{aligned} c_k^* &= \underset{c \in \mathbb{R}^p}{arg\ min} \sum_{x_i \in C_k} ||x_i - c||_2^2 \\ &= \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \end{aligned} \tag{1.4}$$

Le nouveau prototype est alors le barycentre du sous ensemble des individus $x_i \in C_k$.

La figure 1.3 retrace le principe de l'algorithme KM. À la première itération, 3 prototypes sont définis aléatoirement et les première affectations (représentés par les colorations) sont réalisées relativement à ces prototypes. À l'itération 2 on observe le déplacement des prototypes par la traînée rouge et une réaffectation correspondante à la nouvelle position de ceux-ci. La dernière illustration montre l'algorithme stabilisé qui parvient à trouver 3 groupes convexes et homogènes.

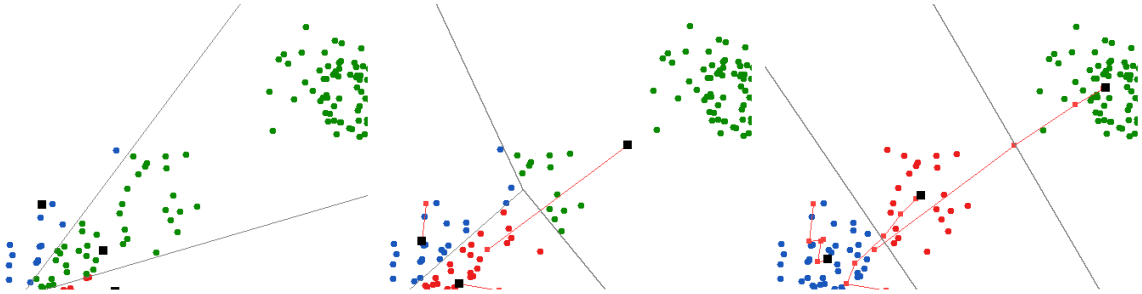

FIGURE 1.3 — Illustration des étapes de KM à partir des itérations 1, 2, et 8 correspondant à la stabilisation ($n_k = 3$).

Algorithme 3 KM

ENTRÉES : $\mathcal{X}$, n_k
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation aléatoire des n_k centres de groupes $\{c_1, ..., c_{n_k}\}$
2 : Mise à jour des groupes C_k $\forall k \in [1..n_k]$ en utilisant (1.3)
3 : Mise à jour des centres de groupe c_k $\forall k \in [1..n_k]$ en utilisant (1.4)
4 : Si la valeur de Q_{KM} change alors aller en **2**

On notera qu'il s'agit là d'un problème d'optimisation non convexe, c'est à dire que l'on ne peut avoir de garantie d'atteindre l'optimum global du critère. À chaque étape, la mise à jour des groupes est optimale selon la définition actuelle des centres. Les nouveaux centres eux

remplissent les conditions d'optimalité du premier ordre. Ainsi l'optimalité locale ou globale est complètement déterminée par l'initialisation des centres. En général, KM est exécuté plusieurs fois avec des initialisations différentes et le meilleur résultat est retenu.

Parmi les avantages, on notera la complexité linéaire de l'algorithme en le nombre d'individus, la simplicité d'implémentation et l'interprétation naturelle du modèle et de l'algorithme associé. La convergence théorique est également prouvée, par le fait que le critère à minimiser est positif, admet l'existence d'un optimum, puis que sa valeur décroît à chaque étape de l'algorithme.

Parmi les inconvénients, on peut noter que KM est limité par la représentation des individus. Chaque individu doit ainsi être décrit par un vecteur numérique de dimensionnalité p. Ainsi il n'est pas directement applicable si les données sont représentées directement par une matrice de proximité de type similarité ou dissimilarité. Un autre désavantage concerne le fait que KM ne peut produire que des groupes convexes, et de diamètre homogène.

1.3.1.2 SC : *clustering* spectral

Le *clustering* spectral (SC) [Luxburg, 2007] est une autre approche discriminative de partitionnement, qui aurait pu être traitée parmi les approches basées sur le voisinage, car elle permet de prendre en compte la topologie naturelle des données. En réalité, il s'agit d'un KM appliqué à l'ensemble des individus projetés dans un sous espace particulier. Cet espace de projection de dimensions n_k est construit de telle sorte que des paquets d'individus proches se forment naturellement dans chaque dimension. Le critère objectif correspond donc à une variante de KM [Dhillon et al., 2005] qui ne sera pas détaillée ici.

Algorithme

L'algorithme 4 repose sur une représentation des données sous formes d'un graphe de similarité G traduisant la notion de proximité entre individus. Il existe plusieurs façons de construire un tel graphe à partir des données :

- dans le graphe de voisinage ϵ, une arête existe entre deux individus $x_i \in \mathcal{X}$ et $x_j \in \mathcal{X}$ si $d(x_i, x_j) \leq \epsilon$;
- dans le graphe des k plus proches voisins $k\mathcal{NN}$, une arête existe entre les individus $x_i \in \mathcal{X}$ et $x_j \in \mathcal{X}$ si $x_j \in k\mathcal{NN}(x_i)$ *i.e.* x_j est parmi les k individus les plus proches de x_i ;
- le graphe complet, une arête existe pour toutes les paires d'individus.

Les différentes arêtes du graphe sont munies d'un poids correspondant à la similarité entre les deux individus concernés par l'arête, similarité qui peut être calculée de multiples manières et le choix en est laissé selon le cadre applicatif.

Le sous-espace dans lequel projeter les données s'obtient en calculant par diagonalisation les vecteurs propres du *laplacien* du graphe choisi. Le résultat utilisé étant que les vecteurs propres du *laplacien* caractérisent des composantes connexes du graphe lorsque leurs valeurs propres associées sont nulles, ou bien des zones de fortes densité (mais non déconnectées du graphe) lorsqu'elles sont petites. Le *laplacien* L du graphe est défini à partir de la matrice d'adjacence W du graphe et de la matrice diagonale D des degrés de ses sommets (les individus):

$$L = D - W$$

avec W la matrice d'adjacence définit par :

$$W_{ij} = \begin{cases} 1 & \text{s'il existe une arête entre } x_i \text{ et } x_j \\ 0 & \text{s'il n'existe pas d'arête entre } x_i \text{ et } x_j \end{cases}$$

et D la matrice diagonale des degrés

$$D = diag(d_1, ..., d_n) \ ; \ d_i = \sum_{x_j \in \mathcal{X}} w_{ij}$$

La valeur $W_{ij} \geq 0$ peut également refléter la similarité entre x_i et x_j plutôt que l'existence d'une arête.

Une étape clé avant le calcul des vecteurs propres et la diagonalisation est la normalisation du *laplacien*. Différentes approches ont été développées selon le type de normalisation proposé [Shi and Malik, 2000] ; [Ng et al., 2001]. Ainsi les normalisations possibles sont les suivantes :

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \quad (1.5)$$

$$L_{rw} = D^{-1} L = I - D^{-1} W \quad (1.6)$$

Le choix de la normalisation a une influence sur les vecteurs propres du *laplacien*, et ceux-ci correspondent alors à des solutions de problèmes relachés de partitionnement de graphes selon différentes heuristiques. En particulier, soit :

– le volume du groupe C_k, noté $vol(C_k)$ défini par :

$$vol(C_k) = \sum_{x_i \in C_k} W(C_k, \mathcal{X} \setminus C_k)$$

où $W(C_k, \mathcal{X} \setminus C_k)$ correspond au nombre d'arêtes, ou à la somme des poids des arêtes entre les individus $x_i \in C_k$ et $x_j \in \mathcal{X} \setminus C_k$:

$$W(C_k, C_l) = \sum_{\substack{x_i \in C_k \\ x_j \in C_l}} W_{ij}$$

– cut une mesure quantifiant la séparabilité des groupes C_1, ..., C_k et défini par :

$$cut(C_1, \ldots, C_{n_k}) = \frac{1}{2} \sum_{k=1}^{n_k} W(C_k, \mathcal{X} \setminus C_k)$$

Minimiser ce critère selon $C = \{C_1, \ldots, C_{n_k}\}$ revient à déterminer le nombre d'arêtes minimal (ou la somme minimale des poids des arêtes) à ôter au graphe afin de déconnecter les n_k groupes.

Les n_k premiers vecteurs propres des *laplaciens* normalisés L_{sym} et L_{rw} associés aux plus petites valeurs propres correspondent à une représentation des individus dans laquelle l'application des K-moyennes permet de résoudre une relaxation du problème de minimisation de la coupure normalisée suivante :

$$\min_C \ Q_{\text{NCUT}} = \min_C \sum_{k=1}^{n_k} \frac{cut(C_k, \mathcal{X} \setminus C_k)}{vol(C_k)} \quad (1.7)$$

Le *clustering* spectral peut donc être vu comme un K-moyennes où les individus sont projetés en paquets d'individus similaires relativement au graphe de similarité construit à partir des données. Si l'on est capable de construire un graphe contenant n_k composantes connexes alors les individus sont projetés en n_k paquets bien séparés car définis uniquement sur une des dimensions de la matrice correspondant aux vecteurs propres du *laplacien* normalisé. Le graphe étant la structure la mieux adaptée pour capturer la topologie des données. Elle permet de retrouver naturellement les zones de fortes densités correspondant à un nombre important d'individus proches. Cette notion de densité est centrale dans le développement des approches basées sur le voisinage qui seront présentées par la suite.

Algorithme 4 SC

ENTRÉES : $\mathcal{X}$, n_k
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : construire G représentant $\mathcal{X}$. Déterminer W et D
2 : construire L_n selon (1.5) ou (1.6)
3 : construire P dont les colonnes sont les n_k premiers vecteurs propres
4 : si $L_n = L_{sym}$ alors re-normaliser les lignes de P (somme à 1)
5 : $C =$ *clustering* des lignes de P par KM

1.3.2 Approches basées sur la densité

1.3.2.1 DBSCAN : *clustering* basé sur la densité

Un des premiers algorithmes dont l'objectif est explicitement de capturer les zones de fortes densités, définissant ainsi un groupe, est DBSCAN [Ester et al., 1996]. Il s'agit d'une approche exclusivement algorithmique qui se fonde sur une modélisation particulière du concept de zone dense, et qui parcourt l'ensemble des individus afin de déterminer si ceux-ci appartiennent ou non à une telle zone.

Algorithme

DBSCAN nécessite pour être applicable deux paramètres : ϵ et $MinPts$. Ces paramètres globaux déterminent la manière de trouver les groupes en définissant une topologie, puis en proposant une approche constructive basée sur celle-ci. On distingue à partir de ces paramètres deux familles d'individus, des individus *cœur*, et des individus *frontière*. Un individu x_i est qualifié de *cœur* si il contient dans son voisinage de longueur ϵ au moins $MinPts$ points, sinon il s'agit d'un individu frontière. Le voisinage d'un individu x_i est défini par :

$$\mathcal{N}_\epsilon(x_i) = \{x_j \in \mathcal{X} | d(x_i, x_j) \leq \epsilon\}$$

alors x_i est *cœur* si $|\mathcal{N}_\epsilon(x_i)| \geq MinPts$ et frontière sinon. L'algorithme DBSCAN (cf. algorithme 5) procède alors par un parcours de l'ensemble des individus $\mathcal{X}$ jusqu'à rencontrer un individu *cœur* x_i, dès lors il devient générateur d'un groupe. Les voisins de x_i n'appartenant à aucun groupe sont alors affectés au même groupe que x_i. Les nouveaux individus ainsi reaffectés, si ils sont *cœurs*, propagent la génération du groupe selon le même principe.

Enfin, lorsque le groupe en construction ne peut plus s'étendre, il est alors complètement défini *in extenso* par l'ensemble des individus qui auront été parcourus durant ce processus récursif. Cette opération est répétée pour les individus restant de telle sorte à constituer un ensemble de groupes denses. Les individus qui sont de type frontière et qui ne sont pas dans le voisinage d'un individu de type *cœur* sont considérés comme du bruit, des individus mal définis ou des *outliers* (individus atypiques isolés dans l'espace de représentation). Nous désignons l'ensemble de tels individus par $\mathcal{R}$.

Soient les définitions suivantes :

Atteignabilité directe : x_j est directement atteignable en densité à partir de x_i si $x_j \in \mathcal{N}_\epsilon(x_i)$ et x_i est un individu *cœur* ;

Atteignabilité : x_j est atteignable en densité à partir de x_i si x_j est directement atteignable en densité à partir de x_i ou si $\exists x_k \in \mathcal{X}$ et x_k est un individu *cœur* tel que x_j est directement atteignable en densité à partir de x_k et x_k est atteignable en densité à partir de x_i. On notera alors

$$\mathcal{A}(x_i) = \{x_j \in \mathcal{X} | x_j \text{ est atteignable par } x_i\}$$

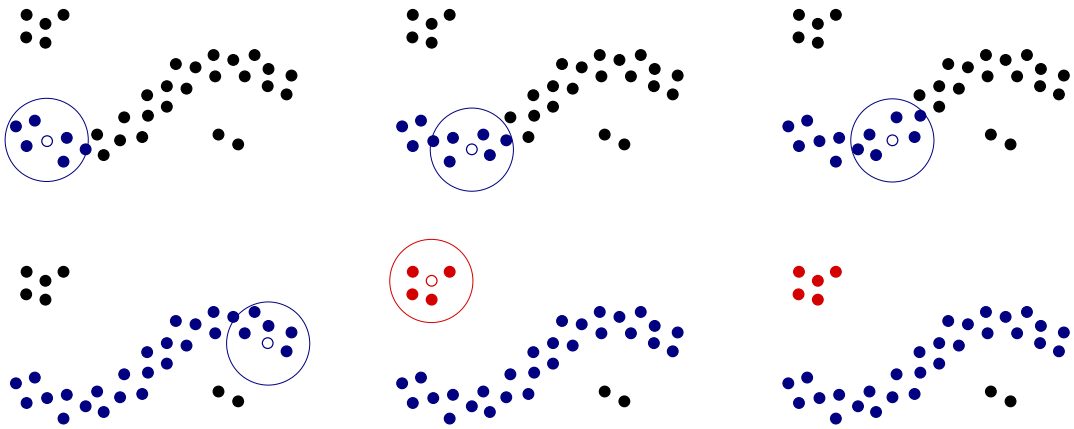

FIGURE 1.4 — Illustration des étapes de DBSCAN pour un voisinage de $MinPts = 4$ individus et $\epsilon = rayon\ du\ cercle$ fixés.

Chaque groupe est alors généré par un individu *cœur* x_i, et contient l'ensemble des individus atteignables en densité à partir de x_i.

Algorithme 5 DBSCAN

ENTRÉES : $\mathcal{X}$, $MinPts$, ϵ
SORTIES : $C = \{C_1, ..., C_{n_k}\}$, $\mathcal{R}$
1 : $i = 1$, $k = 1$ et $\mathcal{R} = \emptyset$
2 : $C_k = \emptyset$
3 : Tant que $|\mathcal{N}_\epsilon(x_i)| < MinPts$ et $x_i \notin \bigcup_{1 \leq g \leq k} C_g$ Faire $i{+}{+}$, $\mathcal{R} = \mathcal{R} \cup \{x_i\}$
4 : $C_k = C_k \cup \{x_i\} \cup \mathcal{A}(x_i)$
5 : Si $\exists x_j \in \mathcal{X}$ tel que $x_j \notin \bigcup_{1 \leq g \leq k} C_g \cup \mathcal{R}$ alors $k{+}{+}$ et aller en **2**.

DBSCAN présente de nombreux avantages, comme la détection automatique du nombre n_k de groupes et la détection des éléments atypiques ou *outliers*. L'approche permet de plus de capturer des groupes de formes variées et impossibles à retrouver avec des algorithmes de partitionnement classiques tels que KM. Mais ces avantages ont un prix, celui du choix des paramètres ϵ et $MinPts$ qui sont difficiles à estimer *a priori*. Cependant les auteurs ont proposé une approche heuristique pour déterminer une bonne valeur de ϵ à partir de $MinPts$ fixé.

1.3.2.2 SOM : les cartes auto-organisatrices

Les cartes auto-organisatrices [Kohonen, 1988] constituent une famille d'algorithmes d'apprentissage réalisant un *clustering* des individus en tenant compte de la topologie présente dans les données. Le principe est de faire évoluer un ensemble de prototypes (appelés aussi neurones) liés entre eux au moyen d'un graphe G qui représente une hypothèse topologique (souvent une grille) sur ces derniers. Le nombre de prototypes, prédéfini, doit être plus grand que le nombre de groupes supposé, ainsi le surnombre de prototypes permet de capturer la forme des groupes.

Objectif

L'objectif visé est que l'ensemble des prototypes approxime la distribution naturelle des individus dans l'espace. La stabilité de la carte topologique est obtenue comme l'optimum du critère objectif suivant:

$$\underset{c}{min}\ Q_{\text{SOM}}(c) = \underset{c}{min} \sum_{k=1}^{n_k} \sum_{x_i \in \mathcal{X}} K(c_k, f^*(x_i))||x_i - c_k||_2^2$$

où $c_k \in \mathbb{R}^p$ est le k-ième prototype. L'idée est alors proche, dans l'esprit, de KM où l'on va chercher à déplacer les prototypes, de sorte à minimiser l'inertie des individus autour de ceux-ci. L'inertie est pondérée par une fonction K quantifiant, pour un terme de l'inertie donné (en fixant k et i), une similarité entre le prototype concerné c_k et le prototype le plus représentatif de l'individu concerné $f^*(x_i)$.

Algorithme

L'algorithme consiste à trouver un moyen de déterminer automatiquement une valeur optimale de similarité $K(c_k, f^*(x_i))$ et d'en déduire naturellement les mises à jours optimales des prototypes, entraînant leur déplacement. Pour cela, le prototype $f^*(x_i)$ est déterminé par :

$$f^*(x_i) = \underset{c \in \{c_1, \ldots, c_{n_k}\}}{arg\ min} ||x_i - c||_2^2 \tag{1.8}$$

La similarité K est définie formellement par :

$$K(c_i, c_j) = \frac{1}{\lambda(t)} \times e^{-\frac{||c_i - c_j||_1}{\lambda^2(t)}}$$

La norme L_1 associée à l'espace G entre c_i et c_j correspondant à une distance géodésique sur cet espace dans lequel sont définis uniquement les prototypes. Plus le prototype c_j est proche du prototype c_i, plus la valeur de $K(c_i, c_j)$ sera élevée. Ainsi, dans le critère on cherche davantage à rapprocher un prototype c_k d'un individu x_i si $c_k = f^*(x_i)$, la similarité correspondante $K(c_k, f^*(x_i))$ étant maximale : $K(c_k, f^*(x_i)) = \frac{1}{\lambda(t)}$.

Pour des raisons de convergence, l'expression de la mesure de similarité K évolue au cours du déroulement itératif de l'algorithme, jusqu'à devenir une mesure quasi-binaire. Cette évolution se fait par l'intermédiaire du paramètre λ dépendant de l'étape d'itération t. Ce paramètre est mis à jour de façon heuristique par :

$$\lambda(t) = \lambda_i (\frac{\lambda_f}{\lambda_i})^{\frac{t}{t_{max}}}$$

où λ_i et λ_f sont des bornes définies *a priori*.

Enfin, les prototypes sont mis à jour par une recherche linéaire (pondérée par $K(c_k, f^*(x_i))$, qui lui n'est pas linéaire en c_k) :

$$c_k^* = c_k - \epsilon(t) K(c_k, f^*(x_i))(x_i - c_k) \tag{1.9}$$

où $\epsilon(t)$ est un pas d'optimisation variable qui diminue avec le temps pour garantir la convergence.

L'algorithme SOM existe sous différentes formes. Dans l'approche initiale, la carte est mise à jour pour chaque présentation d'un individu x_i par la règle (1.9) après avoir déterminé son prototype représentant par (1.8). L'algorithme 6 relate une version dite *batch* pour laquelle la carte est mise à jour de manière itérative une fois que tous les individus lui sont présentés, davantage dans l'esprit de KM.

Algorithme 6 *batch* SOM

ENTRÉES : $\mathcal{X}$, n_k, λ_i, λ_f, G
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : $t = 1$ et $\lambda(t) = \lambda_i$
2 : initialiser aléatoirement les n_k prototypes $\{c_1, ..., c_{n_k}\}$
3 : mise à jour de $f^*_{x_i}$ $\forall x_i \in \mathcal{X}$ selon (1.8)
4 : mise à jour des prototypes c_k $\forall k \in [1..n_k]$ selon (1.9)
5 : si $\lambda(t) > \lambda_f$ alors $t = t + 1$ et aller en **3**.
6 : $C_k = \{x_i \in \mathcal{X} | f^*(x_i) = c_k\}$ $\forall k \in [1..n_k]$

1.4 Approches floues et probabilistes

Il peut arriver, au cours du processus itératif ou à la fin, qu'un individu soit difficile à classer car proche simultanément de plusieurs groupes. La prise de décision faite par les approches par partitionnement présentées précédemment est d'affecter l'individu au groupe le plus proche en oubliant les autres. Une vision plus naturelle est alors d'adoucir cette décision et de maintenir l'incertitude sur l'appartenance d'un individu aux groupes le plus longtemps possible. Cela peut permettre d'éviter tant que possible de s'enraciner trop rapidement vers une solution qui s'avèrerait peu satisfaisante, par exemple, un optimum local dans le cas des approches discriminatives. L'incertitude lors du *clustering* peut être modélisée de différentes façons, les plus courantes consistant à utiliser la théorie des ensembles flous ou bien la théorie des probabilités.

Dans le cadre des ensembles flous [Zadeh, 1965], on considère en général que chaque individu appartient simultanément à tous les groupes mais avec un certain degré d'appartenance. En ce qui concerne les approches probabilistes [Dempster et al., 1977], nous considérons qu'un individus appartient à un seul groupe, qui correspond au groupe le plus probable, mais une probabilité non nulle existe concernant l'évènement d'appartenance à chacun des autres groupes.

1.4.1 FKM : les K-moyennes floues

L'approche discriminative des K-moyennes floues, notée FKM, développée par [Bezdek, 1981] est une généralisation de K-moyennes se basant sur des éléments de la théorie des ensemble flous.

Objectif

Le principe est toujours de minimiser la dispersion des individus relativement aux prototypes, mais pondérée cette fois par le degré d'appartenance de l'individu au groupe. Du point de vue du critère objectif, on présente les K-moyennes floues de la manière suivante comme la minimisation du critère de l'erreur quadratique semblable à KM, mais évaluée pour chaque individu relativement à l'ensemble des prototypes :

$$\begin{array}{rrcl} \min\limits_{c,u} \; Q_{\text{FKM}}(c,u) = & \min\limits_{c,u} \sum\limits_{k=1}^{n_k} \sum\limits_{x_i \in \mathcal{X}} u_{ik}^{\beta} ||x_i - c_k||_2^2 & & \\ s.t. & \sum_{k=1}^{n_k} u_{ik} & = & 1 \quad \forall x_i \in \mathcal{X} \\ & u_{ik} & \geq & 0 \quad \forall x_i \in \mathcal{X},\ \forall k \in [1..n_k] \end{array} \tag{1.10}$$

où $\beta \geq 1$ est un paramètre fixé dans l'objectif et c_k est le prototype du groupe C_k. $u = \{u_{ik}\}$ est l'ensemble des degrés d'appartenance des individus aux groupes. En particulier, u_{ik} indique le degré d'appartenance de l'individu x_i au groupe C_k.

Intuitivement, plus un individu à un moment donné sera proche d'un prototype relativement aux autres, plus son degré d'appartenance à celui-ci sera élevé. Au final, le résultat n'est pas une décision sur l'appartenance d'un individu à un groupe particulier, mais un ensemble d'indicateurs permettant de mesurer l'incertitude sur le groupe auquel appartient cet individu. La solution du problème d'optimisation, l'optimum, correspond à un ensemble de prototypes les plus représentatifs des groupes ainsi que la matrice d'appartenance des individus aux groupes. Cet optimum satisfait les conditions d'optimalité du premier ordre du Lagrangien associé au problème d'optimisation sous contrainte. Comme le critère objectif est convexe lorsque l'une des variables du problème d'optimisation est fixée, on peut obtenir alternativement les mises à jours globalement optimales des degrés d'appartenance pour des centres fixés :

$$u_{ik}^* = \frac{||x_i - c_k||_2^{2/(1-\beta)}}{\sum_{j=1}^{n_k} ||x_i - c_j||_2^{2/(1-\beta)}} \; \forall x_i \in \mathcal{X}, \forall k \in [1..n_k] \tag{1.11}$$

De la même manière, on obtient les centres globalement optimaux relativement aux degrés d'appartenance de la manière suivante :

$$\begin{aligned} c_k^* &= \underset{c \in \mathbb{R}^p}{arg\,min} \sum_{x_i \in \mathcal{X}} u_{ik}^\beta ||x_i - c||_2^2 \\ &= \frac{\sum_{x_i \in X} u_{ik}^\beta x_i}{\sum_{x_i \in X} u_{ik}^\beta} \end{aligned} \tag{1.12}$$

Algorithme

Du point de vue de l'algorithme (cf. Algorithme 7), à la manière de KM, il s'agit également d'un processus itératif, semblable à la résolution d'un système d'équations (mise à jour des centres, et mise à jour des degrés d'appartenances) par une méthode itérative de type *Gauss-Seidel*, qui va alterner cette fois une phase de mise à jour des degrés d'appartenance des individus aux classes et une phase de mise à jour des centres de classes (après une initialisation aléatoire des centres de classes), jusqu'à une stabilisation numérique.

Algorithme 7 FKM

ENTRÉES : $\mathcal{X}$, n_k, β
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation aléatoire des n_k centres de groupes $\{c_1, ..., c_{n_k}\}$
2 : Mise à jour des degrés d'appartenances u_{ik} $\forall x_i \in \mathcal{X}$, $\forall k \in [1..n_k]$ en utilisant (1.11)
3 : Mise à jour des centres de groupe c_k $\forall k \in [1..n_k]$ en utilisant (1.12)
4 : Si Q_{FKM} change alors aller en **2**
5 : $C_k = \{x_i \in \mathcal{X} | u_{i,k} = \underset{k' \in [1..n_k]}{max} u_{ik'}\}$ $\forall k \in [1..n_k]$

Cette fois, le résultat n'est pas une partition stricte, mais une partition floue, ce qui ne nous dit pas à quel groupe appartient un individu. Pour répondre au problème du *clustering* originel, il est nécessaire d'ajouter une étape d'affectation finale (étape 5 dans l'algorithme) des individus aux groupes, à appliquer à l'issue de l'algorithme. La procédure choisie consiste à affecter les individus aux groupes pour lesquels ils ont le plus fort degré d'appartenance.

$$x_i \in C_k \Leftrightarrow k = \underset{k' \in [1..n_k]}{arg\,max}\, u_{ik'}$$

Cette généralisation de KM est toujours formulée comme un problème d'optimisation non convexe selon l'ensemble des variables correspondant aux centres et aux degrés. Ainsi, aucune garantie n'existe concernant l'optimalité globale de la solution, et il convient également dans ce cadre de relancer plusieurs fois l'algorithme. Cependant, empiriquement, FKM est beaucoup plus stable que son analogue strict.

1.4.2 EM : estimation d'un mélange de modèles par Espérance-Maximisation

L'autre outil des mathématiques qui permet de capturer et tenir compte d'une forme d'incertitude sur les classements des individus au sein des groupes est la théorie des probabilités. Dans le cadre du *clustering*, le modèle qui prédomine est celui des mélanges de lois. On suppose toujours que n_k groupes existent, et chaque groupe est représenté par une loi de probabilité paramétrée. Il existe de nombreuses lois de probabilité, mais en général, la loi normale est utilisée, car elle permet de représenter la plus grande majorité de phénomènes, et elle approxime également nombre d'autres lois. On considère alors que l'ensemble des individus $\mathcal{X}$, appelé également échantillon dans ce contexte, suit un mélange de n_k lois paramétrées f. La k-ième loi du mélange, caractérisée par sa fonction de densité f_k est paramétrée par θ_k ainsi qu'une probabilité *a priori* α_k de générer l'ensemble des individus. La tâche de *clustering* est alors de chercher quelles sont les lois (les paramètres des lois) qui permettent au mieux d'expliquer la génération de l'échantillon d'individus $\mathcal{X}$. En d'autres termes, trouver les meilleurs estimateurs des paramètres $\Theta = \{(\alpha_k, \theta_k)\}_{k\in[1..n_k]}$.

Modèle

On associe à chaque composante du mélange (chaque loi) une valeur de probabilité α_k *a priori*, exprimant la probabilité que la k-ième loi soit sélectionnée pour générer chaque individu x_i, que l'on appelle aussi proportion du mélange. Soit X_i les variables aléatoires dont les x_i sont des réalisations, le mélange associé aux n_k lois est alors le suivant:

$$f(X_i;\Theta) = \sum_{k=1}^{n_k} \alpha_k f_k(x_i;\theta_k) \tag{1.13}$$

et le modèle expliquant la génération de l'échantillon $\mathcal{X}$ sous l'hypothèse d'une distribution identique et indépendante des variables X_i s'exprime :

$$f(X;\Theta) = f(X_1,...,X_n;\Theta) = \prod_{i=1}^{n}\sum_{k=1}^{n_k} \alpha_k f_k(x_i;\theta_k) \tag{1.14}$$

Objectif

Maintenant que le modèle est défini, on peut formaliser l'objectif du *clustering* associé. Celui-ci consiste à chercher les paramètres des lois qui maximisent la vraisemblance et, de manière équivalente mais plus adaptée d'un point de vue computationnel, la log-vraisemblance des données complétées par un vecteur aléatoire Z indiquant pour chaque individu x_i, le groupe auquel il semble appartenir ($Z_i = k \Leftrightarrow x_i \in C_k$). La log-vraisemblance $\mathcal{L}$ des paramètres Θ s'exprime par :

$$\mathcal{L}(\Theta;\mathcal{X},Z) = \sum_{x_i\in\mathcal{X}}\sum_{k=1}^{n_k} z_{ik}\log(\alpha_k f_k(x_i;\theta_k))$$

où z_{ik} représente la probabilité *a posteriori* que l'individu x_i ait été généré par la k-ième composante du mélange, selon la valeur de Θ courante notée Θ^-. Le problème de maximisation de

la log-vraisemblance des paramètres relativement à l'observation des données (l'échantillon) $\mathcal{X}$ complétées par le vecteur Z est alors équivalent au problème de maximisation du critère Q_{EM} décrit par:

$$\begin{aligned} \max_{\Theta} Q_{\text{EM}}(\Theta; \Theta^{-}, \mathcal{X}, n_k) = \\ \max_{\Theta} \sum_{x_i \in \mathcal{X}} \sum_{k=1}^{n_k} f(Z_i = k | X_i = x_i; \Theta^{-}) \log(\alpha_k f_k(x_i; \theta_k)) \end{aligned} \tag{1.15}$$

Algorithme

L'algorithme employé pour obtenir l'optimum de ce critère est EM [Dempster et al., 1977]. Cette approche est destinée à estimer les paramètres de n'importe quel modèle statistique, mais son utilisation est ici restreinte à l'estimation des paramètres du mélange de lois. Partant d'une initialisation des paramètres Θ, l'algorithme propose de maximiser la log-vraisemblance des données complétées en alternant deux étapes qui sont :

1. le calcul de l'espérance de la variable caché Z_i permettant d'obtenir une mise à jour des valeurs de probabilités *a posteriori* permettant d'évaluer l'espérance de la log-vraisemblance selon la valeur courante des paramètres Θ. Ainsi, la variable z_{ik} est calculée par :

$$\begin{aligned} z_{ik}^{*} &= f(Z_i = k | X_i = x_i; \Theta) \\ &= \frac{\alpha_k f_k(x_i; \theta_k)}{\sum_{k'=1}^{n_k} \alpha_{k'} f_{k'}(x_i; \theta_{k'})} \end{aligned} \tag{1.16}$$

2. la maximisation du critère Q_{EM} selon θ et conditionnellement à la valeur courante des probabilités *a posteriori* z_{ik} :

$$\begin{aligned} \theta^{*} &= \arg\max_{\theta} \mathcal{L}(\Theta; \mathcal{X}, Z) \\ &= \arg\max_{\theta} \sum_{x_i \in \mathcal{X}} \sum_{k=1}^{n_k} z_{ik} \log(\alpha_k f_k(x_i; \theta_k)) \end{aligned} \tag{1.17}$$

Lorsque les lois sont des lois normales multi-dimensionnelles $f_k \sim \mathcal{N}(c_k, \Sigma_k)$ où c_k est la moyenne et Σ_k est la matrice de variances/covariances, alors la *pdf* f_k est définie, pour des x_i vecteurs lignes, par :

$$f_k(x_i; \theta_k) = \frac{1}{(2\pi)^{k/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x_i - c_k)\Sigma^{-1}(x_i - c_k)^{\top}}$$

La connaissance de la nature des lois permet de déterminer explicitement les formules de mise à jour des paramètres (c_k, Σ_k) $\forall k \in [1..n_k]$. Ainsi, dans le cas du mélange gaussien, on a :

$$\begin{aligned} c_k^{*} &= \frac{\sum_{x_i \in \mathcal{X}} (z_{ik} x_i)}{\sum_{x_i \in \mathcal{X}} z_{ik}} \\ \Sigma_k^{*} &= \frac{\sum_{x_i \in \mathcal{X}} \left(z_{ik} (x_i - c_k)^{\top} (x_i - c_k) \right)}{\sum_{x_i \in \mathcal{X}} z_{ik}} \end{aligned}$$

Enfin, les probabilités *a priori* sont également réestimées par :

$$\alpha_k = \frac{1}{n} \sum_{x_i \in \mathcal{X}} z_{ik} \tag{1.18}$$

Algorithme 8 EM

ENTRÉES : $\mathcal{X}$, n_k, f
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1: Initialisation aléatoire des n_k paramètres $\{\Theta_1, ..., \Theta_{n_k}\}$
2: Étape E : Mise à jour des z_{ik}, $\forall x_i \in \mathcal{X}$, $\forall k \in [1..n_k]$ en utilisant (1.16)
3: Étape M : Mise à jour des θ_k $\forall k \in [1..n_k]$ en utilisant (1.17)
4: Mise à jour des α_k $\forall k \in [1..n_k]$ en utilisant (1.18)
5: Si Q_{EM} change alors aller en **2**
6: $C_k = \{x_i \in \mathcal{X} | z_{ik} = \max_{k' \in [1..n_k]} z_{ik'}\}$ $\forall k \in [1..n_k]$

De la même manière que pour FKM, le résultat de l'algorithme n'est pas une partition stricte. On peut néanmoins en obtenir une en appliquant la règle MAP, du maximum *a posteriori*, qui consiste à affecter un individu x_i au groupe C_k si cet individu a le plus de chance d'avoir été généré par la k-ième composante du mélange, soit :

$$x_i \in C_k \Leftrightarrow k = \arg\max_{k' \in [1..n_k]} z_{ik'}$$

ce qui constitue l'étape **6** de l'algorithme EM pour le *clustering*.

Le modèle de mélange et l'algorithme EM offrent un atout de poids comparée aux autres approches présentées précédemment. En effet celui-ci est générique du point de vue de l'hypothèse faite sur la nature des distributions du mélange expliquant la génération de l'échantillon $\mathcal{X}$. Ainsi nous pouvons utiliser différents type de lois pour modéliser les groupes (lois gaussiennes, multinomiales, poisson, *etc.*), l'algorithme reste le même, seul change le calcul explicite de la mise à jour des paramètres du modèle.

1.5 Bilan

1.5.1 Les liens entre familles d'algorithmes de *clustering*

Les algorithmes présentés constituent un ensemble non exhaustif d'approches classiques pour le *clustering* dédié aux données conventionnelles. Bien d'autres approches existent parmi ces familles d'algorithmes, et la plupart des approches détaillées ont été étendues. De même d'autres familles de méthodes existent, comme les méthodes basées sur :

- les grilles [Gan et al., 2007b] ;
- la factorisation de matrices non négatives noté NMF [Ding et al., 2005], [Li, 2008] ;
- les exemples (les individus) et le passage de messages entre eux [Frey and Dueck, 2007], [Lashkari and Golland, 2008].

La dernière de ces familles offre de belles perspectives et de la nouveauté concernant la modélisation de l'objectif du *clustering*, que l'on peut qualifier de *micro*, car se basant uniquement sur les individus et les interactions possibles entre eux pour former une organisation globale. Les autres familles sont plutôt *macro* et on définit en général un modèle global de groupes auquel on cherche à conformer l'ensemble des individus. On dira dans le cas général, qu'une famille est gouvernée par un paradigme qui correspond à une théorie majoritairement employée pour

résoudre l'objectif posé. L'algèbre linéaire est majoritairement présente dans les approches classiques par partitionnement, de même que l'algorithmique et la théorie des graphes l'est pour les approches basées sur le voisinage et la recherche de groupes denses. Enfin la théorie des probabilités et la statistique offrent un cadre privilégié pour les approches intégrant l'incertitude pour produire un modèle plus robuste et permettant d'obtenir des partitions plus adaptées et interprétables. Quoiqu'il en soit, ces différents paradigmes convergent parfois et certains travaux participent alors à une unification de différentes approches de *clustering*. On notera le résultat majeur de l'équivalence entre KM et une variante classificatoire de EM [Celeux and Govaert, 1992] pour l'estimation des paramètres d'un mélange de gaussiennes homoscédastiques (de variance constante pour tous les groupes et pour toutes les dimensions à l'intérieur de ces groupes). La variante classificatoire de EM consiste simplement à appliquer la règle MAP à chaque étape de l'algorithme, remplaçant ainsi, pour chaque individu x_i le vecteur des probabilités *a posteriori* par un vecteur indicateur où un unique 1 indique quelle est la composante du mélange ayant le plus de chance de générer x_i une fois les paramètres établis. De même, des travaux récents montrent l'équivalence au sein du cadre théorique des approches NMF, d'un *clustering* par factorisation de matrice non négative et de SC [Ding et al., 2005]. Dans le même esprit, et en utilisant les outils similaires de l'algèbre linéaire, des travaux ont également unifié une généralisation de KM avec plusieurs heuristiques de partitionnement de graphes dans le contexte de SC [Dhillon et al., 2005]. On notera aussi certains travaux qui combinent judicieusement des arguments de divers paradigmes afin d'en exploiter les meilleurs parts comme l'approche des graphes gaussiens génératifs [Aupetit, 2006] qui permet de capturer à la fois la topologie des données et de plus de donner une interprétation statistique des résultats.

Les travaux d'unification en *clustering* sont très importants car ils aident à réorganiser les recherches dans cette thématique où la production scientifique est parmi les plus prolifiques et où il est difficile de suivre en temps réel l'intégralité des approches proposées [Jain, 2008]. Après cette perspective positive des travaux autour du *clustering*, nous nous intéressons dorénavant à certains points qui restent satellites autour du *clustering* mais qui constituent des problématiques de recherche à eux seul pour enrichir les techniques de classification non supervisée :

- les paramètres types des approches de *clustering* que sont le nombre de groupes, et également dans une certaine mesure le choix de la mesure de proximité, et la capacité des approches présentées précédemment à tenir compte d'autres mesures que celles pour lesquelles elles ont été développées (en général la norme L_2) ;
- le problème d'évaluer ce qu'est une bonne partition de l'ensemble d'individus $\mathcal{X}$. En effet ce point est central et à l'heure actuelle, personne n'est encore capable de définir une mesure d'évaluation d'une bonne partition universelle et absolue, hormis l'évaluation par un expert dans un contexte complètement applicatif ;
- le problème du choix de l'algorithme dès lors que l'on est confronté à un ensemble d'individus que l'on cherche à regrouper, sans hypothèses ou expertises supplémentaires.

1.5.2 Le problème du nombre de groupes

Le premier problème est relatif principalement aux approches par partitionnement strict ou floue en un nombre de groupes fixé. Dans un cadre complètement non supervisé, aucune connaissance sur ce nombre de groupes n'est disponible et celui-ci doit automatiquement être appris à partir des données. Une première approche consiste à appliquer un même algorithme pour différentes valeurs du nombre de groupes n_k et retenir celui pour lequel la valeur du critère objectif est optimale. Ceci est valable pour les approches où la fonction objectif est connue, par exemple le critère inertiel de KM. Le principal problème de cette procédure est que dans la plupart des cas, le nombre de groupes optimal tend à produire une solution dégénérée. Le nombre de groupes pour obtenir un *clustering* optimal de $\mathcal{X}$ au sens du critère Q_{KM} par KM est

$|\mathcal{X}|$ *i.e.* chaque individu constitue son propre groupe. La même remarque prévaut lors de l'observation du critère de maximum de log-vraisemblance dans le cadre des modèles de mélange pour un nombre de composantes croissant. Afin de pallier à ce genre de problème, des auteurs ont proposé, notamment dans ce dernier cadre, d'intégrer le nombre de composantes comme un paramètre du modèle, puis de pénaliser le critère classique de log-vraisemblance pour des paramètres Θ^* optimaux, par une fonction des degrés de libertés du nombre de groupes, traduisant la complexité du modèle au sens de la *Statistique*. Ainsi, dans l'exemple des modèles de mélange, si Θ^* correspond aux paramètres optimaux du critère de log-vraisemblance $\mathcal{L}(\Theta; \mathcal{X}, Z)$ et N_k est la variable aléatoire associée au nombre de groupes, alors plusieurs mesures de la log-vraisemblance pénalisée $\mathcal{L}(N_k)$ peuvent être suggérées :

– le critère d'information de Akaike AIC [Aikake, 1973] :

$$\mathcal{L}(N_k) = 2dl(N_k) - 2\mathcal{L}(\Theta^*; \mathcal{X}, Z)$$

– le critère d'information bayésienne BIC [Schwarz, 1978] :

$$\mathcal{L}(N_k) = \ln(n)dl(N_k) - 2\mathcal{L}(\Theta^*; \mathcal{X}, Z)$$

où $dl(N_k)$ correspond au degré de liberté de N_k et est déterminé par le nombre de paramètres nécessaires pour estimer la log vraisemblance $\mathcal{L}(\Theta^*; \mathcal{X}, Z)$.

Ces critères constituent le socle des différentes approches de *sélection de modèles* en *Statistique*, qui consiste à prendre, parmi une population de modèles (par exemple, parmi les modèles de mélange de nombre de composantes différentes) celui qui est le plus en adéquation avec les observations. Pour finir, ils permettent d'éviter le sur-apprentissage induit par l'augmentation du nombre de composantes du mélange en trouvant un *bon* compromis. D'autres techniques enfin proposent de ne pas pénaliser le critère de vraisemblance classique, mais de repérer une faible variabilité, statistiquement significative du critère de vraisemblance entre deux valeurs de n_k données [Biernacki, 2009]. Ces méthodes, utilisables quelquesoit l'algorithme de *clustering* formant une partition en n_k groupes en adaptant le critère, se dénomment plus communément les méthodes du coude.

1.5.3 Le problème de l'évaluation

L'évaluation d'un résultat de *clustering* est toujours un problème ouvert, car on ne connaît pas toujours l'étiquette des individus. On ne peut en général pas se comparer à une classification de référence correspondant aux classes des individus que l'on aimerait retrouver par l'approche de *clustering* employée. Cependant, même lorsqu'une telle classification cible existe, de multiples moyens existent pour effectuer la comparaison. Les différents critères d'évaluation sont présentés en trois familles :

– les critères internes n'exploitant aucune classification de référence ;
– les critères externes visant à quantifier l'écart ou la similarité entre le *clustering* produit et la classification de référence ;
– les critères subjectifs, car relatifs à un algorithme ou une famille d'algorithmes particuliers.

Les critères internes et les critères subjectifs ne seront pas présentés dans la mesure où les approches proposées ont systématiquement été évaluées *via* une classification de référence. Cependant, leurs descriptions peuvent être trouvées en détail dans [Gan et al., 2007a].

Lorsque toutes les étiquettes de classes sont disponibles, on peut utiliser un critère d'évaluation externe mesurant l'adéquation entre la classification obtenue C par l'algorithme de *clustering* et la classification de référence $\mathcal{C}$. De nombreuses méthodes existent et nous relaterons ici celles qui ont été utilisées pour valider les différentes contributions, ainsi que celles qui participent au cœur de quelques approches qui seront développées par la suite dans les états de l'art spécifiques à chaque problématique traitée.

1.5.3.1 Mesures basées sur l'énumération

Soient M le nombre de paires d'individus et tp, fp, tn, fn les nombres de *vrais-positifs*, *faux-positifs*, *vrais-négatifs*, *faux-négatifs* tels que :

- tp est le nombre de paires d'individus regroupés à la fois dans le clustering C et dans la classification de référence $\mathcal{C}$;
- fp est le nombre de paires d'individus regroupés dans le clustering C mais non dans la classification de référence $\mathcal{C}$;
- tn est le nombre de paires d'individus dans des groupes différents dans C (le *clustering* obtenu) et dans $\mathcal{C}$ (la classification de référence);
- fn est le nombre de paires d'individus dans des groupes différents dans C mais ensemble dans la classification de référence $\mathcal{C}$.

La relation liant M, tp, fp, tn et fn est :

$$M = tp + fp + tn + fn = \frac{n(n-1)}{2}$$

Indice de Rand. L'indice de Rand est obtenu en observant la proportion de paires d'individus classés de la même manière dans C et dans $\mathcal{C}$:

$$\text{Rand}(C, \mathcal{C}) = \frac{tp + tn}{M} \tag{1.19}$$

Indice de Jaccard. L'indice de Jaccard s'exprime comme le nombre de paires correctement regroupés sur le nombre de paires d'individus identifiés ensemble dans C ou dans $\mathcal{C}$:

$$\text{Jaccard}(C, \mathcal{C}) = \frac{tp}{M - tn} \tag{1.20}$$

F-mesure. La F-mesure combine précision et rappel sur les paires d'individus. La précision reflète la proportion de paires correctement identifiées sur le nombre de paires d'individus retrouvées dans C :

$$\text{Précision}(C, \mathcal{C}) = \frac{tp}{tp + fp}$$

Le rappel correspond à la proportion de paires correctement identifiées par rapport au nombre de paires d'individus classés ensemble dans $\mathcal{C}$:

$$\text{Rappel}(C, \mathcal{C}) = \frac{tp}{tp + fn}$$

La F-mesure est alors une mesure mélangeant linéairement les deux critères par :

$$\text{F-mesure}(C, \mathcal{C}, \beta) = \frac{(\beta^2 + 1) \times \text{Précision}(C, \mathcal{C}) \times \text{Rappel}(C, \mathcal{C})}{\beta^2 \times \text{Précision}(C, \mathcal{C}) + \text{Rappel}(C, \mathcal{C})} \tag{1.21}$$

Les indices de Rand, Jaccard et la F-mesure ont des valeurs d'autant plus fortes que le *clustering* obtenu est de bonne qualité relativement à la classification de référence.

1.5.3.2 Mesures statistiques basées sur l'entropie.

Soient α_k, α_c et α_{ck} les nombres d'individus respectivement dans le groupe C_k, dans la classe $\mathcal{C}_c$ et dans l'intersection de C_k et $\mathcal{C}_c$:

$$\begin{aligned} \alpha_k &= \frac{|C_k|}{n} \\ \alpha_c &= \frac{|\mathcal{C}_c|}{n} \\ \alpha_{ck} &= \frac{|C_k \cap \mathcal{C}_c|}{n} \end{aligned}$$

Les différentes mesures suivantes visent à quantifier l'information semblable dans le clustering produit C et la classification de référence $\mathcal{C}$.

Entropie moyenne. Soit $H(C_k, \mathcal{C}_c)$ l'entropie d'information conjointe du groupe C_k et de la classe $\mathcal{C}_c$:

$$H(C_k, \mathcal{C}_c) = -\alpha_{ck} \times \log(\alpha_{ck}) \tag{1.22}$$

L'entropie d'information moyenne $AvgEnt$ utilise les étiquettes de classes pour calculer la moyenne de l'impureté de chaque groupe pondérée par la taille de ceux-ci:

$$\text{AvgEnt}(C, \mathcal{C}) = \sum_{k=1}^{n_k} \alpha_k \left(\sum_{c=1}^{n_c} H(C_k, \mathcal{C}_c) \right)$$

On appelle également information jointe entre C_k et $\mathcal{C}_c$ notée $I(C_k, \mathcal{C}_c)$, quantité négative correspondante à la négentropie conjointe :

$$I(C_k, \mathcal{C}_c) = -H(C_k, \mathcal{C}_c)$$

Information mutuelle. L'information mutuelle normalisée quantifie l'information statistique partagée entre deux distributions (par exemple les distributions des étiquettes de groupes et des étiquettes de classes), elle peut être définie *via* la mesure d'entropie.

Soit $H(C, \mathcal{C})$ l'entropie conjointe des partitions C et $\mathcal{C}$:

$$H(C, \mathcal{C}) = \sum_{k=1}^{n_k} \sum_{c=1}^{n_c} H(C_k, \mathcal{C}_c) \tag{1.23}$$

Soit $H(C)$ et $H(\mathcal{C})$ les entropies des partitions C et $\mathcal{C}$:

$$\begin{aligned} H(C) &= -\sum_{k=1}^{n_k} \alpha_k \times \log(\alpha_k) \\ H(\mathcal{C}) &= -\sum_{c=1}^{n_c} \alpha_c \times \log(\alpha_c) \end{aligned}$$

L'information mutuelle normalisée de façon arithmétique s'exprime alors par:

$$NMI(C, \mathcal{C}) = 2 \times \frac{MI}{H(C) + H(\mathcal{C})} \tag{1.24}$$

avec

$$MI(C, \mathcal{C}) = H(C) + H(\mathcal{C}) - H(C, \mathcal{C}) \tag{1.25}$$

Soient p_{C_k} et $p_{\mathcal{C}_c}$ les distributions des individus sur le groupe C_k et sur la classe $\mathcal{C}_c$ respectivement où :

- $p_{C_k}(Z_i = k)$ vaut 1 si $x_i \in C_k$ et 0 sinon (Z_i est la variable correspondant à l'étiquette de x_i dans le *clustering* C_k) ;
- $p_{\mathcal{C}_c}(l(x_i) = c)$ vaut 1 si $x_i \in \mathcal{C}_c$ et 0 sinon ($l(x_i)$ est la variable correspondant à l'étiquette de x_i dans la *classe* $\mathcal{C}_c$).

On appelle également divergence de Kullback-Leibler (KL) entre C_k et $\mathcal{C}_c$ la mesure positive quantifiant la dissemblance entre les distributions des individus sur les groupes p_{C_k} et la distribution des individus sur les classes p_C :

$$KL(p_{C_k} \,||\, p_{\mathcal{C}_c}) = \sum_{i=1}^{n} p_{C_k}(Z_i = k) \times \log\left(\frac{p_{C_k}(Z_i = k)}{p_{\mathcal{C}_c}(l(x_i) = c)}\right) \tag{1.26}$$

qui se généralise pour la mesure de dissimilarité entre le *clustering* C et la classe $\mathcal{C}$ par :

$$KL(C \,||\, \mathcal{C}) = \sum_{k=1}^{n_k} \sum_{c=1}^{n_c} KL(p_{C_k} \,||\, p_{\mathcal{C}_c}) \tag{1.27}$$

Soit $p_{C_k,\mathcal{C}_c}$ la distribution jointe des individus sur l'intersection du groupe C_k et de la classe $\mathcal{C}_c$ avec $p_{C_k,\mathcal{C}_c}(Z_i = k, l(x_i) = c)$ vaut 1 si $x_i \in C_k$ et $x_i \in \mathcal{C}_c$ et 0 sinon. L'information mutuelle peut alors se réécrire comme la divergence de Kullback-Leibler entre la distribution jointe $p_{C_k,\mathcal{C}_c}$ des *clusterings* et des classes, et la distribution jointe sous hypothèse d'indépendance $p_{C_k} \times p_{\mathcal{C}_c}$ entre les *clusterings* et les classes :

$$MI(C, \mathcal{C}) = KL(p_{C_k,\mathcal{C}_c} \,||\, p_{C_k} \times p_{\mathcal{C}_c})$$

Selon (1.24), l'information mutuelle normalisée peut alors être réécrite par :

$$NMI(C, \mathcal{C}) = 2 \times \frac{KL(p_{C_k,\mathcal{C}_c} \,||\, p_{C_k} \times p_{\mathcal{C}_c})}{H(C) + H(\mathcal{C})}$$

L'entropie moyenne a des valeurs d'autant plus faibles que le *clustering* obtenu est en adéquation avec la classification de référence, tout comme la divergence de Kullback-Leibler. À l'opposé, plus la valeur d'information mutuelle est élevée, plus le résultat est conforme à la classification.

1.5.4 Le choix de la proximité

Tout algorithme de *clustering* repose sur une mesure permettant de quantifier la proximité entre deux individus. Dans le cas le plus général, les données correspondent à un ensemble de mesures de type flottant pour chaque individu $x_i \in \mathcal{X}$, ainsi $x_i \in \mathbb{R}^p$. De ce fait la mesure choisie correspond au carré d'une distance, la plupart du temps euclidienne $||.||_2$ qui correspond à la métrique la plus usuelle pour l'espace $\mathbb{R}^p$. Néanmoins, il peut arriver dans diverses applications que les descriptions des individus soient de type symbolique ou catégorielle ou encore que l'on désire utiliser une mesure de proximité ne se comportant pas comme une distance dans l'espace de description de $\mathcal{X}$. Dans de tels cas, on définit de nouvelles mesures dites de similarité ou de dissimilarité ayant chacune des propriétés particulières telles que la minimalité, la symétrie, l'identité ou l'inégalité triangulaire.

Soit $f : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ une fonction de proximité, on définit les propriétés:

minimalité : f vérifie la minimalité *ssi*

$$\forall x_i \in \mathcal{X},\ f(x_i, x_i) = 0$$

maximalité : f vérifie la maximalité *ssi*

$$\forall (x_i, x_j, x_k) \in \mathcal{X}^3,\ f(x_i, x_i) \geq f(x_j, x_k)$$

symétrie : f vérifie la symétrie *ssi*

$$\forall (x_i, x_j) \in \mathcal{X}^2,\ f(x_i, x_j) = f(x_j, x_i)$$

identité : f vérifie l'identité *ssi*

$$\forall (x_i, x_j) \in \mathcal{X}^2,\ f(x_i, x_j) = 0 \Rightarrow x_i = x_j$$

inégalité triangulaire : f vérifie l'inégalité triangulaire *ssi*

$$\forall (x_i, x_j, x_k) \in \mathcal{X}^3,\ f(x_i, x_j) \leq f(x_i, x_k) + f(x_k, x_j)$$

Parmi les diverses familles de proximités existantes :
- une **distance** telle la distance euclidienne $||.||_2$ satisfait la minimalité, la symétrie, l'identité et l'inégalité triangulaire;
- une **dissimilarité** satisfait la minimalité et la symétrie;
- une **similarité** satisfait la maximalité et la symétrie.

1.5.5 Le choix de l'algorithme

Une autre problématique de choix survient, notamment lorsque les informations de proximités sont fixées, et que l'on ne parvient pas à obtenir un *clustering* satisfaisant avec une approche particulière. Ainsi les données correspondent alors à une matrice de similarité, de dissimilarité ou de distance. L'obtention d'une solution différente et plus intéressante pour le praticien des techniques de *clustering* peut se faire par l'application d'un autre algorithme, capable de prendre en compte la matrice de proximité constituant les données. Parmi les familles d'approches présentées, les algorithmes hiérarchiques DIANA et AGNES, ainsi que DBSCAN et SC ne nécessitent pas de modifications majeures pour être applicables. Les autres méthodes sont fondées sur la distance euclidienne et nécessitent d'être étendues pour pouvoir prendre en compte des mesures de similarité afin de garantir les mêmes propriétés (de convergence notamment). Un exemple type d'un tel travail est l'extension de KM en KM à noyau ou KKM [Kulis et al., 2005] qui sera présenté plus en détail par la suite, mais dont l'idée est de définir une mesure de distance euclidienne à partir des informations de proximités (en général, de similarités). Enfin, de récents paradigmes dont il sera question par la suite proposent de ne pas nécessairement choisir un algorithme, mais d'appliquer plusieurs algorithmes différents. Le choix en est alors laissé à l'utilisateur entre :
- avoir plusieurs résultats de *clusterings* différents pour un même ensemble d'individus mais tous de bonne qualité au sens d'une évaluation particulière ;
- choisir le meilleur *clustering* parmi les différents résultats ;
- construire un *clustering* qui réalise un accord entre les divers résultats possibles.

Ces différents choix sont autant de problématiques auxquelles les contributions proposées dans la suite visent à apporter des éléments de réponse. Ces apports constituent chacun un chapitre de ce travail de thèse.

Classification non supervisée multi-vues centralisée

2

Sommaire

2.1 Introduction **50**

2.2 Contexte **50**

2.3 Approches centralisées **52**

2.3.1 MVDBSCAN : DBSCAN multi-vues 53

2.3.2 COFC : *clustering* flou collaboratif 54

2.3.3 FCPU : *clustering* flou dans les univers parallèles 56

2.3.4 MVADASOM : SOM multi-vues *via* les distances adaptatives 58

2.3.5 COMRAF*: champs aléatoires combinatoires de markov 61

2.3.6 COEM : estimation d'un modèle de mélange pour données multi-vues . 63

2.4 Contributions **66**

2.4.1 Motivation 66

2.4.2 COFKM : *clustering* flou multi-vues 66

2.4.3 COKFKM : *clustering* flou multi-vues à noyaux 73

2.5 Évaluation **77**

2.5.1 Données 78

2.5.2 Protocole expérimental 78

2.5.3 Évaluation interne 79

2.5.4 Évaluation externe 80

2.6 Discussion **87**

2.7 Conclusion **87**

2.1 Introduction

Dans ce chapitre présentant la problématique du *clustering* multi-vues, les contributions CoFKM et CoKFKM sont développées. Elles ont été validées par différentes communautés scientifiques établissant des avancées dans le domaine de la *Fouille de données* et de l'*Apprentissage* [Sublemontier et al., 2009], [Cleuziou et al., 2009], [Sublemontier et al., 2011a]. Le contexte scientifique amenant les propositions sera établi. L'étude d'une famille d'algorithmes de *clustering* multi-vues rencontrées dans l'état de l'art permettra de compléter d'un point de vue technique l'appréhension du problème et sa résolution. Les différentes techniques, dites centralisées sont pour la grande majorité basées sur un principe de minimisation d'un désaccord ou, de manière équivalente, de maximisation d'un accord. À l'instar des algorithmes présentés dans le chapitre 2, elles seront détaillées selon leur nature discriminative, générative, ou purement algorithmique. Par suite, les contributions proposées seront introduites, formalisées et recentrées au cœur des études de l'état de l'art. Les études empiriques réalisées permettent de valider l'intérêt pratique des différentes contributions, et la discussion permettra de présenter les avantages et inconvénients de celles-ci. Pour finir, la conclusion dressera les perspectives d'amélioration du modèle.

L'objectif des approches de *clustering* multi-vues basées sur la réduction de désaccord est de produire une structure permettant d'organiser les données décrites par plusieurs représentations. Celle-ci correspond majoritairement à une partition de taille fixée issue d'une recherche de consensus entre plusieurs algorithmes appliqués sur les différentes vues des données. La notation suivante permet d'harmoniser les formalisations des différentes approches et participe à une meilleure compréhension des apports :

NOTATION	
n :	le nombre d'individus à regrouper.
$n_p^{(r)}$:	le nombre d'attributs décrivant les individus dans la vue r.
n_k :	le nombre de groupes à identifier.
n_c :	le nombre de classes associé aux données.
$\mathcal{X} = \{x_1, ..., x_n\}$:	l'ensemble des n individus à partitionner.
$X^{(r)} \in \mathbb{R}^{n \times n_p^{(r)}}$:	la représentation matricielle de $\mathcal{X}$ dans la vue r.
$x_i^{(r)} \in \mathbb{R}^{n_p^{(r)}}$:	la représentation vectorielle de l'individu x_i dans la vue r.
$C = \{C_1, ..., C_{n_k}\}$:	la structure de *clustering* en n_k groupes à construire.
$\Pi = \{C^{(1)}, ..., C^{(n_r)}\}$:	l'ensemble des n_r *clusterings* locaux dans chaque vue.
$C^{(r)} = \{C_1^{(r)}, ..., C_{n_k}^{(r)}\}$:	l'ensemble des n_k groupes du *clustering* dans la vue r.
$\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{n_c}\}$:	l'ensemble des n_c classes d'individus à retrouver.
$d_{(r)}(x_i, x_j)$:	la distance au sens général entre deux individus x_i et x_j dans la vue r.
$\|\|x_i^{(r)} - x_j^{(r)}\|\|_p$:	la distance de Minkowski entre deux individus x_i et x_j dans la vue r.

2.2 Contexte

Le *clustering* multi-vues et l'hypothèse du consensus. La problématique du *clustering* multi-vues peut être définie ainsi : À partir d'un ensemble de tableaux relationnels et/ou descriptionnels (les vues), trouver une partition stricte de l'ensemble d'individus en tenant compte simultanément de l'ensemble des tableaux. Les différentes vues des données induisent naturellement des *clusterings* propres de bonne qualité et différents. L'hypothèse du consensus traduit le fait qu'une solution de *clustering* différente, obtenue par la prise en compte simultanée de

l'ensemble des vues, doit être de *meilleure* qualité. En particulier, cette solution satisfait un accord, ou un consensus entre les *clusterings* locaux potentiels. Cette problématique s'inscrit dans un cadre large de données :

- réparties sur plusieurs sites ;
- pour lesquelles les descriptions sont accessibles par l'intermédiaire de sources multiples ;
- décrites par des groupes de variables de types différents ;
- décrites dans le temps ou plus généralement dans des conditions différentes.

Les applications. Parmi les nombreux domaines d'applications présentés par exemple dans l'introduction, les approches proposées ont été appliquées à la reconnaissance de chiffres manuscrits et à la classification automatique de pages web.

Dans le premier type d'application, le problème est que les individus, qui sont des instances d'images de caractères manuscrits peuvent être numériquement décrits selon différentes mesures propres à l'analyse et au traitement du signal (coefficients de Fourier, coefficients de Karhunen-Loève, intensité des pixels ou autre descripteurs morphologiques). Chacune de ces mesures capture différents aspects de la forme des chiffres. L'établissement d'une mesure de proximité fondée sur chacune de ces descriptions est un problème car elles sont souvent sensibles à des transformations mineures des individus. L'intensité des pixels est sensible à la translation et les descriptions morphologiques sont insensibles à la rotation rendant par exemple difficile la différenciation du chiffre « 6 » et du chiffre « 9 ». L'utilisation conjointe de différentes représentations des individus peut aider à retrouver les bonnes classes.

Dans le second cas, la tâche est d'effectuer un regroupement de différentes pages où chaque page est tirée d'une université parmi quatre universités américaines. Chaque page correspond soit à un étudiant, un département, une faculté, un projet, un membre salarié ou un cours. De ces pages sont considérées le contenu textuel, pour lequel des mesures de similarité adaptées peuvent être construites afin de retrouver les classes d'origine. Cette représentation est enrichie d'un autre vocabulaire émanant cette fois du texte écrit dans les liens entrant vers chacune des pages. Cet aspect supplémentaire des pages peut aider le *clustering* en permettant d'identifier plus facilement les classes.

Les différents principes d'intégration. En général, même si la mise à disposition d'informations supplémentaires complexifie en général les approches, elles peuvent être vues au contraire comme un moyen supplémentaire de réussir à identifier les bonnes classes. Cela devient donc un atout de pouvoir disposer de plusieurs sources d'information notamment lorsque prises isolément celles-ci ne sont pas suffisantes pour obtenir un *clustering* cible souhaité.

Dans ce contexte, il convient alors de combiner les informations de chacune des vues par l'intermédiaire d'un processus de fusion consistant à identifier l'accord entre les vues et à réduire le conflit. Plusieurs stratégies de fusion peuvent être appliquées, en amont, en aval, ou pendant le processus de classification. La fusion en amont ou *a priori* consiste à combiner les différentes représentations des individus, soit en concaténant les descripteurs lorsque les données sont de type vectoriel ou attribut-valeur, soit en effectuant une combinaison (le plus souvent linéaire) des différentes valeurs de proximité lorsque les données sont relationnelles [Heer and Chi, 2002], [Yamanishi et al., 2004].

La fusion en aval ou *a posteriori* [Reza et al., 2009] vise plutôt à construire localement un *clustering* adapté dans chaque représentation puis à appliquer un processus de conciliation entre les différentes partitions pour parvenir à un *clustering* consensus. Ce problème est étudié plus en détail dans le chapitre 4. Les différentes approches sont schématisées dans la figure 2.1.

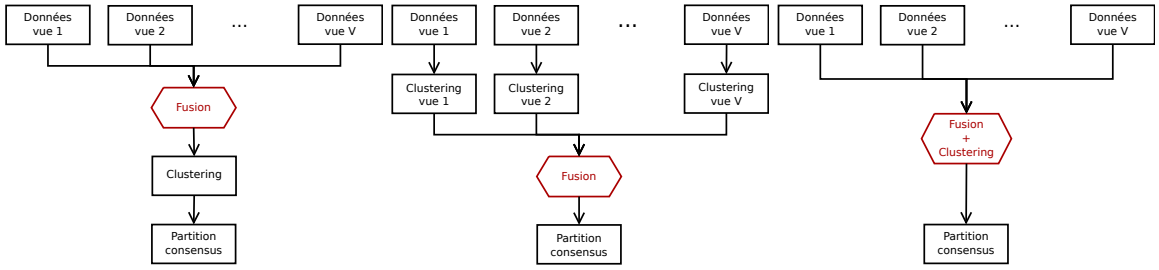

FIGURE 2.1 — Les différentes fusions du *clustering* multi-vues. Dans l'ordre, ci-dessus, les fusions *a priori*, *a posteriori* et dans le processus de *clustering*.

Ce chapitre concerne les approches réalisant un consensus pendant le processus de *clustering*. Toutes fonctionnent sur le principe d'une minimisation d'un terme de désaccord, ou de manière duale la maximisation d'une fonction d'accord entre les *clusterings* naturels en construction localement dans chaque vue. Cette optimisation simultanée peut être explicite *via* la définition d'une fonction réalisant cet objectif, ou bien implicite *via* un algorithme construisant une solution satisfaisant effectivement un tel accord. L'étude est centrée autour des approches dites *centralisées* et s'inscrivant parmi les familles plutôt discriminatives et génératives. Les approches centralisées visent à réunir dans un traitement unique, des données qui peuvent être elles décentralisées. Historiquement, les approches développées avant la proposition des contributions étaient soit :

locales et restreintes du point de vue de la définition du critère objectif pour garantir de bonnes propriétés de convergence, résultant alors en une construction des groupes peu intuitive [Pedrycz, 2002] ;

globales et plus abouties du point de vue de la formulation du problème, mais pour lequel le problème de convergence vers une solution unique est résolu de manière artificielle et moins élégante [Bickel and Scheffer, 2005].

Parmi les contributions proposées, CoFKM vise à répondre à ces différents problèmes à travers la définition d'un critère objectif simple, flexible, et permettant d'en dériver un algorithme intuitif et facilement implémentable. CoKFKM est une proposition permettant d'étendre CoFKM à des données relationnelles qui peuvent se retrouver couramment parmi les applications.

2.3 Approches centralisées

À l'instar des méthodes de *clustering* classiques, les approches multi-vues centralisées ont été développées en suivant différents paradigmes de modélisation. On dénombre ainsi :

- les approches purement algorithmiques ;
- les approches discriminatives ou basées sur un modèle statistique graphique procédant à l'optimisation d'un critère objectif.

Cependant, à des fins d'observation fine du phénomène de réduction du désaccord entre les *clusterings* locaux de chaque vue, le second paradigme sort victorieux notamment par la possibilité d'exprimer la recherche d'une bonne solution comme optimale d'un certain critère objectif intégrant une mesure de ce désaccord. Les critères ainsi proposés prennent le plus souvent la forme d'une combinaison d'un terme classique traduisant la recherche d'un *clustering*

dans chaque vue, pénalisé par un terme exprimant la recherche de l'accord entre ces différents *clusterings*. Ainsi l'objectif est de trouver un compromis entre la découverte de *clusterings* locaux et la recherche du consensus, selon le formalisme général suivant :

$$\text{clustering multi-vues} = \sum_{r=1}^{n_r} \text{objectif local}(r) - \text{désaccord}(\Pi) \tag{2.1}$$

Ainsi, les différentes approches qui peuvent se ramener à un formalisme de ce type seront présentées comme des instances de celui-ci dans la suite de ce chapitre.

2.3.1 MVDBSCAN : DBSCAN multi-vues

Une des premières approches classiques étendues au cadre du traitement de données multi-représentées est DBSCAN (cf. section 1.3.2.1), au travers l'approche de [Kailing et al., 2004], nommée MVDBSCAN. L'idée est de définir un mécanisme de combinaison des différentes représentations dans le but de rendre applicable l'algorithme DBSCAN. Cette applicabilité nécessite de redéfinir les propriétés *cœur* et *frontière* des individus, centraux dans la définition des groupes.

Algorithme

Pour rappel, DBSCAN nécessite deux paramètres : ϵ et $MinPts$. Si $MinPts$ est un paramètre pouvant être défini identiquement dans toutes les représentations, ϵ lui ne peut rendre compte des topologies propres à chaque représentation en étant défini de manière globale. Les auteurs proposent alors de le définir localement pour chaque vue : $\epsilon^{(r)}$. Ainsi, à partir de ces paramètres, on peut définir localement un voisinage pour chaque individu $\mathcal{N}_{\epsilon^{(r)}}(x_i)$ de la manière suivante :

$$\mathcal{N}_{\epsilon^{(r)}}(x_i) = \{x_j \in \mathcal{X} \mid d_{(r)}(x_i, c_k) \leq \epsilon^{(r)}\}$$

Par cette formalisation locale de voisinage, les auteurs proposent alors deux types de voisinage globaux, permettant de décider, dans un contexte plus proche de l'application de DBSCAN, de la propriété pour un individu d'être *cœur*. Les auteurs proposent différents types de voisinage selon la nature des données multi-vues. Ainsi, un voisinage de type *union* $\mathcal{N}_{\cup}(x_i)$ est exprimé par :

$$\mathcal{N}_{\cup}(x_i) = \bigcup_{r \in [1..n_r]} \mathcal{N}_{\epsilon^{(r)}}(x_i) \tag{2.2}$$

De la même manière, un voisinage de type *intersection* $\mathcal{N}_{\cap}(x_i)$ est défini par :

$$\mathcal{N}_{\cap}(x_i) = \bigcap_{r \in [1..n_r]} \mathcal{N}_{\epsilon^{(r)}}(x_i) \tag{2.3}$$

$x_i \in \mathcal{X}$ est alors un individu *cœur* de type union (resp. intersection) si $|\mathcal{N}_{\cup}(x_i)| \geq MinPts$ (resp. $|\mathcal{N}_{\cap}(x_i)| \geq MinPts$). Les auteurs suggèrent de combiner par une union les représentations dans lesquelles les données sont éparpillées, lorsqu'il est difficile de distinguer le bruit (correspondant à des individus mal mesurés) d'une structure de groupes. Enfin, les représentations denses, portant davantage d'informations, sont combinées par une intersection. L'algorithme DBSCAN peut alors être employé, au choix à partir de la définition du type de voisinage (cf. algorithme 9). Les définitions d'atteignabilité sont directement transposées des définitions de DBSCAN et adaptées selon le type de voisinage. On notera alors indépendemment du type de voisinage choisi :

$$A(x_i) = \{x_j \in \mathcal{X} \mid x_j \textit{ est atteignable en densité par } x_i\}$$

Pour rappel, les individus considérés comme du bruit (mal définis ou *outliers*) sont désignés par $\mathcal{R}$.

Algorithme 9 MVDBSCAN

ENTRÉES : $\mathcal{X}$, $MinPts$, $\{\epsilon^{(r)}\}_{r\in[1..n_r]}$, $type$
SORTIES : $C = \{C_1, ..., C_{n_k}\}$, $\mathcal{R}$
1 : Si $type = union$ alors construire $\mathcal{N}(x_i) = \mathcal{N}_{\cup}(x_i)\ \forall x_i \in \mathcal{X}$
2 : Si $type = intersection$ alors construire $\mathcal{N}(x_i) = \mathcal{N}_{\cap}(x_i)\ \forall x_i \in \mathcal{X}$
3 : $C =$ clustering de $\mathcal{X}$ par DBSCAN (cf. algorithme 5) selon $\mathcal{N}$.

Discussion

Cette approche souffre de plusieurs faiblesses, comme l'imposition *a priori* du type de combinaison pour toutes les représentations, et la multiplicité des paramètres. Les auteurs proposent à l'image de DBSCAN un moyen heuristique pour déterminer les valeurs locales de ϵ en fixant l'autre paramètre $MinPts$. En ce qui concerne la combinaison, l'approche a été étendue ultérieurement pour pouvoir considérer simultanément une partie des représentations par union et l'autre partie par intersection, après avoir décidé au travers de critères objectifs de la prévalence de chacune des représentations à un type de combinaison particulier. La combinaison des différentes représentations est représentée au moyen d'une structure d'arbre appelée arbre de combinaison [Achtert et al., 2006].

2.3.2 COFC : *clustering* flou collaboratif

Le *clustering* flou collaboratif, développé par [Pedrycz, 2002] reprend l'approche FKM (cf. section 1.4.1) et en dérive une variante collaborative, notée COFC, pour le contexte multi-vues, il s'agit donc d'une approche discriminative. La collaboration entre les vues est réalisée au travers de l'échange des degrés d'appartenances des individus aux groupes.

Objectif

[Pedrycz, 2002] propose de présenter l'objectif comme la minimisation pour une vue r donnée, d'un critère basé sur FKM, pénalisé par une fonction de désaccord modélisant un écart entre la partition floue locale à construire et les partitions floues provenant des autres vues (2.4). L'auteur propose de renforcer ou diminuer l'impact de la pénalisation en introduisant une matrice de collaboration α telle qu'une grande valeur de $\alpha_{rr'}$ force une plus grande collaboration entre les vues r et r'.

Le critère Q_{CoFC} s'inscrit dans le paradigme des critères pénalisés, ainsi :

$$Q_{\text{CoFC}}(c, u, r) = \text{objectif local}(r) + \text{désaccord}(\Pi)$$

avec

$$\begin{aligned} \text{objectif local}(r) &= \sum_{x_i\in\mathcal{X}} \sum_{k=1}^{n_k} {u_{ik}^{(r)}}^2 d^2_{(r)}(x_i, c_k) \\ \text{désaccord}(\Pi) &= \Delta(\Pi, r) \end{aligned}$$

Dans ce contexte le premier terme du critère, à minimiser, correspond à l'objectif local qui est l'inertie floue semblable à Q_{FKM} à paramètre β fixé ($\beta = 2$). Le second terme, à minimiser également, modélise le désaccord entre les *clusterings* locaux $C^{(r)}$ représentés par leurs centres $c^{(r)}$ et leurs degrés d'appartenance $u^{(r)}$. Pour r donné, ce désaccord est fonction des centres $c^{(r)}$

et mesure l'écart entre les degrés d'appartenance locaux $u^{(r)}$ et les degrés $u^{(\overline{r})}$ des autres vues, renforcé par les variables de collaboration $\alpha^{(r)(\overline{r})}$. Ainsi Δ est défini par :

$$\Delta(c,u,r) = \sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r} \alpha^{(r)(\overline{r})} \sum_{x_i\in\mathcal{X}} \sum_{k=1}^{n_k} (u_{ik}^{(r)} - u_{ik}^{(\overline{r})})^2 d_{(r)}^2(x_i,c_k)$$

Le problème d'optimisation associé est alors exprimé par :

$$\begin{aligned} &\min_{c,u} Q_{\text{CoFC}}(c,u,r) \\ &= \min_{c,u} \sum_{k=1}^{n_k} \sum_{x_i\in\mathcal{X}} {u_{ik}^{(r)}}^2 d_{(r)}^2(x_i,c_k) + \sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r} \alpha^{(r)(\overline{r})} \sum_{x_i\in\mathcal{X}} \sum_{k=1}^{n_k} (u_{ik}^{(r)} - u_{ik}^{(\overline{r})})^2 d_{(r)}^2(x_i,c_k) \\ &s.t.\ \sum_{k=1}^{n_k} u_{ik}^{(r)} = 1 \quad \forall x_i \in \mathcal{X} \\ &\qquad\quad u_{ik}^{(r)} \geq 0 \quad \forall x_i \in \mathcal{X},\ \forall k \in [1..n_k] \end{aligned} \tag{2.4}$$

Dans la version classique FKM, le critère d'inertie flou est modulé par un paramètre $\beta > 1$ qui est ici fixé à 2 dans l'objectif de CoFC, pour des raisons d'optimisation efficace du critère et par extension, de convergence de l'algorithme d'optimisation associé au problème. De ce point de vue, CoFC ne généralise pas pleinement FKM. Ce problème est résolu par l'optimisation alternée des différentes variables c et u.

Algorithme

À l'image de FKM, dès lors que le critère est posé, l'algorithme se déduit naturellement. En effet le but étant de minimiser le critère objectif, l'optimal est atteint lorsque les conditions du premier ordre sont satisfaites. Ainsi, ces conditions permettent d'établir des expressions de mise à jour optimales des degrés d'appartenance, connaissant les prototypes des groupes :

$$u_{ik}^{(r)*} = \frac{\sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r} \alpha^{(r)(\overline{r})} u_{ik}^{(\overline{r})}}{1 + \sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r} \alpha^{(r)(\overline{r})}} + \frac{1}{\sum_{k'=1}^{n_k} \frac{d_{(r)}^2(x_i,c_k)}{d_{(r)}^2(x_i,c_{k'})}} \left(1 - \sum_{k'=1}^{n_k} \frac{\sum_{\overline{r}=1}^{n_r} \alpha^{(r)(\overline{r})} u_{ik'}^{(\overline{r})}}{\sum_{r'=1}^{n_r} \alpha^{(r)(\overline{r})}}\right) \tag{2.5}$$

De la même manière, si on a à disposition les degrés d'appartenance considérés comme optimaux, alors nous pouvons mettre à jour de manière optimale les prototypes des groupes par :

$$c_k^{(r)*} = \frac{\sum_{x_i\in\mathcal{X}} {u_{ik}^{(r)}}^2 x_i^{(r)} + \sum_{\overline{r}=1}^{n_r} \alpha^{(r)(\overline{r})} \sum_{x_i\in\mathcal{X}} (u_{ik}^{(r)} - u_{ik}^{(r')})^2 x_i}{\sum_{x_i\in\mathcal{X}} {u_{ik}^{(r)}}^2 + \sum_{\overline{r}=1}^{n_r} \alpha^{(r)(\overline{r})} \sum_{x_i\in\mathcal{X}} (u_{ik}^{(r)} - u_{ik}^{(r')})^2} \tag{2.6}$$

Discussion

Algorithme 10 CoFC

ENTRÉES : $\mathcal{X}$, n_k, α
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Appliquer FKM sur $\mathcal{X}$, $\forall r \in [1..n_r]$
2 : Mise à jour des $u_{ik}^{(r)}$, $\forall x_i \in \mathcal{X}$, $\forall k \in [1..n_k]$, $\forall r \in [1..n_r]$ en utilisant (2.5)
3 : Mise à jour des $c_k^{(r)}$, $\forall k \in [1..n_k]$, $\forall r \in [1..n_r]$ en utilisant (2.6)
4 : Si Q_{CoFC} change alors aller en **3**
5 : $C_k = \{x_i \in \mathcal{X} | u_{ik}^{(r)} = \max_{k' \in [1..n_k]} u_{ik'}^{(r)}\}$, $\forall k \in [1..n_k]$

L'approche CoFC permet d'obtenir, pour une vue r contenant des informations sur un ensemble d'individus $\mathcal{X}$, un *clustering* flou de $\mathcal{X}$ en exploitant des informations émanant d'autres vues. Ces informations prennent la forme, pour chaque individu, d'un profil d'appartenance à l'ensemble des groupes, tel que les nombres de groupes dans toutes les vues soient identiques : $\forall k \in [1..n_k]$, $\forall r \in [1..n_r]$ $n_k^{(r)} = n_k$. Ce choix d'intégration a l'avantage de préserver la confidentialité des données. En particulier, dans une vue r, il n'est pas possible d'accéder aux propriétés présentes dans les autres vues. Ainsi, seuls les degrés d'appartenance sont échangés entre les vues. Cela réduit le coût opérationnel de transfert d'informations par le réseau entre les différentes parties des données présentes sur ces différents sites.

Néanmoins, l'approche, visant à étendre FKM, ne peut le faire complètement (choix de β). De plus, malgré l'aspect intuitif et facilement interprétable du critère objectif à optimiser, celui-ci induit des formules de mises à jour des variables du problème d'optimisation, elles, très peu intuitives. Enfin, lorsque l'on cherche un *clustering* collaboratif à partir d'une vue r, il n'est pas précisé si les informations provenant des autres vues sont immuables ou si elles évoluent également en parallèle. Quoiqu'il en soit, il n'y a pas de processus de construction des groupes réellement global, où les groupes dans chaque vue sont construits simultanément pour tendre vers une solution consensus bien définie comme l'optimale d'une fonction globale sur les vues.

2.3.3 FCPU : *clustering* flou dans les univers parallèles

Dans le même esprit que l'approche CoFC, d'autres propositions ont pour objectif d'étendre FKM au cadre des représentations multiples. L'approche de *clustering* flou dans les univers parallèles [Wiswedel and Berthold, 2007], notée FCPU a pour objectif de trouver une organisation globale en exploitant simultanément l'ensemble des vues disponibles, appelées univers parallèles. L'idée principale que l'on considère ici est que les individus ne contribuent pas de manière équivalente à la définition des groupes dans les différentes représentations. Les auteurs proposent alors d'introduire une variable modélisant pour chaque individu sa contribution à la définition des groupes dans chaque vue. Cela permet d'observer leur apport aux processus de *clusterings* locaux, qui sont réalisés simultanément.

Objectif

Les auteurs formalisent la recherche de l'ensemble des degrés d'appartenance flous (dans toutes les vues) comme l'optimum d'un critère (Q_{FCPU}) basé sur une combinaison linéaire des inerties floues (type FKM) locales, pondérées par les contributions des individus aux représentations :

$$Q_{\text{FCPU}}(c, u, v) = \sum_{r=1}^{n_r} \text{objectif local}(r)$$

avec

$$\text{objectif local}(r) \quad = \quad \sum_{r=1}^{n_r} \sum_{x_i \in \mathcal{X}} {v_i^{(r)}}^{\gamma} \sum_{k=1}^{n_k} {u_{ik}^{(r)}}^{\beta} d^2_{(r)}(x_i, c_k)$$

Par rapport à la forme globale des critères objectifs des approches centralisées, on peut noter que la recherche d'un accord ne fait pas parti de l'objectif global, dans la mesure où les auteurs se placent dans le cadre où tous les groupes ne sont pas significativement identifiables dans chaque représentation. Le problème d'optimisation correspondant est alors :

$$\begin{array}{lrcll} \underset{c,u,v}{min}\, Q_{\text{FCPU}}(r) = & \displaystyle\sum_{r=1}^{n_r} \sum_{x_i \in \mathcal{X}} {v_i^{(r)}}^{\gamma} \sum_{k=1}^{n_k} {u_{ik}^{(r)}}^{\beta} d^2_{(r)}(x_i, c_k) & & & \\ s.t. & \sum_{k=1}^{n_k} u_{ik}^{(r)} & = & 1 & \forall x_i \in \mathcal{X},\ \forall r \in [1..n_r] \\ & \sum_{r=1}^{n_r} v_i^{(r)} & = & 1 & \forall x_i \in \mathcal{X} \\ & u_{ik}^{(r)} & \geq & 0 & \forall x_i \in \mathcal{X},\ \forall r \in [1..n_r],\ \forall k \in [1..n_k] \\ & v_i^{(r)} & \geq & 0 & \forall x_i \in \mathcal{X},\ \forall r \in [1..n_r] \end{array} \tag{2.7}$$

La solution localement optimale est encore une fois déterminée par optimisation alternée sur les différentes variables et son obtention est complètement dérivée du critère.

Algorithme

De manière similaire à CoFC, le critère objectif, intuitif, permet de dériver un algorithme simple pour chercher un optimum local. Partant d'un ensemble de valeurs initiales des variables du problème d'optimisation (prototypes, degrés d'appartenance et contributions), chacune des variables peut être ré-estimée de manière optimale par une formule issue de la résolution du système émanant de la satisfaction des conditions du premier ordre. Ainsi, pour des valeurs de prototypes et de contributions fixées, les nouveaux degrés d'appartenance sont mis à jour par :

$$u_{ik}^{(r)*} = \frac{\left(d^2_{(r)}(x_i, c_k)\right)^{1/(1-\beta)}}{\displaystyle\sum_{k'=1}^{n_k} \left(d^2_{(r)}(x_i, c_{k'})\right)^{1/(1-\beta)}} \tag{2.8}$$

ce qui correspond exactement à la mise à jour des degrés d'appartenance de FKM dans la vue r.

De la même manière, en fixant les degrés d'appartenance et les contributions, et en établissant la nature de la distance $d^{(r)}$, les nouveaux prototypes sont appris par :

$$c_k^{(r)*} \quad = \quad \frac{\displaystyle\sum_{x_i \in \mathcal{X}} {v_i^{(r)}}^{\gamma} {u_{ik}^{(r)}}^{\beta} x_i}{\displaystyle\sum_{x_i \in \mathcal{X}} {v_i^{(r)}}^{\gamma} {u_{ik}^{(r)}}^{\beta}} \tag{2.9}$$

pour une distance euclidienne $d_{(r)}(x_i, c_k) = ||x_i - c_k||_2$. Chaque centre $c_k^{(r)}$ devient alors le barycentre des individus, pondérés par leur degré d'appartenance au groupe C_k, et pondérés également par leur contribution au *clustering* dans la vue r.

Enfin, pour les degrés d'appartenances et prototypes courants connus, les contributions sont réévaluées par :

$$v_i^{(r)*} = \frac{\left(\sum_{k=1}^{n_{kr}} {u_{ik}^{(r)}}^{\beta} d^2_{(r)}(x_i, c_k)\right)^{1/(1-\gamma)}}{\sum_{r'=1}^{n_r} \left(\sum_{k=1}^{n_{kr'}} {u_{ik}^{(r')}}^{\beta} d^2_{(r')}(x_i, c_k)\right)^{1/(1-\gamma)}} \tag{2.10}$$

Algorithme 11 FCPU

ENTRÉES : $\mathcal{X}$, $\{n_k^{(r)}\}_{r\in[1..n_r]}$, β, γ
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation des $n_k^{(r)}$ centres de groupes $\{c_1^{(r)}, ..., c_{n_k}^{(r)}\}$ dans la vue r
2 : Mise à jour des $u_{ik}^{(r)}$, $\forall x_i \in \mathcal{X}$, $\forall k \in [1..n_k]$, $\forall r \in [1..n_r]$ en utilisant (2.8)
3 : Mise à jour des $c_k^{(r)}$, $\forall k \in [1..n_k]$, $\forall r \in [1..n_r]$ en utilisant (2.9)
4 : Mise à jour des $v_i^{(r)}$, $\forall x_i \in \mathcal{X}$, $\forall r \in [1..n_r]$ en utilisant (2.10)
5 : Si Q_{FCPU} change alors aller en **2**
6 : $C_k = \{x_i \in \mathcal{X} | u_{ik}^{(r)} = \max_{k'\in[1..n_k]} u_{ik'}^{(r)}\}$, $\forall k \in [1..n_k]$

Discussion

FCPU se place dans un cadre général où l'on suppose que les diverses vues des individus sont insuffisantes isolément pour identifier l'ensemble des classes. Ainsi, tous les individus ne sont pas utiles localement pour représenter les groupes. Enfin le critère objectif est intuitif, et contrairement à CoFC, les mises à jour des paramètres le sont aussi.

Néanmoins, même si l'introduction de la variable permettant de capturer la contribution naturelle des individus à la définition des groupes est une idée à retenir, plusieurs problèmes se posent. En effet, un même individu pourrait avoir une forte contribution au *clustering* dans toutes les représentations, ou bien être un individu atypique *i.e.* ne devant naturellement contribuer à la définition d'aucun groupe. Dans les deux cas, la contrainte de sommation à 1 des contributions conduirait à une distribution uniforme des valeurs de ces contributions. Ceci est gênant du point de vue de l'interprétabilité de l'apport de chaque individu pour chaque vue, ce qui est un objectif souhaité de l'approche.

2.3.4 MVADASOM : SOM multi-vues *via* les distances adaptatives

Toujours parmi les extensions d'algorithmes classiques, [dos S. Dantas and de Carvalho, 2011] ont développé l'approche *batch*-SOM (cf. section 1.3.2.2) adaptative dédiée au traitement de plusieurs matrices de dissimilarités, notée MVADASOM. L'objectif est de trouver une carte auto-organisatrice unique permettant d'obtenir un *clustering* des individus multi-représentés en exploitant simultanément les différentes vues.

Objectif

Les auteurs proposent de modifier dans le critère initial Q_{SOM} la mesure de dissimilarité utilisée, en la remplaçant par une moyenne pondérée des dissimilarités disponibles pour chaque

représentation, notée D_{w_k}, définie formellement par :

$$D_{w_k}(x_i, c_k) = \sum_{r=1}^{n_r} w_k^{(r)} d_{(r)}(x_i, c_k) \tag{2.11}$$

Selon le formalisme des approches centralisées, le critère Q_{MVADASOM} s'exprime comme une somme d'objectifs locaux, le consensus étant imposé par la dissimilarité globale aux centres :

$$Q_{\text{MVADASOM}} = \sum_{r=1}^{n_r} \text{objectif local}(r)$$

avec :

$$\text{objectif local}(r) = \sum_{x_i \in \mathcal{X}} \sum_{k=1}^{n_k} K(c_k, f^*(x_i)) w_k^{(r)} d_{(r)}(x_i, c_k)$$

où $c_k \in \mathcal{X}$ est le k-ième neurone et le même pour toutes les vues et les poids $w_k^{(r)}$ permettent de donner une importance relative aux neurones selon les représentations.

Ainsi le problème d'optimisation se formalise comme la recherche du minimum du critère Q_{MVADASOM} :

$$\underset{c,w}{min}\, Q_{\text{MVADASOM}}(c, w) = \underset{c,w}{min} \sum_{k=1}^{n_k} \sum_{x_i \in \mathcal{X}} K(c_k, f^*(x_i)) D_{w_k}(x_i, c_k)$$

et la solution optimale s'obtient par un algorithme similaire à celui des SOM.

Algorithme

L'idée est toujours de trouver les n_k neurones ou prototypes optimaux, identiques pour toutes les représentations puisque ceux-ci sont évalués selon la mesure de dissimilarité globale (2.11). De plus ces prototypes sont choisis non pas dans l'espace dans lequel sont distribués les individus de $\mathcal{X}$, mais parmi $\mathcal{X}$ lui-même, notamment car une description explicite de $\mathcal{X}$ dans un espace vectoriel n'est pas fourni. Ainsi les prototypes correspondent à des individus bien précis de l'échantillon. L'inertie est pondérée par une fonction K quantifiant toujours, pour un terme de l'inertie donné (en fixant k et i), une similarité entre le neurone concerné c_k et le neurone le plus représentatif de l'individu concerné $f^*(x_i)$. Ce dernier est obtenu par:

$$f^*(x_i) = \underset{c \in \{c_1, \ldots, c_{n_k}\}}{arg\ min} \sum_{k=1}^{n_k} \sum_{x_i \in \mathcal{X}} K(c_k, f^*(x_i)) D_{w_k}(x_i, c_k) \tag{2.12}$$

Les auteurs proposent d'évaluer la similarité entre deux neurones c_i et c_j par:

$$K(c_i, c_j) = \frac{e^{-||c_i - c_j||_1^2}}{\lambda(t)^2}$$

La similarité $K(c_k, f^*(x_i))$ est maximale lorsque $f^*(x_i) = c_k$, ainsi $K(c_k, f^*(x_i)) = 1$. La variable $\lambda(t)$ traduisant une température, est fonction du nombre d'itérations souhaité t_{max} et de l'itération courante t. Elle permet de faire évoluer les valeurs de similarité plus rapidement, pour des raisons de convergence.

$$\lambda(t) = \lambda_f \left(\frac{\lambda_i}{\lambda_f}\right)^{\frac{t}{t_{max}}}$$

Algorithme 12 batch-MVADASOM

ENTRÉES : $\mathcal{X}$, n_k, G, λ_i, λ_f, t_{max}
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : $t = 1$ et initialiser aléatoirement les n_k neurones $\{c_1, ..., c_{n_k}\}$
2 : Initialiser $w_k^{(r)} = 1,\ \forall k \in [1..n_k],\ \forall r \in [1..n_r]$
3 : Mise à jour de $f^*(x_i),\ \forall x_i \in \mathcal{X}$ selon (2.12)
4 : Mise à jour des neurones $c_k,\ \forall k \in [1..n_k]$ selon (2.14)
5 : Mise à jour des $w_k^{(r)},\ \forall k \in [1..n_k],\ \forall r \in [1..n_r]$ selon (2.13)
6 : Si Q_{MVADASOM} change $t = t + 1$ et aller en **3**.
7 : $C_k = \{x_i \in \mathcal{X} | f^*(x_i) = c_k\}\ \forall k \in [1..n_k]$

où λ_i et λ_f sont des bornes définies *a priori* et correspondante respectivement à la température initiale de la carte, et à la température finale permettant d'atteindre la convergence. La détermination des neurones les plus représentatifs $f^*(x_i)$ permet de réévaluer les contributions des groupes aux différentes vues, qui est traduit par la variable $w_k^{(r)}$ calculée de manière optimale par l'équation:

$$w_k^{(r)*} = \frac{\left(\prod_{r=1}^{n_r} \sum_{x_i \in \mathcal{X}} \Big(K(c_k, f^*(x_i)) d_{(r)}(x_i, c_k)\Big)\right)^{\frac{1}{n_r}}}{\sum_{x_i \in \mathcal{X}} \Big(K(c_k, f^*(x_i)) d_{(r)}(x_i, c_k)\Big)} \tag{2.13}$$

Ainsi, plus un neurone c_k est représentatif de l'ensemble des individus $x_i \in \mathcal{X}$ dans une vue relativement aux autres, plus la valeur de contribution augmente, car le terme d'inertie du dénominateur est plus faible, à valeur du numérateur identique pour toutes les représentations. Enfin, les neurones sont mis à jour de manière optimale en calculant l'optimum du critère pour des valeurs de $K(c_i, c_j)$ et $w_k^{(r)}$ fixées:

$$c_k^* = \underset{c \in \mathcal{X}}{arg\ min} \sum_{k=1}^{n_k} \sum_{x_i \in \mathcal{X}} K(c_k, f^*(x_i)) D_{w_k}(x_i, c_k) \tag{2.14}$$

Discussion

L'approche MVADASOM étend ingénieusement les SOM à la problématique des données multi-vues, lorsque les individus sont représentés par des tableaux relationnels de dissimilarité. On remarque que le consensus est imposé par la définition du critère objectif, notamment par la définition de la mesure de dissimilarité globale. Ainsi une carte unique est apprise et il n'est pas possible de contrôler le compromis entre les clusterings locaux naturels et le désaccord entre les différentes représentations. Enfin l'autre remarque que l'on peut soulever est sur l'imposition des paramètres supplémentaires pour garantir la convergence, qui alourdissent le critère. Cependant ils découlent directement du modèle des SOM. Dans le même esprit, d'autres approches récentes ont étendu l'approche SOM au cadre des données multi-vues, en optimisant un critère plus proche dans l'esprit, de l'approche COFC [Grozavu and Bennani, 2010],[Grozavu et al., 2011], [Mesghouni et al., 2011].

2.3.5 CoMRaF*: champs aléatoires combinatoires de markov

Parmi les approches de *clustering* de données multi-vues, on trouve également des approches basées sur des modèles graphiques tels que le modèle CoMRaF* [Bekkerman and Jeon, 2007], qui restreint le modèle plus général CoMRaF [Bekkerman et al., 2006].

Modèle

Dans un tel modèle graphique (représenté sous forme de graphe), chaque nœud correspond soit :
- à l'ensemble des individus $\mathcal{X}$ à partitionner ;
- à l'ensemble des propriétés décrivant $\mathcal{X}$ dans une vue, une représentation.

Chaque nœud est associé à une variable aléatoire combinatoire (*v.a.c.*) définie sur l'ensemble des partitions possibles de l'ensemble correspondant à ce nœud. Chaque arête correspond, quant à elle, à une mesure d'intéraction entre les deux *v.a.cs.* qu'elle relie. Dans le cadre général de CoMRaF, on admet qu'il puisse exister des dépendances entre les *v.a.cs.* associées aux représentations (identifiées par $R^{(r)}$ $\forall r \in [1..n_r]$). L'objectif est alors de trouver la réalisation (ou l'instanciation) de chaque variable aléatoire, qui maximise globalement la valeur de probabilité jointe sur l'ensemble des *v.a.cs.*. Dans le cadre spécifique qui nous concerne ici, seule la réalisation de la *v.a.c.* définie sur l'ensemble des partitions de $\mathcal{X}$ nous intéresse, elle sera notée X. Cela conduit au modèle graphique dans lequel le nœud associé à la *v.a.c.* X est central et où chaque réalisation des *v.a.cs.* $R^{(r)}$ (celles-ci sont seulement observées) apporte une information permettant de trouver la meilleure réalisation de X. On considère alors toutes les interactions entre les *v.a.cs.* $R^{(r)}$ et X ce qui donne un modèle en étoile : CoMRaF*(cf. figure 2.2).

Objectif

L'objectif est comme dans la plupart des modèles statistiques, de maximiser la probabilité jointe des variables du modèle (2.15). Comme les *v.a.cs.* $R^{(r)}$ sont seulement observées, elles sont invariantes et leur réalisation correspond à l'ensemble des singletons $S_p^{(r)}$ $\forall p \in [1..|R_r|]$ d'attributs présents dans la vue r. Par exemple, si la vue r représente les individus selon l'ensemble d'attributs $\{a, b, c\}$, alors la *v.a.c.* $R^{(r)}$ observée a pour réalisation $\{\{a\}, \{b\}, \{c\}\}$, et on a $S_1^{(r)} = \{a\}$, $S_2^{(r)} = \{b\}$ et $S_3^{(r)} = \{c\}$. Ainsi, dans le modèle, seule la réalisation C de la variable X est alors une variable du problème d'optimisation qui s'exprime :

$$\max_{C \in \mathbf{P}} Q_{\text{CoMRaF}} = \max_{C \in \mathbf{P}} \sum_{r=1}^{n_r} f^{(r)}(C, R^{(r)}) \tag{2.15}$$

où $\mathbf{P}$ est l'ensemble des partitions de $\mathcal{X}$ et $f^{(r)}$ est une fonction de potentiel mesurant l'intéraction entre les *clusterings* C réalisations de X, et $R^{(r)}$. Par exemple, les auteurs proposent de prendre comme fonction de potentiel, l'information mutuelle entre les variables aléatoires C_k, correspondant au k-ième groupe du *clustering* C, et $S_p^{(r)}$ définies sur C et $R^{(r)}$ respectivement.

Pour résumer, par abus de langage, si on considère les fonctions de potentiels comme des mesures de similarité entre les *clusterings* associés aux nœuds, alors l'objectif consiste à trouver le *clustering* C de $\mathcal{X}$ qui maximise sa similarité globalement et relativement à toutes les vues. Ainsi l'optimum est caractérisé de manière générale comme le MPE, explication la plus probable de la variable X, correspondant au meilleur *clustering* C de $\mathcal{X}$, ainsi :

$$C^*_{MPE} = \arg\max_{C \in \mathbf{P}} \sum_{r=1}^{n_r} f^{(r)}(C, R^{(r)})$$

Algorithme

Les auteurs ont proposé un algorithme permettant de mettre à jour le *clustering* courant de manière à maximiser le critère objectif. Néanmoins cela ne peut se faire en explorant de manière exhaustive l'espace de solutions correspondant à l'ensemble des partitions possibles de $\mathcal{X}$ pour des raisons évidentes de complexité. Ainsi, les auteurs ont alors proposé d'effectuer une recherche locale permettant à partir d'un *clustering* de trouver le MPE de C. Ils restreignent l'espace de recherche à un voisinage $\mathcal{N}(C)$ correspondant à l'ensemble des *clusterings* obtenables en déplaçant un individu d'un groupe de C vers un autre. La règle permettant d'obtenir un maximum local à partir d'un *clustering* C est la suivante :

$$C^* = \underset{C' \in \mathcal{N}(C)}{arg\ max} \sum_{r=1}^{n_r} f^{(r)}(C', R^{(r)}) \tag{2.16}$$

Le voisinage étant relativement « petit », une recherche exhaustive du *meilleur* voisin d'un *clustering* peut alors être effectuée. L'algorithme 13 est alors complètement dépendant de l'initialisation du premier *clustering* et la *meta*-heuristique de recherche est une simple recherche en escalade dont le but est de systématiquement trouver, pour un voisinage fixé de la solution courante, une solution qui maximise le critère objectif posé. La version de CoMRaF* relatée ici considère un nombre de groupes fixé. En effet, ne pas imposer de contraintes sur le nombre de groupes induit dans le cas général, l'obtention d'une solution dégénérée où l'on obtient comme partition optimale l'ensemble des singletons de $\mathcal{X}$. Néanmoins les auteurs proposent d'adapter l'algorithme afin de produire un *clustering* hiérarchique selon une approche ascendante ou descendante.

Algorithme 13 CoMRaF*

Entrées : $\mathcal{X}$, n_k, R, β, γ
Sorties : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation aléatoire de C un *clustering* de $\mathcal{X}$ en n_k groupes
2 : Mise à jour des groupes C en utilisant (2.16)
3 : Si C change alors aller en **2**

Discussion

CoMRaF* est un modèle reposant sur une représentation graphique, ce qui en fait une approche assez intuitive. Il permet de manipuler un nombre quelconque de représentations pour les individus. Il peut être étendu en une recherche de partition s'accordant au mieux avec les diverses vues des données, sans spécifier au préalable un nombre de groupes souhaité. En revanche, la recherche de la meilleure partition repose sur une procédure de parcours de l'espace de recherche très locale (le voisinage est très restreint) et la *meta*-heuristique de recherche associée ne laisse pas assez de place au mauvais choix de l'initialisation, qui est par ailleur délicate sans l'utilisation d'informations externes. En effet si l'on devait étendre ce modèle, dans un premier temps, on pourrait envisager d'encapsuler la recherche de solution par une approche de type recuit simulé plus robuste dans le cas général. De plus, le modèle, même s'il permet d'utiliser tout type de fonction de potentiel bien choisie, nécessite de pouvoir définir des densités de probabilités adaptées entre ces vues, or ceci n'est pas toujours possible. Il peut arriver que certaines représentations n'aient que des variables (ou propriétés) indépendantes pour toute paire d'individus, auquel cas les lois de probabilités jointes entre chacune de ces variables et les individus n'auraient pas grand sens. Les auteurs proposent de résoudre ces cas par l'utilisation du

modèle plus général CoMRaF en cherchant en plus du *clustering* des individus de $\mathcal{X}$, un *clustering* de ces représentations afin de former des groupes de propriétés adaptés. L'astuce consiste à décomposer le modèle CoMRaF en une séquence de modèles CoMRaF* supposée équivalente (Fig. 2.2).

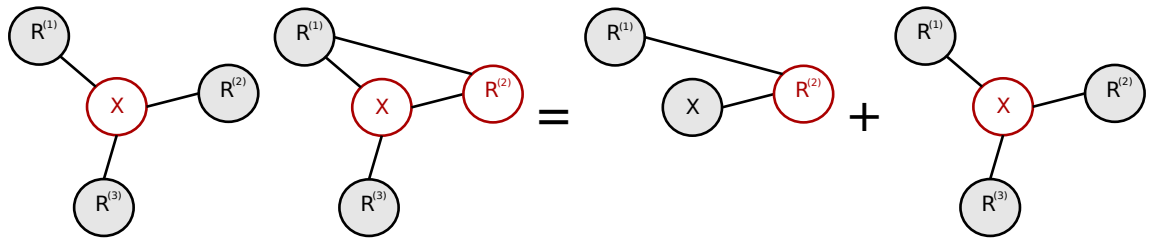

FIGURE 2.2 — Un modèle CoMRaF où les individus de $\mathcal{X}$ sont décrits par 3 représentations. La première figure représente un modèle en étoile CoMRaF*. Dans la suite, les 3 autres figures représente un modèle CoMRaF dans lequel une dépendance est ajoutée entre la *v.a.c.* $R^{(1)}$ et la *v.a.c.* $R^{(2)}$. On cherche la réalisation de X, et de $R^{(2)}$ *v.a.cs.* correspondantes à un *clustering* de $\mathcal{X}$, et un *clustering* de $R^{(2)}$ tels que l'information mutuelle de ceux-ci entre eux et avec chaque autre représentations dont elles sont dépendantes soit maximal. Le premier modèle général CoMRaF (deuxième figure) se décompose en une séquence de deux modèles CoMRaF*.

2.3.6 CoEM : estimation d'un modèle de mélange pour données multi-vues

Toujours parmi les approches statistiques, cette fois génératives, [Bickel and Scheffer, 2005] ont proposé d'étendre le modèle de mélange au cadre de données multi-vues. Ils proposent une variante collaborative, notée CoEM, de l'algorithme EM pour l'estimation des paramètres d'un modèle de mélange de lois expliquant la génération de l'ensemble d'individus multi-représentés.

Modèle

À l'instar d'EM, le modèle considéré est toujours le modèle de mélange, mais cette fois nous supposons l'existence de n_r modèles de mélanges $f^{(r)}$ indépendants et de n_k composantes chacune :

$$f^{(r)}(X_i;\Theta^{(r)}) = \sum_{k=1}^{n_k} \alpha_k^{(r)} f_k^{(r)}(X_i;\theta_k^{(r)}) \tag{2.17}$$

L'objectif est alors d'estimer les paramètres $\Theta = \{\Theta^{(r)}\}_{r\in[1..n_r]}$ expliquant au mieux la génération de l'ensemble d'individus $\mathcal{X}$. Les auteurs proposent d'estimer ces paramètres *via* l'application de l'algorithme EM indépendamment dans chaque représentation en contrôlant la recherche d'une solution unique de *clustering* en s'appuyant sur la recherche de consensus entre les différents modèles locaux.

Objectif

La fonction objectif à maximiser, qui combine linéairement les espérances des log-vraisemblances locales de toutes les vues, est pénalisée par un terme de désaccord $\Delta(\Pi)$ entre les différentes

représentations :

$$Q_{\text{CoEM}}(\Theta;\Theta^{-}) = \sum_{r=1}^{n_r} Q_{\text{EM}}^{(r)}(\Theta^{(r)};\Theta^{-(r)},\mathcal{X},n_k) - \eta\Delta(\Pi)$$

où selon le paradigme des approches multi-vues centralisées (2.1) :

$$\begin{aligned}
\text{objectif local}(r) &= Q_{\text{EM}}^{(r)}(\Theta^{(r)};\Theta^{-(r)},\mathcal{X},n_k) \\
&= \sum_{x_i\in\mathcal{X}}\sum_{k=1}^{n_k} z_{ik}^{(r)} \log(\alpha_k^{(r)} f_k^{(r)}(x_i^{(r)};\theta_k^{(r)})) \\
\text{désaccord}(\Pi) &= \Delta(\Pi)
\end{aligned}$$

La fonction Δ mesure le désaccord entre les *clusterings* en construction dans toutes les vues. Ces clusterings Π sont décrits par les paramètres locaux $\Theta^{(r)} = (\alpha^{(r)},\theta^{(r)})$. Le désaccord est alors formulé par :

$$\Delta(\Theta) = \frac{1}{n_r-1}\sum_{r\neq r'}\sum_{x_i\in\mathcal{X}}\sum_{k=1}^{n_k} f^{(r)}(Z_i=k|X_i=x_i;\Theta^{-(r)}) \log\frac{f^{(r)}(Z_i=k|X_i=x_i;\Theta^{(r)})}{f^{(r')}(Z_i=k|X_i=x_i;\Theta^{(r')})}$$

Le critère peut être simplifié en réinjectant le terme de désaccord dans le premier terme pour faire apparaître une moyenne pondérée sur les différentes représentations de critères de vraisemblance locaux. L'objectif peut alors être formulé comme la maximisation de ce critère :

$$\max_{\Theta} Q_{\text{CoEM}}(\Theta;\Theta^{-}) = \max_{\Theta} \sum_{r=1}^{n_r}\sum_{k=1}^{n_k^{(r)}}\sum_{x_i\in\mathcal{X}} z_{ik\eta}^{(r)} \log(\alpha_k^{(r)} f_k^{(r)}(x_i^{(r)};\theta_k^{(r)}) \tag{2.18}$$

où $z_{ik\eta}^{(r)}$ peut être vue comme une nouvelle estimation des valeurs de probabilités *a posteriori* pour la vue r, et est définie comme une moyenne des valeurs de probabilités *a posteriori* locales :

$$\begin{aligned}
z_{ik\eta}^{(r)} &= f^{(r)}(Z_i=k|X_i=x_i;\Theta^{(r)},\eta) \\
&= (1-\eta)z_{ik}^{(r)} + \frac{\eta}{n_r-1}\sum_{\substack{\bar{r}=1\\ \bar{r}\neq r}}^{n_r} z_{ik}^{(\bar{r})}
\end{aligned} \tag{2.19}$$

Le critère simple vue dans r, $Q_{\text{EM}}^{(r)}$, utilise les individus de l'échantillon $\mathcal{X}$ des données, les variables cachées Z_i et les paramètres des lois $\Theta^{(r)}$ à estimer. Dans l'expression du critère Q_{CoEM}, η ajuste l'importance du désaccord $\Delta(\Pi)$ dans le processus d'optimisation. Ce désaccord est proche d'une divergence de *Kullback-Leibler* (1.27) entre les distributions de probabilités *a posteriori* (courantes et précédentes) sur toutes les paires de vues, ce qui modélise d'une certaine manière un écart entre ces distributions que les auteurs proposent de réduire.

Algorithme

L'algorithme 14 alterne à la manière de EM une étape E de calcul des probabilités *a posteriori* puis une étape M d'estimation des meilleurs paramètres connaissant ces probabilités. La recherche des meilleurs estimateurs des paramètres Θ est réalisée de façon similaire au cadre

EM classique. L'idée est de parcourir les différentes vues et de chercher localement les paramètres optimaux $\Theta^{(r)^*} = \{\theta_k^{(r)}\}_{k\in[1..n_k]}$ relativement aux valeurs de probabilités *a posteriori* globales $z_{ik\eta}^{(r)}$:

$$\theta^* = arg\,\underset{\theta}{max} \sum_{x_i\in\mathcal{X}} \sum_{k=1}^{n_k} z_{ik\eta}^{(r)} \log(\alpha_k^{(r)} f_k^{(r)}(x_i^{(r)};\theta_k^{(r)})) \tag{2.20}$$

Les valeurs de probabilités *a posteriori* sont ré-estimées, non de manière optimale, mais reposent sur les estimateurs locaux obtenus par la règle classique de EM :

$$z_{ik}^{(r)} = \frac{\alpha_k^{(r)} f_k^{(r)}(x_i^{(r)};\theta_k^{(r)})}{\sum_{l=1}^{n_k} \alpha_l^{(r)} f_k^{(r)}(x_i^{(r)};\theta_l^{(r)})} \tag{2.21}$$

Enfin, les valeurs de probabilités *a priori* sont également ré-estimées de manière indépendante de la nature des composantes du mélange :

$$\alpha_k^{(r)} = \frac{1}{n_r n} \sum_{r=1}^{n_r} \sum_{x_i\in\mathcal{X}} z_{ik}^{(r)} \tag{2.22}$$

En règle générale le résultat produit par la répétition des deux étapes précédentes est tel qu'un désaccord nul ne puisse être trouvé. Ainsi, pour certains individus, on ne peut décider de leur appartenance à un groupe particulier. Ils peuvent appartenir à des groupes différents dans des vues différentes. Les auteurs proposent alors à la fin de l'algorithme d'appliquer une nouvelle règle MAP (maximum *a posteriori*) en observant les différents résultats locaux :

$$x_i \in C_k \Leftrightarrow k = \underset{k'\in[1..n_k]}{arg\,max}\, z_{ik'} = \frac{\prod_{r=1}^{n_r} \alpha_{k'}^{(r)} f_{k'}^{(r)}(x_i^{(r)};\theta_k^{(r')})}{\sum_{l=1}^{n_k} \prod_{r=1}^{n_r} \alpha_l^{(r)} f_k^{(r)}(x_i^{(r)};\theta_l^{(r)})} \tag{2.23}$$

Algorithme 14 CoEM

Entrées : $\mathcal{X}$, n_k, $\{f^{(r)}\}_{r\in[1..n_r]}$
Sorties : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation aléatoire des $\Theta^{(r)}$ $\forall r \in [1..n_r]$
2 : Étape E : Mise à jour des $z_{ik}^{(r)}$ en utilisant (2.21)
3 : Mise à jour des $z_{ik\eta}^{(r)}$ en utilisant (2.19)
4 : Étape M : Mise à jour des $\theta_k^{(r)}$ en utilisant (2.20)
5 : Mise à jour des $\alpha_k^{(r)}$ en utilisant (2.22)
6 : Si Q_{CoEM} change alors aller en **2**
7 : $C_k = \{x_i \in \mathcal{X} | z_{ik} = \underset{k'\in[1..n_k]}{max}\, z_{ik'}\}$, $\forall k \in [1..n_k]$

Discussion

Les auteurs donnent une formulation de Q_{CoEM} par une somme de log-vraisemblances sur chaque vue où les probabilités *a posteriori* sont obtenues par des moyennes pondérées des probabilités *a posteriori* locales. Malheureusement, en utilisant la nouvelle expression (cette fois intuitive) pour le calcul des probabilités *a posteriori* le critère ne peut pas être maximisé dans son ensemble sauf en annulant la contribution du désaccord *i.e.* $\eta \to 0$ et ainsi en ne tenant plus compte de la collaboration à travers les itérations. Notons qu'une affectation finale au groupe est obtenue à partir des paramètres du modèle collaboratif appris.

2.4 Contributions

2.4.1 Motivation

L'approche proposée CoFKM (*Collaborative Fuzzy K-means*) comme réponse à la problématique du *clustering* multi-vues offre une solution aux problèmes pratiques et théoriques rencontrés dans la plupart des approches de l'état de l'art. L'approche est de type discriminative et se fonde sur les développements effectués sur CoEM (cf. section 2.3.6) dans l'expression du critère objectif, puis sur CoFC (cf. section 2.3.2) dans la recherche d'une solution convergente sans artifice (tels que l'annulation de la recherche de consensus dans CoEM). Pour palier au problème de la convergence CoFKM se positionne dans le cadre flou de FKM (cf. section 1.4.1) ; un nouveau terme de désaccord (inspiré de CoEM) est proposé pour rendre le modèle plus simple à paramétrer et le processus d'apprentissage plus intuitif. Enfin, le paramètre η utilisé par Bickel & Sheffer pour assurer la convergence est conservé dans CoFKM car il permet de lier l'expression du critère aux différents paradigmes du *clustering* multi-vues : fusion *a priori*, *a posteriori* et dans le processus. Dans un second temps, l'objectif fixé dans le développement de l'approche CoFKM sera étendu pour la prise en compte de données relationnelles *i.e.* lorsque les données sont représentées par des matrices de proximité entre individus : similarité ou dissimilarité. L'extension CoKFKM est telle qu'elle offre les mêmes garanties de convergence que le modèle de base CoFKM. Enfin ces deux nouvelles approches sont testées sur des données standard afin de les valider expérimentalement.

2.4.2 CoFKM : *clustering* flou multi-vues

L'approche proposée est une extension des K-moyennes floues (cf. section 1.4.1). L'objectif est de produire un *clustering* global en intégrant pendant la phase de construction des groupes, les différentes représentations des individus.

Objectif

Pour rappel, le critère objectif de FKM à minimiser correspond à une inertie pondérée :

$$Q_{\text{FKM}}(c,u) = \sum_{k=1}^{n_k} \sum_{x_i \in \mathcal{X}} u_{ik}^{\beta} ||x_i - c_k||_2^2$$

avec $\sum_{k=1}^{n_k} u_{ik} = 1 \ \wedge u_{ik} \geq 0 \ \forall x_i \in \mathcal{X}$.

Les variables du problème sont les centres de groupes (c) et les degrés d'appartenance des individus x_i aux groupes (u). Partant d'une solution aléatoire des centres, l'expression du lagrangien du problème et la dérivation des conditions du premier ordre associées au problème permettent d'établir les mises à jour optimales des variables connaissant une solution courante.

Ces mises à jours sont données par :

$$c_k^* = \frac{\sum_{x_i \in \mathcal{X}} u_{ik}^{\beta} x_i}{\sum_{x_i \in \mathcal{X}} u_{ik}^{\beta}} \quad ; \quad u_{ik}^* = \frac{||x_i - c_k||_2^{2/(1-\beta)}}{\sum_{k'=1}^{n_k} ||x_i - c_{k'}||_2^{2/(1-\beta)}}$$

Soient c et u l'ensemble des centres et degrés tels que :

- $c = \{c^{(r)}\}_{r \in [1..n_r]}$ avec $c^{(r)} = \{c_1^{(r)}, \ldots, c_{n_k}^{(r)}\}$;
- $u = \{u^{(r)}\}_{r \in [1..n_r]}$ avec $u^{(r)} = \{u_{ik}^{(r)}\}_{\substack{x_i \in \mathcal{X} \\ k \in [1..n_k]}}$.

Suivant le formalisme général des approches de *clustering* multi-vues centralisées, on cherche à optimiser un critère global tel que la solution optimale soit une solution de compromis entre de bonnes solutions locales dans chaque vue :

$$Q_{\text{CoFKM}}(c, u) = \sum_{r=1}^{n_r} \text{objectif local}(r) + \text{désaccord}(\Pi) \tag{2.24}$$

Soit $Q_{\text{FKM}}^{(r)}$ le critère objectif de FKM dans la vue r, le critère objectif multi-vues proposé est défini par :

$$\begin{aligned} \text{objectif local}(r) &= Q_{\text{FKM}}^{(r)} \\ \text{désaccord}(\Pi) &= \Delta(\Pi) \end{aligned}$$

Le désaccord $\Delta(\Pi)$ permet de mesurer l'écart entre les *clusterings* locaux déterminés complètement par les degrés d'appartenances locaux, et les centres de groupes locaux. L'expression du désaccord peut alors être formulé par $\Delta(c, u)$ défini par :

$$\Delta(c, u) = \frac{1}{n_r - 1} \sum_{r=1}^{n_r} \sum_{\substack{\overline{r}=1 \\ \overline{r} \neq r}}^{n_r} \sum_{x_i \in \mathcal{X}} \sum_{k=1}^{n_k} \left((u_{ik}^{(\overline{r})^{\beta}} - u_{ik}^{(r)^{\beta}}) ||x_i^{(r)} - c_k^{(r)}||_2^2 \right)$$

Lorsque les *clusterings* locaux sont parfaitement similaires *i.e.* :

$$\forall x_i \in \mathcal{X} \; \forall k \in [1..n_k] \; \forall (r, \overline{r}) \in [1..n_r]^2, \; u_{ik}^{(r)} = u_{ik}^{(\overline{r})}$$

le terme $\Delta(c, u)$ est nul. Dans cette expression, on somme les différences entre les *clusterings* obtenues dans r et $\overline{r}$, $\forall (r, \overline{r}) \in [1..n_r]^2$. L'expression précédente peut-être écrite comme une somme sur les paires $(r, \overline{r})$ telles que $r > \overline{r}$:

$$\Delta(c, u) = \frac{1}{n_r - 1} \sum_{r=1}^{n_r} \sum_{\overline{r}=1}^{r-1} \sum_{x_i \in \mathcal{X}} \sum_{k=1}^{n_k} \left((u_{ik}^{(r')^{\beta}} - u_{ik}^{(r)^{\beta}}) (||x_i^{(r)} - c_k^{(r)}||_2^2 - ||x_i^{(\overline{r})} - c_k^{(\overline{r})}||_2^2) \right)$$

Le terme de désaccord pénalise le critère. Il peut être considéré comme une divergence entre les organisations puisque plus $(u_{ik}^{(r)^{\beta}} - u_{ik}^{(\overline{r})^{\beta}})$ est petit, plus faible est le désaccord.

Afin de conserver des inerties ($Q_{\text{FKM}}^{(r)}$) comparables entre les différentes vues, il est nécessaire de procéder à une normalisation des données :

- chaque descripteur de la vue r est réduit de telle sorte à obtenir une variance unitaire ;
- soit $n_p^{(r)}$ le nombre de descripteurs de la vue r, un poids égal à $n_p^{(r)^{-1/2}}$ est associé à chaque descripteur appartenant à la vue r, de manière à annuler l'impact du déséquilibre du nombre de dimensions entre vues.

La normalisation appliquée implique que $||x_i^{(\overline{r})} - c_k^{(\overline{r})}||_2^2$ et $||x_i^{(r)} - c_k^{(r)}||_2^2$ sont comparables. $||x_i^{(r)} - c_k^{(r)}||_2^2$ étant inversement proportionnel à $u_{ik}^{(r)}$, on peut considérer le terme $(||x_i^{(\overline{r})} - c_k^{(\overline{r})}||_2^2 - ||x_i^{(r)} - c_k^{(r)}||_2^2)$ comparable à $(u_{ik}^{(r)} - u_{ik}^{(\overline{r})})$. Ainsi, le désaccord peut-être vu comme une distance entre les *clusterings* locaux représentés par $\{u^{(r)}\}$ et $\{u^{(\overline{r})}\}$. L'avantage est que notre terme de désaccord a le même ordre de grandeur que l'inertie locale, ainsi la somme de ces expressions peut être considérée comme un critère global cohérent Q_{CoFKM}.

$$\begin{aligned} Q_{\text{CoFKM}}(c,u) &= \left(\sum_{r=1}^{n_r} Q_{\text{FKM}}^{(r)}\right) + \eta\Delta(c,u) \\ &= \left(\sum_{r=1}^{n_r}\sum_{x_i\in\mathcal{X}}\sum_{k=1}^{n_k}({u_{ik}^{(r)}}^{\beta}||x_i^{(r)} - c_k^{(r)}||_2^2)\right) + \eta\Delta(c,u) \\ &= \sum_{r=1}^{n_r}\sum_{x_i\in\mathcal{X}}\sum_{k=1}^{n_k}(u_{ik\eta}^{(r)}||x_i^{(r)} - c_k^{(r)}||_2^2) \end{aligned} \tag{2.25}$$

où

$$u_{ik\eta}^{(r)} = (1-\eta){u_{ik}^{(r)}}^{\beta} + \frac{\eta}{n_r - 1}(\sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r} {u_{ik}^{(\overline{r})}}^{\beta}) \tag{2.26}$$

L'objectif est alors la minimisation de ce critère d'inertie pénalisé Q_{CoFKM} sous les contraintes que chaque $u^{(r)}$ forme une partition floue :

$$\begin{array}{rrcll} \min\limits_{c,u} Q_{\text{CoFKM}}(c,u) & = & \min\limits_{c,u} \sum\limits_{r=1}^{n_r}\sum\limits_{k=1}^{n_k}\sum\limits_{x_i\in\mathcal{X}} u_{ik\eta}^{(r)}||x_i^{(r)} - c_k^{(r)}||_2^2 & & \\ s.t. & \sum_{k=1}^{n_k} u_{ik}^{(r)} & = & 1 \quad \forall x_i\in\mathcal{X},\ \forall r\in[1..n_r] & (cs1) \\ & u_{ik}^{(r)} & \geq & 0 \quad \forall x_i\in\mathcal{X},\ \forall k\in[1..n_k],\ \forall r\in[1..n_r] & (cs2) \end{array} \tag{2.27}$$

Algorithme

Comme dans la majorité des approches discriminatives basées sur un critère objectif, l'algorithme permettant d'en trouver une solution optimale découle directement de la résolution du problème d'optimisation. Ainsi, dans le cadre de l'optimisation sous contraintes, on considère le lagrangien $\mathcal{L}$ associé au problème :

$$\mathcal{L}(c,u,\lambda) = Q_{\text{CoFKM}} + \sum_{r=1}^{n_r}\sum_{x_i\in\mathcal{X}} \lambda_i^{(r)}(\sum_{k=1}^{n_k} u_{ik}^{(r)} - 1)$$

où $\lambda = \{\lambda_i^{(r)}\}_{\substack{x_i\in\mathcal{X}\\ r\in[1..n_r]}}$ sont les multiplicateurs de lagrange associés aux contraintes. Si (c^*, u^*) est un optimum (local), alors il existe un unique λ^* tel que c^*, u^* et λ^* satisfont les conditions du premier ordre suivantes :

$$\left\{\begin{array}{rcll} \dfrac{\partial\mathcal{L}(c^*,u^*,\lambda^*)}{\partial c_k^{(r)}} & = & 0 & (cond\ 1) \\ \dfrac{\partial\mathcal{L}(c^*,u^*,\lambda^*)}{\partial u_{ik}^{(r)}} & = & 0 & (cond\ 2) \\ \dfrac{\partial\mathcal{L}(c^*,u^*,\lambda^*)}{\partial {\lambda_i^{(r)}}^*} & = & 0 & (cond\ 3) \end{array}\right.$$

Les différentes dérivées partielles issues de $(cond\ 1)$, $(cond\ 2)$ et $(cond\ 3)$ mènent respectivement aux expressions:

$$\begin{aligned}
\frac{\partial \mathcal{L}(c^*,u^*,\lambda^*)}{\partial u_{ik}^{(r)}} &= (1-\eta)\beta {u_{ik}^{(r)}}^{*(\beta-1)} ||x_i^{(r)} - {c_k^{(r)}}^*||_2^2 \\
&\quad + \frac{\eta}{n_r-1}\beta {u_{ik}^{(r)}}^{*(\beta-1)} \left(\sum_{\substack{\overline{r}=1 \\ \overline{r}\neq r}}^{n_r} ||x_i^{(\overline{r})} - {c_k^{(\overline{r})}}^*||_2^2 \right) - \lambda_i^{(r)*} \\
\frac{\partial \mathcal{L}(c^*,u^*,\lambda^*)}{\partial c_k^{(r)}} &= -2 \sum_{x_i \in \mathcal{X}} \left({u_{ik\eta}^{(r)}}^* (x_i^{(r)} - {c_k^{(r)}}^*) \right) \\
\frac{\partial \mathcal{L}(c^*,u^*,\lambda^*)}{\partial \lambda_i^{(r)*}} &= \sum_{k=1}^{n_k} {u_{ik}^{(r)}}^* - 1
\end{aligned}$$

Comme pour FKM, l'algorithme (cf. Algorithme 15) propose, partant d'une solution initiale (c,u), de construire progressivement une solution meilleure au sens de l'objectif Q_{CoFKM}, en alternant consécutivement deux étapes d'optimisation :

- le calcul des centres optimaux ${c_k^{(r)}}^*$ à partir des degrés $u_{ik}^{(r)}$;
- le calcul des degrés optimaux ${u_{ik}^{(r)}}^*$ à partir des centres $c_k^{(r)}$.

Les suites ainsi construites convergent vers une solution localement optimale de Q_{CoFKM}. L'équation $\frac{\partial \mathcal{L}(c^*,u^*,\lambda^*)}{\partial \lambda_i^{(r)*}} = 0$ redonne la contrainte :

$$\frac{\partial \mathcal{L}(c^*,u^*,\lambda^*)}{\partial \lambda_i^{(r)*}} = 0 \Leftrightarrow \sum_{k=1}^{n_k} {u_{ik}^{(r)}}^* = 1 \tag{2.28}$$

Les équations $\frac{\partial \mathcal{L}(c^*,u^*,\lambda^*)}{\partial c_k^{(r)}} = 0$ et $\frac{\partial \mathcal{L}(c^*,u^*,\lambda^*)}{\partial u_{ik}^{(r)}} = 0$ impliquent respectivement :

$$\begin{aligned}
{c_k^{(r)}}^* &= \frac{\sum_{x_i \in \mathcal{X}} ({u_{ik\eta}^{(r)}}^* x_i^{(r)})}{\sum_{x_i \in \mathcal{X}} {u_{ik\eta}^{(r)}}^*} && (2.29) \\
{u_{ik\eta}^{(r)}}^* &= \left(\frac{\beta}{\lambda^{(r)*}} \right)^{1/(1-\beta)} \Bigg((1-\eta)||x_i^{(r)} - {c_k^{(r)}}^*||_2^2 && (2.30) \\
&\quad + \frac{\eta}{n_r-1} \sum_{\substack{\overline{r}=1 \\ \overline{r}\neq r}}^{n_r} ||x_i^{(\overline{r})} - {c_k^{(\overline{r})}}^*||_2^2 \Bigg)^{1/(1-\beta)}
\end{aligned}$$

L'équation (2.29) à condition de connaître la valeur courante de u, est sous forme close et correspond à la formule de mise à jour des centres. Cette expression est la même que celle de FKM où les degrés d'appartenance servant à pondérer le calcul du barycentre sont les degrés collaboratifs $u_{ik\eta}^{(r)}$ et non les degrés locaux. En utilisant la contrainte présente dans (2.28), on

peut déterminer la valeur de $\lambda^{(r)^*}$:

$$\sum_{k=1}^{n_k} u_{ik}^{(r)^*} = 1$$
$$\Leftrightarrow \sum_{k=1}^{n_k} \left(\left(\frac{\beta}{\lambda_i^{(r)}} \right)^{1/(1-\beta)} \left((1-\eta)||x_i^{(r)} - c_k^{(r)^*}||_2^2 + \frac{\eta}{n_r - 1} \sum_{\substack{\bar{r}=1 \\ \bar{r}\neq r}}^{n_r} ||x_i^{(\bar{r})} - c_k^{(\bar{r})^*}||_2^2 \right)^{1/(1-\beta)} \right) = 1$$

d'où

$$\lambda_i^{(r)^{*1/(1-\beta)}} = \beta^{1/(1-\beta)} \sum_{k=1}^{n_k} \left((1-\eta)||x_i^{(r)} - c_k^{(r)^*}||_2^2 + \frac{\eta}{n_r - 1} \sum_{\substack{\bar{r}=1 \\ \bar{r}\neq r}}^{n_r} ||x_i^{(\bar{r})} - c_k^{(\bar{r})^*}||_2^2 \right)^{1/(1-\beta)}$$

En réintroduisant cette expression dans (2.30), on est en mesure de déterminer seulement à partir de la valeur des centres, les nouveaux degrés d'appartenance :

$$u_{ik}^{(r)^*} = \frac{\left((1-\eta)||x_i^{(r)} - c_k^{(r)^*}||_2^2 + \frac{\eta}{n_r - 1} \sum_{\substack{\bar{r}=1 \\ \bar{r}\neq r}}^{n_r} ||x_i^{(\bar{r})} - c_k^{(\bar{r})^*}||_2^2 \right)^{1/(1-\beta)}}{\sum_{k'=1}^{n_k} \left((1-\eta)||x_i^{(r)} - c_{k'}^{(r)^*}||_2^2 + \frac{\eta}{n_r - 1} \sum_{\substack{\bar{r}=1 \\ \bar{r}\neq r}}^{n_r} ||x_i^{(\bar{r})} - c_{k'}^{(\bar{r})^*}||_2^2 \right)^{1/(1-\beta)}} \tag{2.31}$$

Finalement, partant d'une initialisation aléatoire des centres $c_k^{(r)}$, on calcule, à chaque étape :
- les valeurs optimales de $u_{ik}^{(r)^*}$ pour des valeurs fixées de $c_k^{(r)}$;
- les valeurs optimales de $c_k^{(r)^*}$ pour des valeurs fixées de $u_{ik}^{(r)}$.

Ainsi, par cet algorithme, la décroissance du critère Q_{CoFKM} est garantie, ce qui assure la convergence (vers un optimum local).

Construction de la partition finale

La méthode proposée assure l'obtention d'un optimum local du critère Q_{CoFKM}. Cependant, même si l'un des objectifs du critère compromis est d'obtenir pour chaque individu des profils d'appartenance aux groupes semblables dans toutes les vues, nous ne pouvons garantir que cette condition soit vérifiée par l'optimalité de la solution. Ainsi les centres de groupes et les degrés d'appartenance optimaux sont en général différents selon les vues. Le but étant d'obtenir un résultat de *clustering* unique, les résultats locaux dans chaque vue sont fusionnés au travers d'une règle d'affectation globale, permettant d'obtenir une partition stricte des individus. Cette règle nécessite de calculer, pour chaque individu $x_i \in \mathcal{X}$ et chaque groupe $C_k \in C$, un degré d'appartenance global, correspondant à une moyenne géométrique des degrés d'appartenance locaux :

$$u_{ik} = \left(\prod_{r=1}^{n_r} u_{ik}^{(r)} \right)^{1/n_r} \tag{2.32}$$

L'individu x_i est alors affecté au groupe C_k maximisant u_{ik} :

$$x_i \in C_k \Leftrightarrow k = \underset{k' \in [1..n_k]}{arg\ max}\, u_{ik'}$$

Cette règle, ainsi que le critère objectif lui-même, requiert l'association de chaque groupe simultanément dans toutes les vues. Dans ce contexte, un même groupe $C_k \in C$ est identifié par son indice $k \in [1..n_k]$ dans toutes les vues. Ainsi, les prototypes locaux $c_k^{(r)}$ se réfèrent au même et unique groupe C_k. La consistance de cette identification est suggérée par la façon dont sont initialisées les variables. L'initialisation consiste à choisir aléatoirement n_k individus comme centres de tous les groupes de même indice. Ainsi, pour tout $k \in [1..n_k]$, les centres c_{kr} correspondent à toutes les vues du même individu. Cependant, le processus de clustering peut entraîner une dérive de cette association.

Algorithme 15 CoFKM

ENTRÉES : $\mathcal{X}$, n_k
SORTIES : $C = \{C_1, ..., C_{n_k}\}$

1 : Initialisation aléatoire des $c_k^{(r)}$ sous la contrainte :

$$[\![\exists x_i \in \mathcal{X},\ (c_k^{(r)} = x_i^{(r)}) \wedge (c_k^{(r')} = x_i^{(r')})]\!]$$

2 : Mise à jour des $u_{ik}^{(r)}$ en utilisant (2.31)
3 : Mise à jour des $c_k^{(r)}$ en utilisant (2.29)
4: Si Q_{CoFKM} change alors aller en **2**
5: $C_k = \{x_i \in \mathcal{X} | u_{ik} = \underset{k' \in [1..n_k]}{max} u_{ik'}\}$, $\forall k \in [1..n_k]$

Discussion

L'approche proposée CoFKM est une généralisation :

- de FKM appliqué à la concaténation des différentes représentations, ce qui correspond à un mécanisme de fusion *a priori*;
- d'un cas simple de fusion *a posteriori* où FKM est appliqué simultanément et indépendamment dans toutes les représentations avant d'être concilié par la procédure d'affectation.

Généralisation d'une approche *a priori*. Considérons le critère Q_{CoFKM} pour lequel la valeur de η est fixée : $\eta = \frac{n_r - 1}{n_r}$. Le critère peut alors être réécrit :

$$\begin{aligned}
Q_{\text{CoFKM}}(c,u) &= \sum_{r=1}^{n_r}\sum_{k=1}^{n_k}\sum_{x_i \in \mathcal{X}} u_{ik\eta}^{(r)} ||x_i^{(r)} - c_k^{(r)}||_2^2 \\
&= \sum_{r=1}^{n_r}\sum_{k=1}^{n_k}\sum_{x_i \in \mathcal{X}} \left((1-\eta) {u_{ik}^{(r)}}^{*\beta} + \frac{\eta}{n_r - 1} (\sum_{\substack{\overline{r}=1 \\ \overline{r} \neq r}}^{n_r} {u_{ik}^{(\overline{r})}}^{*\beta}) \right) ||x_i^{(r)} - c_k^{(r)}||_2^2 \\
&= \sum_{r=1}^{n_r}\sum_{k=1}^{n_k}\sum_{x_i \in \mathcal{X}} \left((1 - \frac{n_r - 1}{n_r}) {u_{ik}^{(r)}}^{*\beta} + \frac{(n_r - 1)}{n_r(n_r - 1)} \sum_{\substack{\overline{r}=1 \\ \overline{r} \neq r}}^{n_r} {u_{ik}^{(\overline{r})}}^{*\beta}) \right) ||x_i^{(r)} - c_k^{(r)}||_2^2 \\
&= \sum_{r=1}^{n_r}\sum_{k=1}^{n_k}\sum_{x_i \in \mathcal{X}} \left(\frac{1}{n_r} \sum_{r'=1}^{n_r} {u_{ik}^{(r')}}^{*\beta} \right) ||x_i^{(r)} - c_k^{(r)}||_2^2
\end{aligned}$$

La valeur de $u_{ik}^{(r')^*}$ peut être déterminée, toujours selon (2.31) et restreint à $\eta = \frac{n_r-1}{n_r}$:

$$\begin{aligned} u_{ik}^{(r)^*} &= \frac{\left((1-\eta)||x_i^{(r)} - c_k^{(r)^*}||_2^2 + \frac{\eta}{n_r-1}\sum_{\substack{\bar{r}=1\\ \bar{r}\neq r}}^{n_r} ||x_i^{(\bar{r})} - c_k^{(\bar{r})^*}||_2^2\right)^{1/(1-\beta)}}{\sum_{k'=1}^{n_k}\left((1-\eta)||x_i^{(r)} - c_{k'}^{(r)^*}||_2^2 + \frac{\eta}{n_r-1}\sum_{\substack{\bar{r}=1\\ \bar{r}\neq r}}^{n_r} ||x_i^{(\bar{r})} - c_{k'}^{(\bar{r})^*}||_2^2\right)^{1/(1-\beta)}} \\ &= \frac{\left(\frac{1}{n_r}||x_i^{(r)} - c_k^{(r)^*}||_2^2 + \frac{1}{n_r}\sum_{\substack{\bar{r}=1\\ \bar{r}\neq r}}^{n_r} ||x_i^{(\bar{r})} - c_k^{(\bar{r})^*}||_2^2\right)^{1/(1-\beta)}}{\sum_{k'=1}^{n_k}\left(\frac{1}{n_r}||x_i^{(r)} - c_{k'}^{(r)^*}||_2^2 + \frac{1}{n_r}\sum_{\substack{\bar{r}=1\\ \bar{r}\neq r}}^{n_r} ||x_i^{(\bar{r})} - c_{k'}^{(\bar{r})^*}||_2^2\right)^{1/(1-\beta)}} \end{aligned}$$

et en utilisant le fait que la somme des carrés des distances aux centres locales correspond aux carrés des distances aux centres dans l'espace concaténé :

$$\sum_{r=1}^{n_r} ||x_i^{(r)} - c_k^{(r)^*}||_2 = ||x_i - c_k^*||_2$$

où x_i correspond à la concaténation des vecteurs $x_i^{(r)}$. Les degrés optimaux se réécrivent alors :

$$u_{ik}^{(r)^*} = \frac{||x_i - c_k^*||_2^2}{\sum_{k'=1}^{n_k} ||x_i - c_{k'}{}^*||_2^2} \tag{2.33}$$

et ainsi $u_{ik}^{(r)^*} = u_{ik}^{(r')^*}$ $\forall x_i \in \mathcal{X}$, $\forall r \in [1..n_r]$, $\forall k \in [1..n_k]$. Les degrés locaux $u_{ik}^{(r)^*}$ sont donc indépendants de r et peuvent être notés u_{ik}.

Le critère Q_{CoFKM} se réécrit dans ce contexte :

$$\begin{aligned} Q_{\text{CoFKM}}(c,u) &= \sum_{r=1}^{n_r}\sum_{k=1}^{n_k}\sum_{x_i\in\mathcal{X}} u_{ik}^{\beta}||x_i^{(r)} - c_k^{(r)}||_2^2 \\ &= \sum_{k=1}^{n_k}\sum_{x_i\in\mathcal{X}} u_{ik}^{\beta}||x_i - c_k||_2^2 \end{aligned}$$

Finalement, on peut voir CoFKM comme une généralisation de FKM appliquée à la concaténation des représentations vectorielles, où l'on peut forcer l'obtention d'une solution correspondant à un consensus en choisissant une valeur $\eta < \frac{(n_r-1)}{n_r}$.

Généralisation d'une approche *a posteriori*. Soit $\eta = 0$ le critère Q_{CoFKM} peut alors être réécrit comme une somme sur toutes les vues des critères FKM classiques :

$$\begin{aligned} Q_{\text{CoFKM}_{\eta=0}}(c,u) &= \left(\sum_{r=1}^{n_r} Q_{\text{FKM}}^{(r)}\right) \\ &= \sum_{r=1}^{n_r}\sum_{x_i\in\mathcal{X}}\sum_{k=1}^{n_k} u_{ik}^{(r)^{\beta}}||x_i^{(r)} - c_k^{(r)}||_2^2 \end{aligned}$$

Les mises à jour optimales des variables du problème sont alors données par :

$$c_k^{(r)*} = \frac{\sum_{x_i \in \mathcal{X}} {u_{ik}^{(r)}}^{\beta} x_i^{(r)}}{\sum_{x_i \in \mathcal{X}} {u_{ik}^{(r)}}^{\beta}} \quad , \quad u_{ik}^{(r)*} = \frac{||x_i^{(r)} - c_k^{(r)}||_2^{2/(1-\beta)}}{\sum_{k'=1}^{n_k} ||x_i^{(r)} - c_{k'}^{(r)}||_2^{2/(1-\beta)}} \tag{2.34}$$

Le critère est la somme des inerties locales, qui sont optimisées de manières indépendantes par l'algorithme FKM, les mises à jour étant identiques modulo un renommage des variables. La fusion *a posteriori* est réalisée par notre règle d'affectation finale (2.32). Le formalisme collaboratif proposé CoFKM est alors une généralisation de la fusion *a posteriori*, en choisissant $\eta = 0$.

Comparaison avec l'état de l'art. Les approches auxquelles nous nous comparons tant au niveau de l'expression du critère qu'au niveau expérimental sont les approches CoFC et CoEM. L'inconvénient majeur de CoEM (cf. section 2.3.6) réside en la non convergence de l'algorithme proposé pour trouver les meilleurs estimateurs des paramètres Θ. Pour assurer cette convergence, [Bickel and Scheffer, 2005] proposent de faire décroître le paramètre η jusqu'à 0, ce qui correspond à l'optimisation du critère local indépendamment dans toutes les vues et tend à revenir à un mécanisme de fusion *a posteriori*. CoEM peut ainsi être vu comme une approche en deux temps :

1. Durant la première phase ($\eta > 0$) les paramètres sont estimés dans le but d'accroître le consensus mais sans garanties de convergence.
2. Lors de la seconde phase ($\eta = 0$) la valeur du critère global converge par convergence locale dans toutes les vues, mais le terme de pénalité n'est pas considéré.

Le modèle CoFKM est défini de telle sorte que quelque soit la valeur de η, la convergence est assurée puisque le critère global décroît à chacune des étapes de l'algorithme.

En ce qui concerne l'approche CoFC, celle-ci offre de bonnes propriétés de convergence, mais souffre de deux lacunes au regard de FKM qu'elle vise à étendre :
- un manque de généricité dans le sens où il n'est plus possible de moduler la recherche de solution grâce au paramètre de flou β de FKM.
- un manque d'interprétabilité des équations de mise à jour des prototypes et des degrés d'appartenance.

La contribution CoFKM intègre le paramètre de flou et généralise complètement l'algorithme FKM pour le traitement de données multi-représentées par des représentations vectorielles. Les procédures de mises à jour des variables sont intuitives et s'interprètent bien de sorte à faire ressortir la recherche d'un compromis entre les différentes vues.

2.4.3 CoKFKM : *clustering* flou multi-vues à noyaux

CoFKM généralise le modèle classique des K-moyennes floues mais se voit toujours restreint à l'utilisation de la métrique euclidienne. En particulier, ce modèle ne s'applique que dans le cas où les données sont décrites par des vecteurs d'attributs numériques. L'objectif CoKFKM, objet de cette section est d'étendre CoFKM pour le rendre applicable dans le cas où les données sont représentées par plusieurs matrices de proximité. Cette extension est réalisée grâce à l'utilisation de l'astuce du noyau dans un cadre d'apprentissage non supervisé.

Astuce du noyau

L'astuce du noyau a été appliquée de nombreuses fois pour des utilisations variées. L'idée est de réaliser une projection de l'ensemble d'individus d'un espace d'origine X à un nouvel espace $\mathcal{H}$ afin de faciliter la recherche d'un meilleur *clustering* de $\mathcal{X}$. L'objectif est double, l'utilisation d'un noyau permet :

- d'augmenter les chances de capturer les vraies classes des individus, lorsque ceux-ci ne sont pas linéairement séparables dans l'espace de représentation d'origine. Cela permet d'améliorer les performances des approches traditionnelles de *clustering* lorsque l'on peut les évaluer par rapport à une classification de référence.
- de pouvoir étendre tout type d'approche fondée sur la distance euclidienne, qu'elle permet de redéfinir par la définition d'une matrice de proximité.

Soit ϕ la fonction telle que $\phi : X \mapsto \mathcal{H}$. $\phi(x_i)$ est la projection de x_i dans $\mathcal{H}$. La distance euclidienne dans l'espace $\mathcal{H}$ s'exprime par :

$$||\phi(x_i) - \phi(x_j)||_2^2 = \langle \phi(x_i), \phi(x_i) \rangle - 2\langle \phi(x_i), \phi(x_j) \rangle + \langle \phi(x_j), \phi(x_j) \rangle$$

L'astuce consiste alors à interpréter le produit scalaire $\langle \phi(x_i), \phi(x_j) \rangle$ comme une mesure de similarité. Ainsi si on a à disposition une matrice K telle que $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$ ou un moyen de construire K à partir de X, alors on peut complètement redéfinir la distance euclidienne dans $\mathcal{H}$ et appliquer les algorithmes de *clustering* dans cet espace tout en conservant les bonnes propriétés de ceux-ci :

$$||\phi(x_i) - \phi(x_j)||_2^2 = K_{ii} - 2K_{ij} + K_{jj}$$

Il n'est alors pas nécessaire de calculer explicitement $\phi(x_i)$ $\forall x_i \in \mathcal{X}$ pour calculer cette distance.

Dans FKM, le critère objectif est modifié de sorte à réaliser le *clustering* de $\mathcal{X}$ dans $\mathcal{H}$. Ainsi la partition floue solution est un optimum du critère objectif :

$$Q_{\text{KFKM}} = \sum_{k=1}^{n_k} \sum_{x_i \in \mathcal{X}} u_{ik}^{\beta} ||\phi(x_i) - c_k||_2^2$$

Les valeurs optimales des variables c_k et u_{ik} sont données par :

$$c_k^* = \frac{\sum_{x_i \in \mathcal{X}} u_{ik}^{\beta} \phi(x_i)}{\sum_{x_i \in \mathcal{X}} u_{ik}^{\beta}} \quad ; \quad u_{ik}^* = \frac{||\phi(x_i) - c_k||_2^{2/(1-\beta)}}{\sum_{k=1}^{n_k} ||\phi(x_i) - c_k||_2^{2/(1-\beta)}}$$

Il a été montré que même si les centres optimaux ne peuvent pas être calculés (car ϕ est en général inconnue), on peut optimiser le critère Q_{KFKM} grâce à K sous réserve que K soit semi-définie positive. On peut alors calculer le carré de la distance euclidienne entre $\phi(x_i)$ et c_k :

$$||\phi(x_i) - c_k||_2^2 = K_{ii} - 2\frac{\sum_{x_j \in \mathcal{X}} u_{jk}^{\beta} K_{ij}}{\sum_{x_j \in \mathcal{X}} u_{jk}^{\beta}} + \frac{\sum_{x_j \in \mathcal{X}} \sum_{x_l \in \mathcal{X}} u_{jk}^{\beta} u_{lk}^{\beta} K_{jl}}{(\sum_{x_j \in \mathcal{X}} u_{jk}^{\beta})^2}$$

Les centres sont implicitement déplacés dans l'espace de projection lors du calcul des nouvelles distances (dépendantes des nouveaux estimateurs des degrés d'appartenances). On peut alors transposer ce résultat à l'approche CoFKM. On pose dans la suite :

- $c = \{c^{(r)}\}_{r \in [1..n_r]}$ avec $c^{(r)} = \{c_1^{(r)}, \ldots, c_{n_k}^{(r)}\}$;
- $u = \{u^{(r)}\}_{r \in [1..n_r]}$ avec $u^{(r)} = \{u_{ik}^{(r)}\}_{\substack{x_i \in \mathcal{X} \\ k \in [1..n_k]}}$.

Objectif

Soit $\phi = \{\phi^{(r)}\}_{r\in[1..n_r]}$ telle que $\phi^{(r)} : X^{(r)} \mapsto \mathcal{H}^{(r)}$, le critère Q_{CoFKM} peut alors être réécrit en Q_{CoKFKM} pour obtenir une version à noyaux :

$$\begin{aligned} Q_{\text{CoKFKM}} &= \left(\sum_{r=1}^{n_r} Q^{(r)}_{\text{KFKM}}\right) + \eta\Delta(c,u) \\ &= \sum_{r=1}^{n_r}\sum_{x_i\in\mathcal{X}}\sum_{k=1}^{n_k} {u^{(r)}_{ik}}^{\beta} ||\phi^{(r)}(x^{(r)}_i) - c^{(r)}_k||^2_2 + \eta\Delta(c,u) \end{aligned} \tag{2.35}$$

avec

$$\Delta(c,u) = \frac{1}{n_r-1}\sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r}\sum_{x_i\in\mathcal{X}}\sum_{k=1}^{n_k}({u^{(\overline{r})}_{ik}}^{\beta} - {u^{(r)}_{ik}}^{\beta})||\phi^{(r)}(x^{(r)}_i) - c^{(r)}_k||^2_2$$

A l'instar de CoFKM, le *clustering* multi-vues par CoKFKM peut également être exprimé par le problème d'optimisation :

$$\begin{aligned} \min_{c,u} Q_{\text{CoKFKM}}(c,u) &= \min_{c,u} \sum_{r=1}^{n_r}\sum_{k=1}^{n_k}\sum_{x_i\in\mathcal{X}} u^{(r)}_{ik\eta}||\phi^{(r)}(x^{(r)}_i) - c^{(r)}_k||^2_2 \\ s.t. \quad \textstyle\sum_{k=1}^{n_k} u^{(r)}_{ik} &= 1 \quad \forall x_i\in\mathcal{X},\ \forall r\in[1..n_r] \quad (cs1) \\ u^{(r)}_{ik} &\geq 0 \quad \forall x_i\in\mathcal{X},\ \forall k\in[1..n_k],\ \forall r\in[1..n_r] \quad (cs2) \end{aligned} \tag{2.36}$$

avec

$$u^{(r)}_{ik\eta} = (1-\eta){u^{(r)}_{ik}}^{\beta} + \frac{\eta}{n_r-1}\sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r} {u^{(\overline{r})}_{ik}}^{\beta} \tag{2.37}$$

Algorithme

L'algorithme permettant de résoudre ce problème d'optimisation est dérivé directement du critère à la manière de CoFKM. Il s'agit d'un processus qui, partant d'une initialisation particulière des prototypes des groupes, alterne une mise à jour optimale des degrés d'appartenance des individus aux groupes, puis une mise à jour des prototypes des groupes (cf. algorithme 16).

Les degrés d'appartenance sont réévalués de manière optimale de la même manière que dans CoFKM mais les distances euclidiennes utilisées sont associées aux espaces $\mathcal{H}^{(r)}$:

$${u^{(r)}_{ik}}^{*} = \frac{\left((1-\eta)||\phi^{(r)}(x^{(r)}_i) - c^{(r)}_k||^2_2 + \frac{\eta}{n_r-1}\sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r}||\phi^{(\overline{r})}(x^{(\overline{r})}_i) - c^{(\overline{r})}_k||^2_2\right)^{1/(1-\beta)}}{\sum_{k=1}^{n_k}\left((1-\eta)||\phi^{(r)}(x^{(r)}_i) - c^{(r)}_k||^2_2 + \frac{\eta}{n_r-1}\sum_{\substack{\overline{r}=1\\ \overline{r}\neq r}}^{n_r}||\phi^{(\overline{r})}(x^{(\overline{r})}_i) - c^{(\overline{r})}_k||^2_2\right)^{1/(1-\beta)}} \tag{2.38}$$

L'équation de mise à jour des prototypes des groupes est également connue et consiste à calculer les centres de masse des différents groupes :

$${c^{(r)}_k}^{*} = \frac{\sum_{x_i\in\mathcal{X}} {u^{(r)}_{ik\eta}}^{\beta}\phi^{(r)}(x^{(r)}_i)}{\sum_{x_i\in\mathcal{X}} {u^{(r)}_{ik\eta}}^{\beta}}$$

Néanmoins, comme dans toute approche à noyaux, la projection $\phi^{(r)}(x_i^{(r)})$ n'est pas calculable ou il n'est pas souhaitable de la calculer, la mise à jour ne peut avoir lieu explicitement. Ainsi, après avoir déterminé les valeurs de $u_{ik\eta}^{(r)}$ par (2.37), les centre optimaux peuvent être déterminés implicitement par la réévaluation des distances $d_{(r)}(x_i, c_k{}^*) = ||\phi^{(r)}(x_i^{(r)}) - c_k^{(r)^*}||_2$ dans $\mathcal{H}^{(r)}$:

$$\begin{aligned} d^2_{(r)}(x_i, c_k{}^*) &= ||\phi^{(r)}(x_i^{(r)}) - c_k^{(r)^*}||_2^2 \\ &= K_{ii}^{(r)} - 2\frac{\sum_{x_j \in \mathcal{X}} u_{jk\eta}^{(r)\ \beta} K_{ij}^{(r)}}{\sum_{x_j \in \mathcal{X}} u_{jk\eta}^{(r)\ \beta}} + \frac{\sum_{x_j \in \mathcal{X}} \sum_{x_l \in \mathcal{X}} u_{jk\eta}^{(r)\ \beta} u_{ik\eta}^{(r)\ \beta} K_{jl}^{(r)}}{(\sum_{x_j \in \mathcal{X}} u_{jk\eta}^{(r)\ \beta})^2} \end{aligned} \tag{2.39}$$

Une fois le processus itératif terminé, des degrés d'appartenance aux groupes globaux u_{ik} sont calculés, à la manière de CoFKM, selon l'équation (2.32)

Algorithme 16 CoKFKM

ENTRÉES : $\mathcal{X}$, n_k, $\{K^{(r)}\}_{r \in [1..n_r]}$
SORTIES : $C = \{C_1, ..., C_{n_k}\}$

1 : Initialisation aléatoire des $c_k^{(r)}$ sous la contrainte :

$$[\![\ \exists x_i \in \mathcal{X},\ (c_k^{(r)} = x_i^{(r)}) \wedge (c_k^{(\overline{r})} = x_i^{(\overline{r})})\ ]\!]$$

2 : Mise à jour des $u_{ik}^{(r)^*}$ en utilisant (2.38)
3 : Mise à jour des $u_{ik\eta}^{(r)\ *}$ en utilisant (2.37)
4 : Mise à jour des $d_{(r)}(x_i, c_k{}^*)$ par (2.39)
5 : Si Q_{CoFKM} change alors aller en **2**
6 : $C_k = \{x_i \in \mathcal{X} | u_{ik} = \max_{k' \in [1..n_k]} u_{ik'}\},\ \forall k \in [1..n_k]$

Discussion

La version à noyaux CoKFKM généralise complètement CoFKM. En effet, il suffit de choisir comme matrices noyaux pour chaque vue les matrices des produits scalaires individus dans l'espace de description d'origine $X^{(r)}$. Soit $K_{ij}^{(r)} = \langle \phi(x_i^{(r)}), \phi(x_j^{(r)}) \rangle = \langle x_i^{(r)}, x_j^{(r)} \rangle$, alors on a bien $||\phi^{(r)}(x_i^{(r)}) - c_k^{(r)}||_2^2 = ||x_i^{(r)} - c_k^{(r)}||_2^2$.

Le critère optimisé correspond exactement à celui de CoFKM appliqué cette fois dans $\mathcal{H} = \{\mathcal{H}^{(r)}\}_{r \in [1..n_r]}$. L'intérêt de CoKFKM réside essentiellement dans la possibilité d'utiliser différentes matrices de proximité, en particulier des matrices de similarité, plus adaptées aux données. Cependant l'utilisation de cette astuce peut avoir un coût, notamment du point de vue de la complexité qui est présenté par la suite.

Complexité algorithmique

L'objectif de ce paragraphe est ici d'étudier les pertes associées à l'utilisation de CoKFKM (plus général) par rapport à CoFKM, au sens de la complexité algorithmique. L'algorithme CoFKM (cf. algorithme 15) se décompose en trois étapes :

1. Le calcul des degrés d'appartenances locaux $u_{ik}^{(r)}$ par (2.31).
   Pour chaque x_i, k et r, une somme pondérée sur les vues $\overline{r}$ des distances aux prototypes est calculée. La distance dans une vue r se calculant en $\mathcal{O}(n_p^{(r)})$, le calcul de $u_{ik}^{(r)}$ s'effectue alors en $\mathcal{O}(n_r.n_p^{(r)})$. L'étape de mise à jour complète des degrés a pour complexité au pire des cas $\mathcal{O}(n_k.n_r^2.n.\sum_{r=1}^{n_r} n_p^{(r)})$.
2. Le calcul des degrés collaboratifs $u_{ik\eta}^{(r)}$ par (2.26).
   Il suffit de calculer pour chaque x_i,k et r une somme pondérée sur les vues des degrés locaux déjà évalués. La mise à jour de tous les $u_{ik\eta}^{(r)}$ se fait ainsi en $\mathcal{O}(n_k.{n_r}^2.n)$.
3. Le calcul des centres $c_k^{(r)}$ par (2.29).
   Il suffit de calculer pour chaque k et r une moyenne pondérée sur les individus. La mise à jour de tous les $c_k^{(r)}$ a un coût de $\mathcal{O}(n_k.n_r.n)$.

La complexité à l'issue des trois étapes devient $\mathcal{O}(n_k.n_r.n(1+n_r+(\sum_{r=1}^{n_r} n_p^{(r)}).n_r))$. La complexité de CoFKM est alors $\mathcal{O}(n_k.n_r.n(((\sum_{r=1}^{n_r} n_p^{(r)})+1).n_r))$.

Dans le cas de l'algorithme CoKFKM (cf. algorithme 16), des trois étapes de calcul, seule la dernière change, puisqu'il n'est pas possible de calculer explicitement les centres dans l'espace de projection. Ceux-ci sont déplacés implicitement pendant le calcul des distances. De ce fait ces distances sont désormais stockées en mémoire, ce qui n'était pas nécessaire dans CoFKM, ainsi :

1. Le calcul des degrés d'appartenances est moins coûteux : $\mathcal{O}(n_k.{n_r}^2.n)$.
2. Le coût du calcul des degrés collaboratifs est inchangé : $\mathcal{O}(n_k.{n_r}^2.n)$.
3. La mise à jour des distances aux centres par (2.39) se réalise en $\mathcal{O}(n_k.n_r.n^2)$.

La complexité au pire des cas, à l'issue des trois étapes, est de l'ordre de $\mathcal{O}(n_k.n_r.n(n+2.n_r))$. Si on émet les hypothèses suivantes (largement vérifiées dans les cas concrets d'applications)

- $n >> n_r$ *i.e.* on a à disposition plus d'individus que de vues ;
- $\sum_{r=1}^{n_r} n_p^{(r)} >> n_r$ *i.e.* la dimensionnalité de la concaténation des représentations vectorielles de chaque vue est largement plus élevé que le nombre de vues ;

alors les complexités des deux approches à comparer deviennent :

CoFKM : $\mathcal{O}(n_k.n_r.n.(\sum_{r=1}^{n_r} n_p^{(r)}))$;

CoKFKM : $\mathcal{O}(n_k.n_r.n.n)$.

En d'autres termes, si le nombre d'individus n est beaucoup plus grand que la somme des dimensionnalités $n_p^{(r)}$, alors l'approche CoFKM est moins complexe et plus rapide d'exécution. En revanche, dans le cas de la malédiction de la dimensionnalité, où le nombre d'attributs est beaucoup plus grand que le nombre d'individus, l'approche à noyaux devient moins complexe, et se justifie alors comme une variante efficace.

2.5 Évaluation

Les approches CoFKM et CoKFKM ont été validées expérimentalement en suivant différentes procédures d'évaluation internes et externes. Les jeux de données qui ont servi de base de validation sont tirés de travaux de recherche comme celui de [Strehl and Ghosh, 2003] [1] ou de bases de données disponibles en ligne telles l'*UCI Machine Learning Repository* [2] ou *WebKB* [3].

1. *http://strehl.com/*
2. *http://archive.ics.uci.edu/ml/*
3. *http://www.mpi-inf.mpg.de/ bickel/mvdata/*

2.5.1 Données

Le premier jeu de données *multiple features* ou *mfeat* correspond à un ensemble de 2000 chiffres manuscrits (images) numérisées par six techniques d'encodage d'images :
- les coefficients de Fourier : $X_1 \in [0,1]^{2000\times 76}$;
- les corrélations de profils : $X_2 \in \mathbb{N}^{2000\times 216}$;
- les coefficients de Karhunen-Loève : $X_3 \in \mathbb{R}^{2000\times 64}$;
- les descripteurs morphologiques : $X_4 \in \mathbb{R}^{2000\times 6}$;
- les nombres de pixels dans des fenêtres 2×3 : $X_5 \in \mathbb{N}^{2000\times 240}$;
- les moments de Zernike : $X_6 \in \mathbb{R}^{2000\times 47}$.

Ainsi, chaque individu (chiffre) est représenté par six représentations vectorielles et chacune de ces représentations est insuffisante pour retrouver les différents groupes d'images représentant un même chiffre. Dix classes sont à retrouver (les chiffres de 0 à 9), avec 200 individus par classe.

Le jeu *2D2K* contient 1000 individus générés par un mélange de deux gaussiennes bidimensionnelles sphériques (pour une classe donnée, la valeur de variance est égale dans les deux dimensions). À partir de ces données bidimensionnelles, trois représentations sont construites artificiellement :
- la première vue correspond à la première dimension : $X_1 \in \mathbb{R}^{1000\times 1}$;
- la seconde vue correspond à la seconde dimension : $X_2 \in \mathbb{R}^{1000\times 1}$;
- la troisième vue correspond de nouveau à la première dimension : $X_3 \in \mathbb{R}^{1000\times 1}$.

Deux classes sont à retrouver et s'identifient avec les deux composantes du mélange.

WebKB est un jeu de donnée réel correspondant à une collection de 4501 pages web académiques tirées d'universités des États-Unis (Cornell, Texas, Washington et Wisconsin) et regroupées manuellement en six classes de pages concernant respectivement les étudiants, la faculté, le personnel, les départements, les cours et les projets de recherche. Deux représentations sont disponibles :
- la première vue concerne le texte de chaque page web : $X_1 \in \mathbb{N}^{4501\times 25000}$;
- la seconde vue correspond au texte de tous les liens entrants : $X_2 \in \mathbb{N}^{4501\times 900}$.

La première représentation est très volumineuse en terme de dimensionnalité et les deux prennent la forme de matrices très creuses. Ceci constitue un défi pour les méthodes de classifications actuelles, et se retrouve fréquemment dans les applications de type fouille de textes ou fouille du web. Les classes sont cette fois non homogènes en taille et les vues sont très déséquilibrées et inégales quant à la quantité d'informations qu'elles apportent.

2.5.2 Protocole expérimental

Les deux premiers jeux de données ont servi à valider principalement l'approche CoFKM dédiée au cas où les individus sont définis par des représentations vectorielles. Le troisième jeu de donnée valide l'apport de l'extension à noyaux CoKFKM.

Tous les jeux de données se sont vu appliqués la normalisation imposée par CoFKM selon un principe d'équité entre toutes les représentations, et entre tous les attributs de chaque représentation. Dans un premier temps, différentes expériences ont été conduites dans le but de justifier l'intérêt des approches collaboratives centralisées comparées aux approches *a priori* (par concaténation) et *a posteriori*, d'une part en détaillant les gains de performances obtenus par rapport à ces techniques, et d'autre part en caractérisant la solution consensus en terme d'évaluation

interne. Dans un second temps, la performance de CoFKM est étudiée comparativement aux approches de l'état de l'art telles CoFC et CoEM.

Les résultats obtenus correspondent à une moyenne de 20 exécutions pour *multiple features*, 100 exécutions pour *2D2K* et 10 exécutions pour *WebKB*. Les différentes méthodes ont été comparées chaque fois avec la même initialisation. Les paramètres de CoFKM sont fixés à $\beta = 1.25$ (valeur couramment employée) lorsque la performance de l'algorithme n'est pas évaluée selon ce paramètre, et $\eta = \frac{n_r - 1}{2 \times n_r}$, ce qui correspond à une valeur *heuristique* de collaboration entre les versions *a priori* ($\eta = \frac{n_r - 1}{n_r}$) et *a posteriori* ($\eta = 0$) de CoFKM.

En ce qui concerne CoEM(et EM), l'estimation des paramètres d'un modèle de mélange gaussien général est inefficace, différents modèles parcimonieux ont alors été observés :
- le cas des matrices de variances/covariances de la forme $\sigma_k.I$ (vs1) ;
- le cas des matrices de la forme $\sigma.I$ (le même σ pour toutes les composantes du mélange) (vs2) ;
- le cas des matrices diagonales (vs3).

Le paramètre η de CoEM quant à lui décroît progressivement pour garantir la convergence. Pour l'application de l'algorithme CoFC, il n'est pas spécifié que l'application de l'algorithme puisse se faire de manière simultanée sur tous les sites (les différentes vues). Plusieurs cas ont alors été envisagés dans les tests comparatifs :
- CoFC-*vue* réalise un FKM indépendant dans chaque vue. Les matrices de partitions floues résultantes sont ensuite fixées pour toutes les vues sauf celle dans laquelle se déroule le *clustering* par CoFC.
- CoFCGlobal-*vue* réalise un FKM dans chaque vue, mais cette fois les matrices de partitions floues évoluent par CoFC simultanément dans dans toutes les vues.

2.5.3 Évaluation interne

Un premier objectif justifiant l'intérêt des approches centralisées concerne la stabilité de la qualité du *clustering* final au regard de chacune des vues. L'idée est ici d'observer si le *clustering* obtenu à l'issue du processus collaboratif est bon sur chacune des vues. Une telle observation confirmerait l'idée qu'une bonne solution globale peut être obtenue tout en assurant que toutes les vues s'accordent pour conforter la qualité de cette solution. La procédure d'évaluation interne est la suivante :
- on compare les critères internes (inerties) obtenues par CoFKM et ses variantes *a priori* et *a posteriori*;
- on observe les valeurs de ses critères dans chacune des vues, et ceci à la fois avant et après la règle d'affectation (2.32).

L'objectif visé est qu'une solution consensus soit bonne sur toutes les vues (stable) au sens du critère interne avant la règle d'affectation, et que cette règle ne détériore pas trop cette stabilité.

Les figures 2.3 et 2.4 confirment l'intuition sur les approches multi-vues centralisées. Dans les deux cas, au sens du critère interne et avant fusion, CoFKM permet d'apprendre une solution meilleure que celle de sa variante concaténée (*a priori*) et surtout l'écart entre les inerties locales est plutôt faible dans le cas de l'approche centralisée (ce qui traduit la stabilité de la solution sur toutes les vues). La version *a posteriori* est celle qui optimise localement les inerties (sans collaborations entre les vues), elle se positionne comme une référence (avant fusion). En revanche, si l'on observe l'impact de la règle d'affectation permettant d'obtenir un *clustering* unique pour toutes les vues, la qualité de l'approche sans collaborations se détériore complètement. Le résultat de référence après fusion est la concaténation qui reste inchangée puisque le degré d'appartenance d'un individu aux groupes est le même dans toutes les vues (avant ou après la règle).

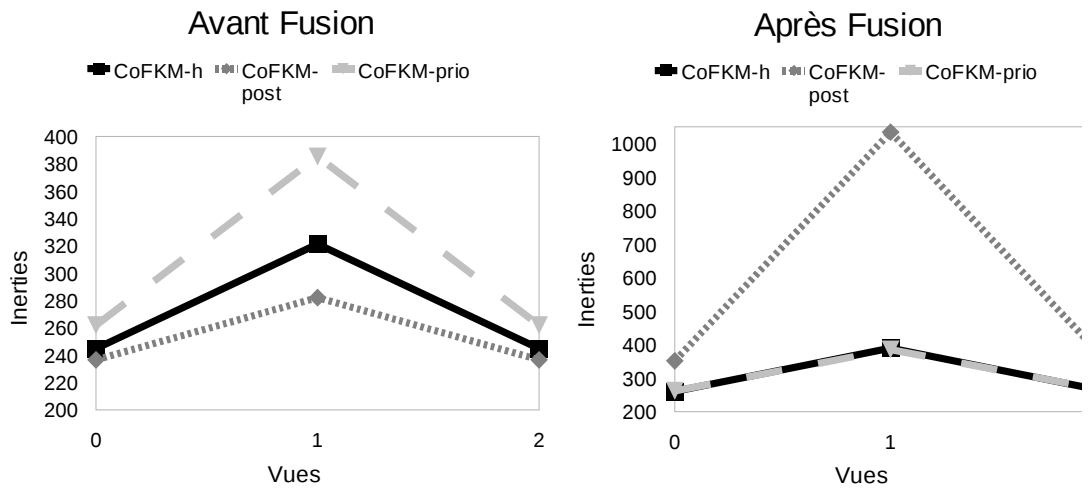

FIGURE 2.3 — Comparaisons des valeurs de critère interne dans chaque vue avant et après fusion (règle d'affectation) pour CoFKM et ses variantes *a priori* et *a posteriori* pour *2D2K*.

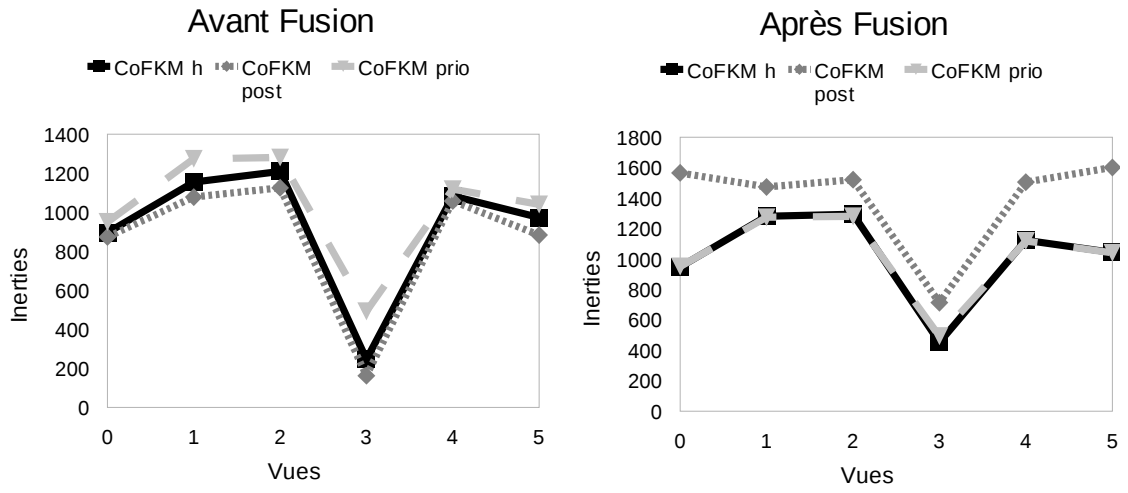

FIGURE 2.4 — Comparaisons des valeurs de critère interne dans chaque vue avant et après fusion (règle d'affectation) pour CoFKMet ses variantes *a priori* et *a posteriori* pour *multiple features*.

Nous constatons que CoFKM devient sensiblement équivalent à sa variante concaténée. Une autre façon de mesurer l'impact du désaccord sur le critère objectif de CoFKM est d'observer la proportion de ces deux valeurs. Cette mesure est faite dans le graphique fig.2.7.

2.5.4 Évaluation externe

L'évaluation externe vise à mesurer la performance de CoFKM par rapport à l'état de l'art dans l'objectif de retrouver une classification de référence. Les critères de mesure de performance sont ceux décrits dans la section 1.5.3 : la *F-mesure* ou *F-score* (évaluée grâce au *rappel* et à la *précision*), l'*entropie moyenne* ou *AvgEnt* et l'information mutuelle normalisée *NMI*.

Les différentes expériences réalisées visent à :

– confirmer l'intérêt d'utiliser toutes les vues des données afin d'améliorer la qualité du *clustering* produit ;
– insister sur l'importance de maintenir la recherche d'une solution réalisant un compromis des différentes solutions locales naturelles ;
– étudier l'impact des paramètres β et η sur la qualité du *clustering* produit ;
– observer l'apport de l'extension pour le traitement de données décrites par des matrices de similarités.

Intérêt de l'utilisation de toutes les descriptions.

Les premiers travaux autour du *clustering* de données multi-représentées visaient à démontrer l'apport de l'utilisation conjointe des différentes vues afin de garantir une meilleure qualité du *clustering* produit comparativement à l'utilisation d'une représentation unique. CoFKM a ainsi été éprouvé sur les jeux *mfeat* et *2D2K* et comparé à FKM appliqué séparément sur chacune des vues.

	% F-mesure			*AvgEnt*			*NMI*		
CoFKM	**92.01**	±	**0.00**	**0.29**	±	**0.00**	**0.91**	±	**0.00**
FKM-fac	66.69	±	3.89	0.98	±	0.09	0.70	±	0.03
FKM-fou	33.19	±	1.76	2.24	±	0.06	0.32	±	0.02
FKM-kar	23.04	±	1.19	2.97	±	0.09	0.11	±	0.03
FKM-mor	57.04	±	4.25	1.16	±	0.11	0.65	±	0.03
FKM-pix	70.41	±	2.93	0.88	±	0.06	0.74	±	0.02
FKM-zer	42.56	±	1.23	1.73	±	0.03	0.48	±	0.01
EM gmm(vs1)-fac	23.55	±	4.20	2.65	±	0.30	0.20	±	0.09
EM gmm(vs1)-fou	18.12	±	0.06	3.25	±	0.03	0.02	±	0.01
EM gmm(vs1)-kar	19.01	±	0.48	3.10	±	0.08	0.07	±	0.02
EM gmm(vs1)-mor	38.20	±	3.48	1.71	±	0.15	0.48	±	0.05
EM gmm(vs1)-pix	21.49	±	2.16	2.79	±	0.23	0.16	±	0.07
EM gmm(vs1)-zer	18.66	±	0.23	3.06	±	0.07	0.08	±	0.02
EM gmm(vs2)-fac	62.67	±	5.20	1.06	±	0.13	0.68	±	0.04
EM gmm(vs2)-fou	42.73	±	3.11	1.69	±	0.09	0.49	±	0.03
EM gmm(vs2)-kar	56.05	±	2.45	1.25	±	0.06	0.62	±	0.02
EM gmm(vs2)-mor	57.13	±	3.59	1.17	±	0.10	0.65	±	0.03
EM gmm(vs2)-pix	63.38	±	5.68	1.01	±	0.13	0.70	±	0.04
EM gmm(vs2)-zer	40.39	±	1.29	1.79	±	0.05	0.46	±	0.02
EM gmm(vs3)-fac	63.78	±	5.64	0.99	±	0.13	0.70	±	0.04
EM gmm(vs3)-fou	45.50	±	3.51	1.54	±	0.10	0.54	±	0.03
EM gmm(vs3)-kar	58.38	±	3.59	1.11	±	0.09	0.67	±	0.03
EM gmm(vs3)-mor	50.40	±	3.99	1.42	±	0.11	0.57	±	0.03
EM gmm(vs3)-pix	42.50	±	4.18	1.57	±	0.14	0.53	±	0.04
EM gmm(vs3)-zer	37.05	±	0.80	1.85	±	0.03	0.44	±	0.01

Tableau 2.1 — Évaluation externe sur *mfeat* de CoFKM comparé aux approches mono-vues. CoFKM surpasse les approches floues et probabilistes FKM et EM selon différents modèles parcimonieux, selon les 3 critères d'évaluation.

Les tableaux 2.1, 2.2 permettent d'observer le profit obtenu de l'utilisation conjointe de toutes les représentations. Pour les deux jeux de données, CoFKM surpasse assez nettement les

	% *F-mesure*			*AvgEnt*			*NMI*		
CoFKM	**94.18**	±	**0.00**	**0.18**	±	**0.00**	**0.82**	±	**0.00**
FKM-2d2kv1	85.32	±	5.88	0.40	±	0.19	0.60	±	0.19
FKM-2d2kv2	82.64	±	0.00	0.45	±	0.00	0.55	±	0.00
FKM-2d2kv3	85.32	±	5.88	0.40	±	0.19	0.60	±	0.19
EM gmm(vs1)-v1	79.19	±	8.53	0.50	±	0.21	0.50	±	0.21
EM gmm(vs1)-v2	79.74	±	4.26	0.50	±	0.07	0.50	±	0.07
EM gmm(vs1)-v3	79.19	±	8.53	0.50	±	0.21	0.50	±	0.21
EM gmm(vs2)-v1	85.12	±	5.82	0.40	±	0.19	0.60	±	0.19
EM gmm(vs2)-v2	82.64	±	0.00	0.45	±	0.00	0.55	±	0.00
EM gmm(vs2)-v3	85.12	±	5.82	0.40	±	0.19	0.60	±	0.19
EM gmm(vs3)-v1	82.80	±	6.41	0.44	±	0.20	0.56	±	0.20
EM gmm(vs3)-v2	82.46	±	1.54	0.46	±	0.02	0.54	±	0.02
EM gmm(vs3)-v3	82.80	±	6.41	0.44	±	0.20	0.56	±	0.20

Tableau 2.2 — Évaluation externe sur *2D2K* de CoFKM comparé aux approches mono-vues. CoFKM surpasse les approches floues et probabilistes FKM et EM selon différents modèles parcimonieux, selon les 3 critères d'évaluation.

approches floues et probabilistes FKM et EM quelque soit la représentation sur laquelle elles sont appliquées, et selon tous les critères d'évaluation.

Intérêt de la recherche d'un compromis.

L'intérêt principal de la contribution CoFKM est notamment de justifier le critère proposé comme une variante du critère de CoEM offrant des propriétés de convergence tout en maintenant la recherche d'un accord entre les vues (η ne décroît pas). De la même manière l'intérêt de ce maintien est observé au regard de CoFC qui dans son expression la plus simple fixe toutes les vues sauf une dans laquelle une solution réalisant un accord est recherchée. Les tableaux 2.3 et 2.4 permettent de mesurer les qualités respectives de ces approches.

CoFKM se comporte mieux sur le jeu de données *mfeat* où il surpasse les autres approches de l'état de l'art. En revanche les résultats sont bien plus ténus sur le jeu *2D2K* pour lequel une variante parcimonieuse de CoEM dans le cas d'un mélange de gaussiennes offre les meilleurs résultats. Les résultats de CoFC sont mauvais et tendent à produire des groupes déséquilibrés en taille, ce qui tend à augmenter le *Rappel* mais diminuer la *Précision*, de même que la *F-mesure*. En réalité cette dégénérescence est dû à l'imposition du paramètre de flou β fixé à 2 dans le critère objectif de CoFC.

Enfin, dans le but de justifier empiriquement la démarcation de la contribution proposée par rapport aux variantes de fusion *a priori* et *a posteriori*, l'approche à été évaluée comparativement à celles-ci. Les tableaux 2.5 et 2.6 reflètent l'apport de la recherche d'un *clustering* par une approche centralisée. CoFKM se comporte mieux sur *mfeat* que les variantes *a priori* (concat) et *a posteriori* déclinées identiquement de CoFKM et CoEM. Encore une fois les différences sur *2D2K* sont moins flagrantes et cette fois la fusion *a priori* est plus efficace. Toutefois l'objectif des approches centralisées n'est pas de surpasser les fusions *a priori*. Celle-ci n'est en effet pas possible lorsque l'on se place dans un contexte général de données distribuées et de traitement centralisés. Les informations de *clustering* (degrés d'appartenances et prototypes) sont moins

	% *F-mesure*			*AvgEnt*			*NMI*		
CoFKM	**92.01**	±	**0.00**	**0.29**	±	**0.00**	**0.91**	±	**0.00**
CoEM gmm(vs1)	39.81	±	5.34	1.61	±	0.13	0.52	±	0.04
CoEM gmm(vs2)	82.80	±	4.44	0.50	±	0.09	0.85	±	0.03
CoEM gmm(vs3)	74.96	±	5.42	0.72	±	0.12	0.78	±	0.04
CoFC-fac	51.73	±	5.03	1.34	±	0.16	0.60	±	0.05
CoFC-fou	55.88	±	4.85	1.23	±	0.13	0.63	±	0.04
CoFC-kar	56.13	±	4.91	1.23	±	0.14	0.63	±	0.04
CoFC-mor	59.74	±	5.72	1.17	±	0.15	0.65	±	0.05
CoFC-pix	52.56	±	5.11	1.32	±	0.16	0.60	±	0.05
CoFC-zer	56.61	±	4.79	1.19	±	0.14	0.64	±	0.04
CoFC Global-fac	30.77	±	0.08	2.47	±	0.01	0.26	±	0.00
CoFC Global-fou	31.00	±	0.07	2.45	±	0.01	0.26	±	0.00
CoFC Global-kar	31.00	±	0.05	2.45	±	0.01	0.26	±	0.00
CoFC Global-mor	31.22	±	0.03	2.45	±	0.00	0.26	±	0.00
CoFC Global-pix	30.81	±	0.05	2.46	±	0.01	0.26	±	0.00
CoFC Global-zer	30.58	±	0.03	2.43	±	0.00	0.25	±	0.00

Tableau 2.3 — Évaluation externe sur *mfeat* de CoFKM comparé aux approches centralisées multi-vues. CoFKM surpasse les approches CoEM et CoFC, selon les 3 critères d'évaluation.

	% *F-mesure*			*AvgEnt*			*NMI*		
CoFKM	94.18	±	0.00	0.18	±	0.00	0.82	±	0.00
CoEM gmm (vs1)	93.85	±	1.09	0.18	±	0.02	0.82	±	0.02
CoEM gmm (vs2)	**95.12**	±	**0.00**	**0.15**	±	**0.00**	**0.85**	±	**0.00**
CoEM gmm (vs3)	66.62	±	0.00	1.00	±	0.00	0.00	±	0.00
CoFC-v1	88.84	±	6.20	0.30	±	0.14	0.70	±	0.14
CoFC-v2	91.95	±	2.94	0.23	±	0.07	0.77	±	0.07
CoFC-v3	88.84	±	6.20	0.30	±	0.14	0.70	±	0.14
CoFC Global-v1	91.22	±	0.00	0.25	±	0.00	0.75	±	0.00
CoFC Global-v2	94.17	±	0.00	0.19	±	0.00	0.81	±	0.00
CoFC Global-v3	91.22	±	0.00	0.25	±	0.00	0.75	±	0.00

Tableau 2.4 — Évaluation externe sur *2D2K* de CoFKM comparé aux approches centralisées multi-vues. CoEM pour un modèle parcimonieux classique dépasse l'approche CoFKM, selon les 3 critères d'évaluation.

coûteuses à échanger et transférer que les descriptions des individus elles mêmes. De plus les informations de *clustering* offrent un résumé et ne dévoilent pas la nature d'un individu particulier, et ainsi respecte la confidentialité des données.

Impact des paramètres sur la qualité du *clustering*.

CoFKM nécessite, pour garantir une certaine flexibilité, de définir deux paramètres β et η représentant le degré de flou, ainsi que l'importance de l'accord souhaité. Des expériences ont permis de mesurer l'influence de chacun de ces paramètres et de justifier les heuristiques. Elles sont représentées dans les graphiques Fig. 2.5.

	% *F-mesure*			*AvgEnt*			*NMI*		
CoFKM	**92.01**	±	**0.00**	**0.29**	±	**0.00**	**0.91**	±	**0.00**
CoFKM post	55.72	±	4.28	1.21	±	0.12	0.64	±	0.04
CoEM gmm post(vs1)	27.46	±	9.01	2.53	±	0.54	0.24	±	0.16
CoEM gmm post(vs2)	57.20	±	5.22	1.18	±	0.14	0.65	±	0.04
CoEM gmm post(vs3)	45.64	±	5.21	1.54	±	0.15	0.54	±	0.05
FKM concat	90.42	±	3.44	0.33	±	0.07	0.90	±	0.02
EM concat(vs1)	32.51	±	6.68	1.77	±	0.25	0.47	±	0.08
EM concat(vs2)	77.90	±	5.72	0.56	±	0.12	0.83	±	0.04
EM concat(vs3)	60.10	±	5.53	1.04	±	0.14	0.69	±	0.04

TABLEAU 2.5 — Comparaison entre CoFKM, et les variantes *a priori* et *a posteriori* pour *multiple features*.

	% *F-mesure*			*AvgEnt*			*NMI*		
CoFKM	94.18	±	0.00	0.18	±	0.00	0.82	±	0.00
CoFKM post	86.28	±	13.27	0.34	±	0.27	0.66	±	0.27
CoEM gmm post(vs1)	80.43	±	14.21	0.45	±	0.29	0.55	±	0.29
CoEM gmm post(vs2)	86.60	±	14.69	0.32	±	0.29	0.68	±	0.29
CoEM gmm post(vs3)	85.47	±	13.36	0.36	±	0.27	0.64	±	0.27
FKM concat	**96.27**	±	**0.00**	**0.13**	±	**0.00**	**0.87**	±	**0.00**
EM concat(vs1)	93.18	±	8.22	0.19	±	0.15	0.81	±	0.15
EM concat(vs2)	**96.27**	±	**0.00**	**0.13**	±	**0.00**	**0.87**	±	**0.00**
EM concat(vs3)	96.07	±	0.00	0.14	±	0.00	0.86	±	0.00

TABLEAU 2.6 — Comparaison entre CoFKM, et les variantes *a priori* et *a posteriori* pour *2D2K*.

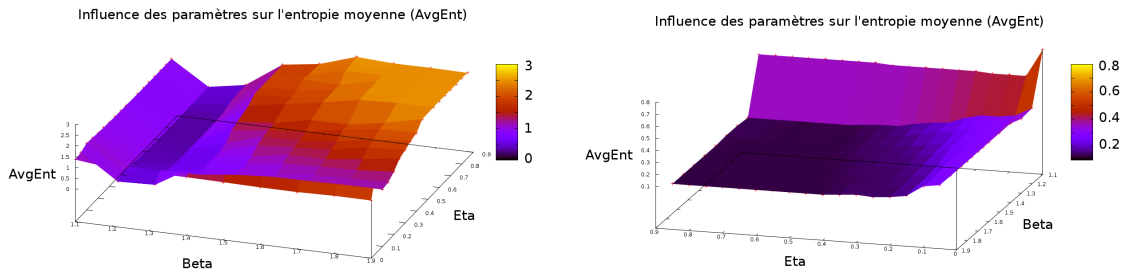

FIGURE 2.5 — Influence des paramètres η et β sur CoFKM pour *mfeat* (à gauche) et *2D2K* (à droite). Selon le jeu de donnée le paramétrage idéal n'est pas le même, ce qui conforte l'idée de proposer une approche plus flexible.

Pour *2D2K*, on peut choisir n'importe qu'elle valeur de β au delà de $\beta = 1.1$ et on peut observer que l'heuristique pour $\eta = \frac{n_r - 1}{2 \times n_r} = \frac{1}{3}$ donne de bons résultats. Pour *mfeat*, une valeur appropriée pour β devrait être proche de 1.2. La valeur $\beta = 2$ donne de très mauvais résultats pour CoFKM, ce qui confirme les résultats obtenus sur CoFC à valeur identique du paramètre

de flou. Le choix heuristique de $\eta = \frac{n_r - 1}{2 \times n_r} = \frac{5}{12}$ donne encore une fois des résultats corrects (Fig.2.6).

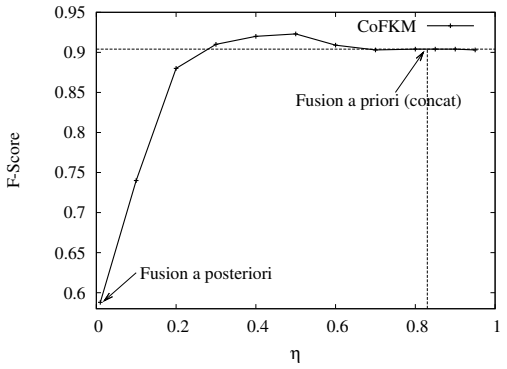

FIGURE 2.6 — CoFKM sur *mfeat* pour différentes valeurs de η. On remarque que l'heuristique de choix de η permet de dépasser la performance de la fusion *a priori*.

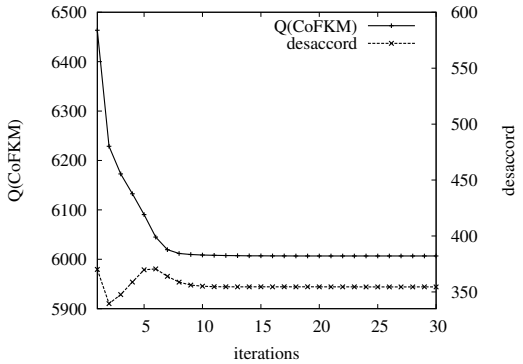

FIGURE 2.7 — Évolution du critère CoFKM sur *mfeat*.

Apports de la variante à noyaux

CoKFKM a été également étudié empiriquement sur une partie du jeu de données *WebKB* (les 100 premiers individus). Ce jeu est assez difficile à traiter, puisqu'il réunit un certain nombre de conditions néfastes pour les approches de classification usuelles :

- la dimensionnalité est très élevée comparée aux nombre d'individus (documents) disponibles ;
- dans la vue représentant le contenu des liens entrants, beaucoup d'individus n'ont pas de descriptions ;
- les tailles des classes sont déséquilibrées.

Le modèle CoFKM a été comparé avec CoEM pour un mélange de lois multinomiales, puis avec l'extension CoKFKM en choisissant comme matrices de similarité, la distance du cosinus entre les documents, considérée comme plus efficace sur les données textuelles que les produits scalaires classiques. Il s'agit en fait, de normaliser ces derniers par la taille des vecteurs documents correspondant. Soient x_i et x_j deux vecteurs de termes correspondant à des descriptions de deux documents, la matrice de similarité du cosinus K_{cos} entre x_i et x_j est définie par :

$$K^{(r)}_{cosij} = \frac{\langle x_i^{(r)}, x_j^{(r)} \rangle}{||x_i^{(r)}||_2 . ||x_j^{(r)}||_2}$$

Dans le cas où les vecteurs de documents sont centrés et réduits, il s'agit d'une reformulation dans le cadre de la *Recherche d'Information*, de la corrélation entre x_i et x_j.

Les algorithmes CoKFKM, CoFKM et CoEM ont été modifiés pour prendre en compte notamment les descriptions vides de la plupart des individus. En effet, lorsqu'un objet n'a pas de description dans une vue, on ne l'intègre pas dans la définition des centres (dans CoFKM), ou par le calcul des distances aux centres (dans CoKFKM).

De plus, et afin d'obtenir une version moins coûteuse en temps que l'approche CoKFKM, une version accélérée de CoKFKM est proposée. Elle correspond à une variante dans laquelle

la définition des distances aux centres (2.39) est réexprimée, de sorte à ne pas tenir compte de tous les individus, mais seulement d'un pourcentage prédéfini parmi les plus proches. Dans l'esprit, ce principe tend à faire comporter CoKFKM comme une variante moins floue et plus proche d'une extension multi-vues de KM. Soit $q_{\%}$ le pourcentage prédéfini, on peut associer à chaque centre c_k l'ensemble $\mathcal{N}(c_k)$ des $q = q_{\%}n$ individus ayant les degrés d'appartenance au groupe C_k les plus élevés. Ainsi, étant donnés x_i, c_k et r, si l'on veut calculer $d^2_{(r)}(x_i, {c_k}^*)$, nous ne considérerons que les $q_{\%} = \frac{n}{n_k}$ individus $x_i \in \mathcal{N}(c_k)$ qui sont les plus représentatifs du groupe C_k :

$$d^2_{(r)}(x_i, {c_k}^*) = K^{(r)}_{ii} - 2\frac{\sum\limits_{x_j \in \mathcal{N}(c_k)} {u^{(r)}_{jk\eta}}^{\beta} K^{(r)}_{ij}}{\sum\limits_{x_j \in \mathcal{N}(c_k)} {u^{(r)}_{jk\eta}}^{\beta}} + \frac{\sum\limits_{x_j \in \mathcal{N}(c_k)} \sum\limits_{x_l \in \mathcal{N}(c_k)} {u^{(r)}_{jk\eta}}^{\beta} {u^{(r)}_{lk\eta}}^{\beta} K^{(r)}_{jl}}{\left(\sum\limits_{x_j \in \mathcal{N}(c_k)} {u^{(r)}_{jk\eta}}^{\beta}\right)^2}$$

Le choix heuristique $q_{\%} = \frac{n}{n_k}$ correspond à l'hypothèse d'homogénéité de la taille des groupes.

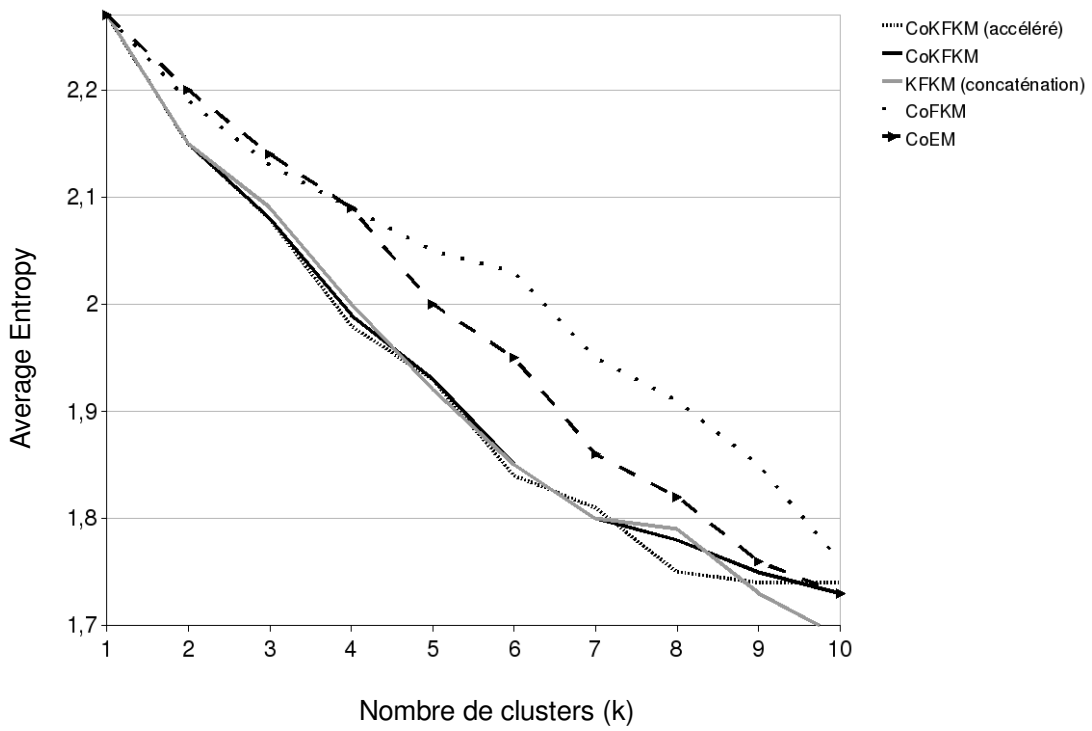

FIGURE 2.8 — Tests comparatifs entre CoKFKM, CoFKM et CoEM.

La figure 2.8 montre l'évolution de l'entropie moyenne en fonction du nombre de groupes. CoEM se comporte mieux que CoFKM, mais l'apport le plus significatif concerne l'utilisation de matrices noyaux cosinus, ce qui n'est pas gérable tel quel par CoEM. Les résultats obtenus par CoKFKM sont sensiblement équivalents à ceux obtenus par concaténation avec une approche FKM à noyau classique. Enfin, l'accélération est une heuristique prometteuse et elle laisse entrevoir des perspectives sur l'extension à noyaux.

2.6 Discussion

Les contributions CoFKM et CoKFKM réalisent un traitement centralisé de données multi-vues, ou multi-sources éventuellement décentralisées. Elles s'inscrivent complètement dans le paradigme des approches discriminatives vues comme des problèmes d'optimisation d'un critère objectif pénalisé : le compromis entre la recherche de *clusterings* locaux dans chaque vue et la recherche d'un accord. CoFKM permet de concilier les approches floues qui se retrouvent régulièrement parmi les méthodes de *clustering* centralisées, avec les approches probabilistes de type CoEM en offrant de bonnes propriétés de convergence quelque soit l'importance de la recherche d'une solution consensus. L'algorithme proposé est simple, intuitif, facilement implémentable et parallélisable. Il est flexible de par son paramétrage mais reste contrôlable par le nombre réduit de ces paramètres. Il est facilement extensible par sa variante à noyaux et permet de prendre en compte des données multi-vues où celles-ci sont décrites, soit par des représentations vectorielles, soit par des tableaux relationnels.

Malgré les avantages, CoFKM et CoKFKM sont limités sur plusieurs aspects. Tout d'abord, à l'image de CoEM, le nombre de groupes doit être donné et identique dans toutes les vues, ce qui est extrêmement restrictif. Il est en général admis que le nombre naturel de groupes dans chaque vue soit différent. Cependant, dans le contexte où l'on cherche un *clustering* unique des individus, cet argument négatif semble ne plus tenir. Un autre inconvénient concerne la recherche des *clusterings* locaux. Celle-ci est réalisée uniquement selon l'objectif de FKM. Cette imposition restreint encore une fois l'approche car elle ne permet pas de prendre en compte la recherche d'un *clustering* local adapté dans le cas où les individus sont distribués selon des formes arbitraires, et non nécessairement des classes convexes et bien séparées. Ceci est gérable, mais difficile à contrôler, par l'utilisation de matrices de proximité adaptées dans chaque vue et l'utilisation de CoKFKM. La difficulté de découvrir des groupes de formes arbitraires dans l'espace de description d'origine est alors reportée sur la construction de matrices de similarité adaptées capables de suggérer un nouvel espace dans lequel les groupes seraient compactes et bien séparés, à l'image du Laplacien normalisé de SC (cf. section 1.3.1.2).

2.7 Conclusion

Ce chapitre a permis de présenter la problématique du *clustering* multi-vues. L'étude a été centrée sur les approches dites centralisées, et les différentes alternatives proposées dans la littérature ont été dressées. Les contributions proposées prennent leurs racines dans quelques-unes de ces approches, CoFC et CoEM, afin de les étendre et de tirer parti du meilleur de chacune. L'approche CoFKM définie présente de bonnes propriétés puisqu'elle généralise différentes solutions de fusion, permet de lui associer une solution algorithmique efficace et convergente, se compose de peu de paramètres et est donc moins sensible à ce paramétrage. L'extension CoKFKM permet de traiter les cas où les données sont décrites par plusieurs matrices de similarité et est ainsi beaucoup plus flexible pour gérer des cas concrets d'applications. Les résultats empiriques développés valident les contributions et viennent confirmer l'apport de celles-ci comparé aux approches existantes.

Les divers inconvénients relevés notamment lors de rencontres avec des spécialistes de la communauté *fouille de données* ont permis de réfléchir à d'autres techniques de classification non supervisée, réalisant un minimum d'hypothèses sur la forme de la distribution des individus dans chaque représentation (ou le critère objectif local correspondant) ou le nombre de groupes local le plus adapté. L'idée est de proposer un traitement séquentiel sur l'ensemble des représentations de sorte que pour chaque représentation, la recherche d'un *clustering* soit guidée par les derniers résultats émanant des autres vues et considérés comme autant de superviseurs.

L'approche envisagée se fonde alors sur des éléments d'apprentissage semi-supervisé dont il est question dans le prochain chapitre.

Classification non supervisée et intégration de connaissances

3

Sommaire

3.1 Introduction **90**

3.2 Contexte **90**

3.3 Approches par satisfaction des contraintes **92**

3.3.1 COP-KMEANS : les K-moyennes sous contraintes 92

3.3.2 CCHC : *clustering* semi-supervisé hiérarchique en lien complet 94

3.3.3 SSEM : estimation d'un mélange de modèle semi-supervisé 95

3.4 Approches par objectif pénalisé **98**

3.4.1 PCKM : les K-moyennes contraintes pénalisées 98

3.4.2 SSKM : les K-moyennes semi-supervisées 100

3.5 Approches par altération de la proximité **101**

3.5.1 LLMA : adaptation localement linéaire de la métrique 101

3.6 Approches indépendantes de l'algorithme de *clustering* **104**

3.6.1 BC : *BoostCluster* 104

3.7 Contributions **106**

3.7.1 Motivation 106

3.7.2 BOC : *boosting* de *clustering* 109

3.7.3 UZABOC et ADAUZABOC : *boosting* simple et adaptatif de *clustering* par optimisation 117

3.8 Évaluation **123**

3.8.1 Données 123

3.8.2 Protocole expérimental 124

3.8.3 Évaluation interne 126

3.8.4 Évaluation externe 134

3.9 Discussion **142**

3.10 Conclusion **143**

3.1 Introduction

Ce chapitre présente les contributions apportées au *clustering* semi-supervisé, les approches BoC et UzaBoC. Ces propositions ont été publiées dans la communauté internationale de fouille de données et la communauté francophone de classification [Sublemontier et al., 2011c], [Sublemontier et al., 2011b]. Le contexte scientifique et la problématique seront rappelés. Seront développées également une famille d'algorithmes de *clustering* semi-supervisés ainsi que quelques approches d'apprentissage de distances apportant des solutions au problème. Il sera précisé à chaque fois, à l'image des chapitres précédents, le type d'approche (algoithmique pure, discriminative ou générative). Ensuite sera détaillée une approche particulière de l'état de l'art, concernant les approches dites indépendantes de l'algorithme de *clustering*. Pour finir, les études empiriques réalisées valideront les contributions et quelques perspectives d'amélioration seront discutées.

L'objectif des approches de *clustering* semi-supervisées est de produire une structure permettant d'organiser les données tout en satisfaisant des contraintes fournies pour certaines paires d'individus à regrouper ensemble ou non. La notation choisie pour refléter au mieux les différentes approches proposées est la suivante :

NOTATION	
n :	le nombre d'individus à regrouper.
n_p :	le nombre d'attributs décrivant les individus.
n_k :	le nombre de groupes à identifier.
n_c :	le nombre de classes associé aux données.
$\mathcal{X} = \{x_1, ..., x_n\}$:	l'ensemble des n individus à partitionner.
$X \in \mathbb{R}^{n \times n_p}$:	la représentation matricielle de $\mathcal{X}$.
$x_i \in \mathbb{R}^{n_p}$:	la représentation vectorielle de l'individu x_i.
$C = \{C_1, ..., C_{n_k}\}$:	la structure de *clustering* en n_k groupes à construire.
$c = \{c_1, ..., c_{n_k}\}$:	l'ensemble des n_k prototypes des groupes.
$\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{n_c}\}$:	l'ensemble des n_c classes d'individus à retrouver.
$\mathcal{D} = \{\mathcal{D}_0, ..., \mathcal{D}_n\}$:	la structure de dendrogramme associée aux données.
$d(x_i, x_j)$:	la distance au sens général entre deux individus x_i et x_j.
$d_P(x_i, x_j)$:	la distance entre x_i et x_j dans un sous-espace P.
$\|\|x_i - x_j\|\|_p$:	la distance de Minkowski entre deux individus x_i et x_j.
$\mathcal{ML}$:	l'ensemble des $(x_i, x_j) \in \mathcal{X}^2$ devant être regroupés.
$\mathcal{CL}$:	les $(x_i, x_j) \in \mathcal{X}^2$ devant être séparés.
m :	le nombre de contraintes $\mathcal{ML}$ et $\mathcal{CL}$.
m^+ :	le nombre de contraintes $\mathcal{ML}$.
m^- :	le nombre de contraintes $\mathcal{CL}$.
A :	l'algorithme de *clustering* employé pour obtenir C.
$Link(x_i, x_j, A)$:	x_i et x_j sont regroupés par A.
$\overline{Link}(x_i, x_j, A)$:	x_i et x_j sont séparés par A.
$H \in \{0, 1\}^{n \times n}$:	la matrice de *clustering* associée à C

3.2 Contexte

La problématique du *clustering* semi-supervisé [Davidson and Basu, 2007] correspond à la recherche d'un *clustering* des individus, par un algorithme de *clustering* A, devant respecter un

ensemble de connaissances de classification sur certaines paires d'individus. Ces connaissances prennent la forme de contraintes notées $\mathcal{ML}$ et $\mathcal{CL}$ telles que :

- deux individus x_i et x_j liés par une contrainte $\mathcal{ML}$ (*must-link*) doivent être regroupés par A, plus formellement :
$$(x_i, x_j) \in \mathcal{ML} \Rightarrow Link(x_i, x_j, A)$$
- deux individus x_i et x_j liés par une contrainte $\mathcal{CL}$ (*cannot-link*) doivent être séparés par A, plus formellement :
$$(x_i, x_j) \in \mathcal{CL} \Rightarrow \overline{Link}(x_i, x_j, A)$$

On parle alors également de *clustering* contraint. Les contraintes peuvent être :

- données par l'utilisateur pour guider la recherche d'une solution particulière respectant des résultats obtenus par d'autres moyens (expérience, *etc.*) ;
- extraites à partir de sources d'information externes pouvant provenir d'autres vues des données à traiter.

Ce problème, issu plutôt des applications, à néanmoins donné lieu à beaucoup d'études théoriques et de propositions d'algorithmes. Il a notamment donné naissance au problème de l'intégration de connaissances externes pour la recherche d'un *clustering* de meilleure qualité, légèrement différent du problème d'origine dans la mesure où les contraintes données sont vues comme un moyen d'améliorer la performance des algorithmes de *clustering*.

Historiquement, les premières approches se sont focalisées sur le respect absolu, au sens de la satisfaction logique, de ces contraintes par un algorithme de *clustering* A prédéfini. Ces travaux remontent à l'aube des années 2000 avec la thèse de Kiri Wagstaff alors à l'université de Cornell, NY, qui fût un des précurseurs de ce champ de recherche. L'idée était de modifier le coeur des algorithmes de *clustering* (COBWEBet KM) de telle sorte que les groupes formés ne devaient violer aucune contraintes [Wagstaff and Cardie, 2000] ; [Wagstaff et al., 2001]. Les travaux menés notamment par l'équipe de Ian Davidson à Albany, NY, concernant ce type d'intégration de contraintes, ont vite montré leurs limites au niveau computationnel ainsi qu'au niveau de la satisfiabilité [Davidson and Ravi, 2005a] ; [Davidson and Ravi, 2005b]. Parallèlement à ces études, d'autres équipes de recherche, notamment Dan Klein à Stanford ont suggéré qu'une autre voie pour satisfaire les contraintes données était d'altérer la mesure de proximité disponible ou dérivée des données afin de s'assurer qu'un algorithme bien choisi réussirait à respecter les contraintes [Klein et al., 2002]. Ces travaux intègrent notamment un second principe important dans la thématique de recherche, qui est l'induction de nouvelles contraintes à partir des premières. Cela permet d'accroître l'efficacité des approches de *clustering* contraint tout en conservant une faible quantité de contraintes, possiblement coûteuses, à fournir. La transformation de la représentation d'origine des individus ou de manière quasi-équivalente, de la mesure de proximité associée aux individus va devenir le socle de nombreuses approches censées répondre à la problématique.

L'idée de satisfaire au mieux les contraintes deviendra centrale par la suite, et d'autant plus que l'on considérera une certaine forme d'incertitude associées aux contraintes que l'on estimera désormais devoir satisfaire au mieux. Dans ce nouveau contexte de quasi-satisfaction des contraintes, les travaux ont consisté, pour les algorithmes basés sur l'optimisation d'une fonction objectif, à modifier le critère de sorte que des contraintes non satisfaites conduisent à une pénalisation de celui-ci, comme proposé par Sugato Basu [Basu et al., 2004]. Ils ont ensuite été améliorés dans le but de transformer cette forme de pénalisation de critère, en altération de la mesure de proximité entre les individus comme l'a proposé Kulis [Kulis et al., 2005]. Pour dresser un premier bilan de ces approches, nous constatons que l'intégralité d'entre elles nécessitait d'imposer le critère objectif et/ou l'algorithme de *clustering* lui-même.

Une autre famille d'approches plus indépendantes vis à vis de l'algorithme de *clustering* utilisé, a consisté à considérer le problème de l'intégration des contraintes comme un problème d'apprentissage de proximité (distance et ou similarité), ou de nouvel représentation des individus dans laquelle des objets devant être regroupés (resp. séparés) doivent être proches (resp. éloignés) dans la nouvelle représentation. Une fois la nouvelle proximité induite, n'importe quel algorithme de *clustering* peut être appliqué sous réserve de correspondance entre le type de proximité apprise et le type de proximité sur lequel se fonde l'algorithme (une distance euclidienne pour KM) [Xing et al., 2002a] ; [Zhang et al., 2003]. L'issu de ces travaux est que l'apprentissage de cette nouvelle représentation n'est pas du tout remis en cause par les résultats observés sur l'algorithme de *clustering* employé. En d'autres termes, finalement, un contrôle de l'impact de la nouvelle représentation sur le *clustering* produit n'est pas possible.

Parmi les travaux les plus récents censés répondre à cette nouvelle problématique d'une intégration contrôlée de contraintes pour améliorer effectivement n'importe quel algorithme de *clustering*, nous nous sommes intéressé à BOOSTCLUSTER, proposé par Liu [Liu et al., 2007]. Ce type d'approche permet de construire de manière incrémentale un ensemble d'hypothèses de *clustering*. Les différentes familles d'approches permettant d'intégrer des connaissances externes sont représentées dans les schémas Fig. 3.1.

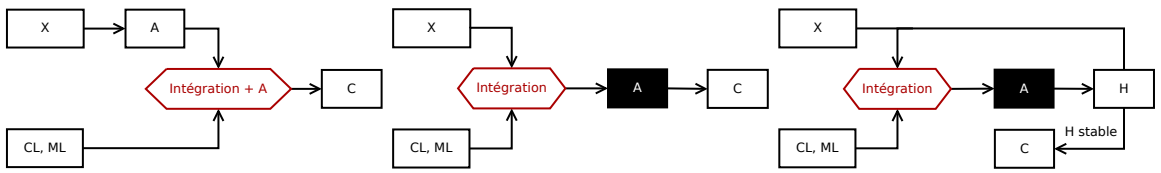

FIGURE 3.1 — Les différents types d'intégration dans le *clustering* semi-supervisé. Dans l'ordre, ci-dessus, l'intégration de contraintes dans l'algorithme A prédéfini, l'intégration de contraintes dans la définition de la proximité, avant l'application de l'algorithme A quelconque et enfin l'intégration contrôlée par l'algorithme de *clustering* quelconque A.

La contribution de ce chapitre correspond à des alternatives à cette approche, selon différents paradigmes de résolution. La première contribution BOC se fonde sur le principe du *boosting* de manière semblable à BOOSTCLUSTER. La seconde contribution UZABOC utilise des éléments d'optimisation numérique. Le chapitre est organisé comme suit : après avoir détaillé plus formellement les approches clés du développement autour du *clustering* semi-supervisé ou *clustering* contraint, citées précédemment de manière introductive, je présenterai les concepts apportés par BOOSTCLUSTER puis les concepts que nous proposons ainsi que les différentes approches. Nous conclurons sur notre étude de la problématique après avoir réalisé une étude empirique de l'approche et dressé quelques perspectives.

3.3 Approches par satisfaction des contraintes

3.3.1 COP-KMEANS : les K-moyennes sous contraintes

L'approche COP-KMEANS [Wagstaff et al., 2001] est parmi les premières approches de *clustering* semi-supervisé. Il s'agit d'une approche discriminative basée sur l'algorithme KM (1.3.1.1).

Objectif

L'objectif est de déterminer les prototypes optimaux d'un ensemble de n_k groupes de telle sorte que les groupes ainsi constitués ne violent aucune contrainte. On peut formaliser cet objectif sous la forme d'un problème d'optimisation sous contraintes de la manière suivante :

$$\begin{array}{rrcll} \underset{c,\,C}{min}\, Q_{\text{KM}}(c,C) = & \underset{c,C}{min} \sum_{k=1}^{n_k} \sum_{x_i \in C_k} ||x_i - c_k||_2^2 & & & \\ s.t. & C_k^2 \cap \mathcal{CL} & = & \emptyset & \forall C_k \in C \\ & \bigcup_{1 \leq k \leq n_k} (C_k^2 \cap \mathcal{ML}) & = & \mathcal{ML} & \end{array} \tag{3.1}$$

L'espace des solutions associé à ce problème d'optimisation est alors réduit pour ne contenir que les solutions satisfaisant effectivement les contraintes $\mathcal{ML}$ et $\mathcal{CL}$ données.

Algorithme

Le problème étant trop difficile à résoudre analytiquement, les auteurs proposent alors une approche purement algorithmique (algorithme 17) pour le résoudre. Ainsi, à l'image de KM, l'algorithme alterne une mise à jour des groupes et des prototypes de groupes selon le principe de résolution d'un système d'équation par une méthode itérative en partant d'une initialisation prédéfinie des prototypes de groupe. L'initialisation est aléatoire dans le but d'avoir plus de chance d'atteindre l'optimum global si il existe, après plusieurs exécutions de l'algorithme. La mise à jour (ou construction) des groupes C_k^* est différente de la règle classique de KM puisqu'elle est conditionnée par le respect de toutes les contraintes. Pour ce faire, les auteurs proposent un algorithme heuristique. L'idée est de parcourir dans l'ordre l'ensemble $\mathcal{X}$ et d'affecter chaque individu x_i au groupe le plus proche tel qu'aucune contrainte ne soit violée. Cette affectation peut se formaliser par la règle :

$$\begin{aligned} &\forall x_i \in \mathcal{X}\ \forall (x_i, x_j) \in \mathcal{ML}\ \forall (x_i, x_{j'}) \in \mathcal{CL}, \\ &\quad \exists C_k^* \left((c_k = \underset{c \in \{c_1,\ldots,c_{n_k}\}}{arg\ min} \sum_{k=1}^{n_k} \sum_{x_i \in C_k^*} ||x_i - c||_2^2) \wedge x_j \in C_k^* \wedge x_{j'} \notin C_k^* \right) \\ &\quad \Rightarrow C_k^* = C_k^* \cup \{x_i\} \end{aligned} \tag{3.2}$$

Notons qu'il est possible de ne pouvoir affecter x_i à aucun groupe, si notamment pour tous les groupes il existe un individu x_j dans ceux-ci tel que $(x_i, x_j) \in \mathcal{CL}$. De plus il s'agit d'une règle heuristique qui rend la recherche de la solution optimale gloutonne, dans le sens où l'obtention de la solution optimale est dépendante de l'ordre de parcours des individus lors de la construction des groupes. La règle de mise à jour des prototypes de groupes, elle, est la même que celle de KM, *i.e.* c_k^* est le centre de gravité du groupe C_k^* :

$$c_k^* = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

Discussion

Le premier problème qui n'en est réellement un que selon le cadre applicatif, est qu'il peut ne pas exister de solution. Dans un contexte applicatif où l'utilisateur veut obtenir un *clustering* des individus satisfaisant les contraintes, l'approche est limitée, car si l'ensemble des contraintes forme une théorie inconsistante, alors il n'existe par définition aucun moyen de les satisfaire toutes simultanément et l'espace des solutions associé au problème d'optimisation est vide.

Algorithme 17 Cop K-moyennes

ENTRÉES : $\mathcal{X}$, n_k, $\mathcal{ML}$, $\mathcal{CL}$
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation aléatoire des n_k centres de groupes $\{c_1, ..., c_{n_k}\}$
2 : Mise à jour des groupes C_k $\forall k \in [1..n_k]$ en utilisant la règle d'affectation (3.2)
3 : Mise à jour des centres de groupe c_k $\forall k \in [1..n_k]$ en utilisant (3.3)
4 : Si Q_{KM} change alors aller en **2**

Ainsi, il est préférable de pouvoir relâcher quelques contraintes pour être sûr de pouvoir fournir un *clustering* à l'utilisateur, mais cette tâche est difficile au sens de la complexité, puisque le problème de satisfiabilité de l'ensemble des contraintes est à lui seul NP-complet. Le second problème est que l'algorithme est dépendant de l'ordre de parcours des individus lors de l'étape de construction des groupes ce qui rend l'application de l'algorithme moins bien contrôlé et atténue les garanties sur l'obtention de l'optimum (toujours local).

3.3.2 CCHC : *clustering* semi-supervisé hiérarchique en lien complet

Une autre approche purement algorithmique a été développée par [Klein et al., 2002]. Elle vise a tirer parti d'un faible ensemble de contraintes, dans le but d'induire un plus grand ensemble de contraintes favorisant l'amélioration de la qualité d'un *clustering*. L'algorithme s'appuie sur le postulat qu'un individu x_i proche d'un autre individu x_j impliqué dans une contrainte $(x_j, x_l) \in \mathcal{ML}$ (resp. $(x_j, x_l) \in \mathcal{CL}$) doit être impliqué dans le même type de contrainte $(x_i, x_l) \in \mathcal{ML}$ (resp. $(x_i, x_l) \in \mathcal{CL}$), plus formellement :

$$\begin{aligned} \forall (x_i, x_j) \in \mathcal{X}^2,\ (x_i \textit{ proche de } x_j \wedge (x_j, x_l) \in \mathcal{ML}) &\Rightarrow (x_i, x_l) \in \mathcal{ML} \\ \forall (x_i, x_j) \in \mathcal{X}^2,\ (x_i \textit{ proche de } x_j \wedge (x_j, x_l) \in \mathcal{CL}) &\Rightarrow (x_i, x_l) \in \mathcal{CL} \end{aligned}$$

Algorithme

Pour réaliser effectivement l'idée du postulat, les auteurs proposent de réaliser implicitement une projection non linéaire des individus de $\mathcal{X}$ dans un certain espace non défini. Partant de $\mathcal{X} \subset \mathbb{R}^p$ et d'une mesure de distance sur $\mathbb{R}^p$, les auteurs proposent une gestion séparée des contraintes de type $\mathcal{ML}$ et des contraintes de type $\mathcal{CL}$. Pour les contraintes $\mathcal{ML}$, les auteurs proposent d'imposer directement une valeur de distance nulle entre les individus impliqués dans une de ces contraintes, ainsi :

$$\forall (x_i, x_j) \in \mathcal{X}^2,\ (x_i, x_j) \in \mathcal{ML} \Rightarrow d(x_i, x_j) = 0 \tag{3.3}$$

L'étape d'induction de nouvelles contraintes est réalisée en appliquant un algorithme de plus court chemin entre toutes les paires d'individus, dans le but de rétablir pour d les propriétés d'une métrique pour $\mathbb{R}^p$. Par ce choix d'intégration de contraintes $\mathcal{ML}$, on espère que tout algorithme de *clustering* les satisfasse normalement. La gestion des contraintes $\mathcal{CL}$ est quant à elle réalisée, dans un premier temps, en imposant une valeur maximum de distance entre les individus impliqués dans de telles contraintes :

$$\forall (x_i, x_j) \in \mathcal{X}^2,\ (x_i, x_j) \in \mathcal{CL} \Rightarrow d(x_i, x_j) = \max_{(x_i, x_j) \in \mathcal{X}^2} d(x_i, x_j) + 1 \tag{3.4}$$

Ce type d'intégration ne garanti pas qu'un algorithme de *clustering* satisfasse exactement les contraintes $\mathcal{CL}$. Les auteurs proposent dans ce cas de choisir un algorithme de *clustering*

particulier pour respecter l'ensemble des contraintes : l'algorithme de *clustering* hiérarchique par lien complet CLINK (section 1.2.2 du chapitre 2). Ainsi, si deux amas (groupes) A_1 et A_2 contiennent respectivement deux individus x_1 et x_2 impliqués dans une même contrainte $\mathcal{CL}$, alors la distance entre A_1 et A_2 est la plus élevée et les amas ne sont pas fusionnés par CLINK (2). Les individus impliqués dans une contrainte $\mathcal{ML}$ sont quant à eux regroupés dès la base du dendrogramme.

Algorithme 18 CCHC

ENTRÉES : $\mathcal{X}$, $d(.,.)$, n_k, $\mathcal{ML}$, $\mathcal{CL}$
SORTIES : $\mathcal{D}$
1 : Intégrer les contraintes $\mathcal{ML}$ par (3.3)
2 : Appliquer l'algorithme du plus court chemin $\forall (x_i, x_j) \in \mathcal{X}^2$
3 : Intégrer les contraintes $\mathcal{CL}$ par (3.4)
4 : Construire $\mathcal{D}$ par CLINK

Discussion

L'approche CCHC s'avère extrêmement efficace pour satisfaire absolument les contraintes données et induire de bonnes contraintes lorsque le postulat de départ est vérifié. Cependant la mise en œuvre par altération de la proximité est trop brutale, et la description des individus perd son sens, ou au moins aucun lien n'est fait *a posteriori* entre la nouvelle distance apprise et la description des individus lorsqu'elle existe (importance de certains descripteurs relativement aux autres). De plus, il peut arriver à l'image de COP-KMEANS qu'il n'existe pas de solutions satisfaisant les contraintes. Les cas extrêmes sont rares, mais ils existent notamment :

- si tous les individus de $\mathcal{X}$ sont impliqués dans l'ensemble des contraintes $\mathcal{ML}$, alors un *clustering* de $n_k \geq 2$ groupes violera au moins une de ces contraintes.
- si l'ensemble des contraintes $\mathcal{CL}$ contient une clique de taille c, alors un *clustering* de $n_k < c$ groupes violera au moins une de ces contraintes.

Finalement, les auteurs s'attachent à préserver la caractérisation de la proximité apprise d qui doit être une métrique. Ceci est validé par l'application de l'algorithme de plus court chemin sur toutes les paires d'individus. En revanche lors de l'intégration des contraintes $\mathcal{CL}$, cette caractérisation est perdue. En effet, si on dispose de $(x_1, x_2, x_3, x_4) \in \mathcal{X}^3$ tels que $(x_1, x_2) \in \mathcal{ML}$, $(x_2, x_3) \in \mathcal{ML}$, $(x_1, x_3) \in \mathcal{CL}$ et x_4 n'est impliqué dans aucune contrainte, alors on a :

$$\begin{aligned} d(x_1, x_3) &= \max_{(x_i,x_j)\in\mathcal{X}^2} d(x_i, x_j) + 1 = D \\ d(x_1, x_2) &= d(x_2, x_3) = 0 \end{aligned}$$

et ainsi $D = d(x_1, x_3) > d(x_1, x_2) + d(x_2, x_3) = 0$ ce qui contredit l'inégalité triangulaire (cf. section 1.5.4).

3.3.3 SSEM : estimation d'un mélange de modèle semi-supervisé

Le *clustering* par estimation de paramètre d'un modèle de mélange gaussien a également été étendu au *clustering* semi-supervisé par [Shental et al., 2003]. Dans cette approche, les auteurs proposent d'intégrer les deux types de contraintes $\mathcal{ML}$ et $\mathcal{CL}$ a travers la définition d'un modèle adapté étendant le modèle de mélange simple.

Modèle

Pour rappel, le modèle de mélange (cf. section 1.4.2) est défini par :

$$f(X_i;\Theta) = \sum_{k=1}^{n_k} \alpha_k f_k(x_i;\theta_k)$$

où les α_k et les $f_k(x_i;\theta_k)$ correspondent respectivement aux valeurs de probabilité *a priori* de la sélection de la k-ième composante et à la fonction de densité gaussienne correspondant à la variable X_i paramétrée par $\Theta_k = (c_k, \Sigma_k)$.

Les auteurs proposent de reprendre l'expression du modèle d'une part pour intégrer les contraintes $\mathcal{ML}$. Ainsi ils redéfinissent l'échantillon $\mathcal{X}$ comme l'union de sous-ensembles disjoints appelés *chunklets* :

$$\mathcal{X} = \bigcup_{l=1}^{n_l} \mathcal{X}_l$$

où chaque *chunklet* $\mathcal{X}_l$ correspond à un ensemble d'individus devant partager la même étiquette l, et par extension, liés par une contrainte $\mathcal{ML}$. n_l désigne le nombre naturel de *chunklets* défini par les contraintes $\mathcal{ML}$ ou par l'absence de contraintes. Ainsi, les individus non impliqués dans une contrainte $\mathcal{ML}$ définissent à eux seuls un *chunklet*. Dans ce contexte, les *chunklets* sont complétés par un vecteur aléatoire Z_l indiquant pour chaque individu x_i d'un *chunklet* $\mathcal{X}_l$ le groupe auquel il semble appartenir et les données $\mathcal{X}$ sont complétées par $Z = (Z_1, \ldots, Z_{n_l})$. Pour l'intégration des contraintes $\mathcal{CL}$ les auteurs remarquent que l'hypothèse d'une distribution *i.i.d* des variables cachées Z_l correspondantes aux *chunklets* est violée car il faut maintenir le fait que deux individus x_i et x_j, appartenant respectivement aux *chunklets* $\mathcal{X}_{l_1}$ et $\mathcal{X}_{l_2}$ et tels que $(x_i, x_j) \in \mathcal{CL}$ entraîne que les réalisations des variables cachées Z_{l_1} et Z_{l_2} doivent être différentes :

$$\forall (x_i, x_j) \in \mathcal{X}_{l_1} \times \mathcal{X}_{l_2}\ (x_i, x_j) \in \mathcal{CL} \Rightarrow z_{l_1} \neq z_{l_2}$$

Cette condition peut être réalisée en introduisant une dépendance entre les variables cachées Z_l. Le modèle de mélange gaussien, après introduction des *chunklets* peut alors être étendu en un réseau de markov définit par :

- les sommets qui sont soit les variables observées $X_i = x_i$ correspondant aux individus soit les variables cachées Z_l indiquant l'étiquette des individus du *chunklet* $\mathcal{X}_l$ correspondant ;
- les arêtes connectant chaque variable cachée Z_l à un individu x_i du *chunklet* que celle-ci représente sont caractérisées par leur fonction potentiel $f(x_i|Z_l = z_l;\Theta)$ avec $e(x_i) = z_l$ où $e : \mathcal{X} \mapsto \{1..n_l\}$ donne l'identifiant de l'étiquette de x_i. Un tel identifiant peut être obtenu à partir des contraintes $\mathcal{ML}$ et $\mathcal{CL}$ de départ ;
- les arêtes connectant les variables cachées Z_{l_1} et Z_{l_2} entre elles sont caractérisées par leur fonction potentiel $1 - \delta_{z_{l_1}, z_{l_2}}$ où δ est le symbole de Kronecker. Ainsi la valeur de cette fonction est binaire et maximale lorsque toute paire d'individus tirés parmi deux *chunklets* liés et différents, ont une étiquette de groupe différente :

$$\forall (x_i, x_j) \in \mathcal{X}_{l_1} \times \mathcal{X}_{l_2},\ \big(l_1 \neq l_2 \Rightarrow e(x_i) \neq e(x_j)\big) \Rightarrow \delta(z_{l_1}, z_{l_2}) = 0$$

Un tel modèle graphique est représenté en figure 3.2.

Objectif

Le critère objectif à optimiser correspond toujours à la vraisemblance des données $\mathcal{X}$ complétée par Z sous l'hypothèse d'existence des *chunklets*. Soit E_s l'évènement : « Z se conforme aux

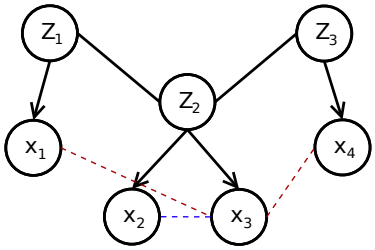

FIGURE 3.2 — Réseau de Markov pour le *clustering* semi-supervisé correspondant aux contraintes $(x_2, x_3) \in \mathcal{ML}$ et $(x_1, x_3) \in \mathcal{CL}$, $(x_3, x_4) \in \mathcal{CL}$. Les individus x_2 et x_3 doivent appartenir au même *chunklet*, traduit par le fait qu'ils partagent la même étiquette donnée par la réalisation de la variable Z_2. Les contraintes $\mathcal{CL}$ sont traduites par les liens entre les variables cachées correspondantes aux individus impliqués dans ces contraintes. Des contraintes $(x_1, x_2) \in \mathcal{CL}$ et $(x_2, x_4) \in \mathcal{CL}$ sont implicitement créées.

contraintes », la vraisemblance des paramètres étant donnée le modèle est donné par :

$$\mathcal{L}(\Theta; \mathcal{X}, Z, E_s) = \frac{1}{f(E_s|\Theta)} \prod_{l=1}^{L} \prod_{x_i \in \mathcal{X}_l} \alpha_{z_l}^{|z_l|} f(x_i|Z_l = e(x_i), \Theta) \tag{3.5}$$

$$\prod_{(x_i,x_j) \in \mathcal{CL}} (1 - \delta_{e(x_i),e(x_j)}) \tag{3.6}$$

et le problème d'optimisation consiste à maximiser la log-vraisemblance des données complétées :

$$\max_{\Theta} Q_{\text{ConsEM}}(\Theta) = \max_{\Theta} \log \mathcal{L}(\Theta; \mathcal{X}, Z, E_s) \tag{3.7}$$

Algorithme

L'algorithme permettant de résoudre le problème d'optimisation 19 est complètement basé sur EM. Il alterne une étape (E) de calcul de l'espérance des variables cachées correspondant aux *chunklets* tel qu'elle soit conformes aux contraintes, et une étape (M) d'estimation des meilleurs paramètres selon les dernières valeurs de probabilité *a posteriori*.

L'étape E permet de réévaluer les valeurs de probabilité *a posteriori* z_{ik} par :

$$\begin{aligned} z_{ik} &= f(Z_i = k | X_i = x_i; \Theta^*, E_s) \\ &= \frac{\alpha_k^{|\mathcal{X}_l|} \prod_{l=1}^{n_l} \prod_{x_i \in \mathcal{X}_l} f(x_i|z_l = k = e(x_i); \theta_k)}{\sum_{k'=1}^{n_k} \alpha_{k'}^{|\mathcal{X}_l|} \prod_{l=1}^{n_l} \prod_{x_i \in \mathcal{X}_l} f(x_i|z_l = k' = e(x_i); \theta_{k'})} \end{aligned} \tag{3.8}$$

L'étape M permet de réévaluer les paramètres Θ du modèle. Dans le cas de l'approche proposé, le modèle de mélange est gaussien, ainsi chaque composante du mélange correspond à une loi normale paramétrée par sa moyenne c_k et sa variance Σ_k. celles-ci sont calculés de manière

optimale par :

$$c_k = \frac{\sum_{x_i \in \mathcal{X}_i} x_i f(Z_l = e(x_i)|X_i = x_i, \Theta, E_s)}{\sum_{x_i \in \mathcal{X}_i} f(Z_l = e(x_i)|X_i = x_i, \Theta, E_s)} \tag{3.9}$$

et

$$\Sigma_k = \frac{\sum_{l=1}^{n_l} \sum_{x_i \in \mathcal{X}_l} (x_i - c_k)(x_i - c_k)^\top f(Z_l = e(x_i)|X_i = x_i, \Theta, E_s)}{\sum_{l=1}^{n_l} \sum_{x_i \in \mathcal{X}_l} f(Z_l = e(x_i)|X_i = x_i, \Theta, E_s)} \tag{3.10}$$

Algorithme 19 EM contraint

ENTRÉES : $\mathcal{X}$, n_k, $\mathcal{ML}$, $\mathcal{CL}$
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1: Initialisation aléatoire des n_k paramètres des lois $\{(c_1, \Sigma_1), ..., (c_{n_k}, \Sigma_{n_k})\}$
2: Étape E : Mise à jour des z_{ik} en utilisant (3.8)
3: Étape M : Mise à jour des c_k et Σ_k en utilisant (3.9) et (3.10)
4: Si Q_{ConsEM} change alors aller en **2**
5: $C_k = \{x_i \in \mathcal{X} | z_{ik} = \max_{k' \in [1..n_k]} z_{ik'}\}\ \forall k \in [1..n_k]$

3.4 Approches par objectif pénalisé

3.4.1 PCKM : les K-moyennes contraintes pénalisées

Parmi les premières approches de *clustering* semi-supervisé autorisant le non respect de quelques contraintes au profit de l'obtention d'une solution intéressante, [Basu et al., 2004] ont proposé une variante de KM (1.3.1.1) pour laquelle la solution optimale au sens du critère des K-moyennes doit pouvoir respecter au mieux les contraintes données.

Objectif

Le problème prend alors la forme d'un critère à optimiser, correspondant au critère de KM :
- pénalisé par un terme modélisant le non respect des contraintes $\mathcal{CL}$;
- récompensé par un terme modélisant le respect des contraintes $\mathcal{ML}$;

Si une contrainte est violée, alors un poids est ajouté au critère à minimiser. Ainsi le problème d'optimisation est représenté de la manière suivante :

$$\begin{aligned} &\min_{c,\ C} Q_{\text{PCKM}}(c, C) \\ &= \min_{c,\ C} \frac{1}{2} \sum_{k=1}^{n_k} \sum_{x_i \in C_k} ||x_i - c_k||_2^2 + \sum_{k=1}^{n_k} \sum_{\substack{(x_i,x_j) \in C_k^2 \\ (x_i,x_j) \in \mathcal{CL}}} w_{ij} + \sum_{k=1}^{n_k} \sum_{\substack{\overline{k}=1 \\ \overline{k} \neq k}}^{n_k} \sum_{\substack{(x_i,x_j) \in C_k \times C_{\overline{k}} \\ (x_i,x_j) \in \mathcal{ML}}} w_{ij} \end{aligned} \tag{3.11}$$

où les w_{ij} sont des paramètres donnés représentant les poids associés aux contraintes. Ils traduisent, pour chaque contrainte, l'impact de la violation de celle-ci sur le critère objectif de KM.

Algorithme

L'algorithme développé (algorithme 20) pour atteindre un optimum local du critère Q_{PCKM} est semblable à KM. Il alterne une étape d'affectation des individus à leur groupe le plus proche au sens de l'inertie pénalisée, et une étape de mise à jour des prototypes de ces groupes :

1. la phase d'affectation consiste à construire *in extenso* les n_k groupes par :

$$C_k^* = \{x_i \in \mathcal{X} \mid \underset{c \in \{c_1,\ldots,c_{n_k}\}}{arg\ min} \frac{1}{2}||x_i - c||_2^2 + \sum_{\substack{x_j \in C_k \\ (x_i,x_j)\in \mathcal{CL}}} w_{ij} + \sum_{\substack{\overline{k}=1 \\ \overline{k}\neq k}}^{n_k} \sum_{\substack{x_j \in C_{\overline{k}} \\ (x_i,x_j)\in \mathcal{ML}}} w_{ij} = c_k\} \tag{3.12}$$

2. la phase de mise à jour des prototypes permet de redéfinir les éléments représentatifs de ces groupes en recalculant les barycentres :

$$c_k^* = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i \tag{3.13}$$

Cependant pour faciliter la recherche d'une solution satisfaisant au mieux les contraintes, et ainsi éviter de tomber trop facilement dans des optimums locaux non souhaités, les auteurs proposent d'adapter la procédure d'initialisation. Ainsi chaque ensemble de contraintes $\mathcal{ML}$ et $\mathcal{CL}$ est augmenté le plus possible selon une logique de satisfaction associée aux contraintes. Si deux individus x_i et x_j sont liés par une contrainte $\mathcal{ML}$ et si x_j et x_k sont liés par une contrainte $\mathcal{ML}$, alors x_i et x_k sont également liés par une contrainte $\mathcal{ML}$:

$$\forall (x_i, x_j, x_k) \in \mathcal{X}^3, \quad (x_i, x_j) \in \mathcal{ML} \wedge (x_j, x_k) \in \mathcal{ML} \Rightarrow \mathcal{ML} = (x_i, x_k) \cup \mathcal{ML} \tag{3.14}$$

Ainsi l'opération de clôture transitive est appliquée au graphe associé aux contraintes $\mathcal{ML}$. Soit $\mathcal{N}$ l'ensemble des n_λ composantes connexes du graphe des $\mathcal{ML}$:

$$\mathcal{N} = \{\mathcal{N}_\lambda\}_{\lambda \in [1..n_\lambda]}$$

et soit $\mathcal{N}(x_i) = \{x_j \in \mathcal{N}_\lambda \mid x_i \in \mathcal{N}_\lambda\}$ alors l'ensemble des contraintes $\mathcal{CL}$ est augmenté de telle sorte que s'il existe une contrainte $\mathcal{CL}$ entre x_i et x_j tels que $\mathcal{N}(x_i) \neq \mathcal{N}(x_j)$, alors une contrainte $\mathcal{CL}$ est créée pour toute paire $(x_{k_1}, x_{k_2}) \in \mathcal{N}(x_i) \times \mathcal{N}(x_j)$:

$$\begin{aligned} &\forall (x_i, x_j, x_k) \in \mathcal{X}^3, \\ &\quad \forall x_k \in \mathcal{N}(x_i),\ (x_i, x_j) \in \mathcal{CL} \wedge \mathcal{N}(x_i) \neq \mathcal{N}(x_j) \Rightarrow \mathcal{CL} = \mathcal{CL} \cup (x_j, x_k) \\ &\quad \forall x_k \in \mathcal{N}(x_j),\ (x_i, x_j) \in \mathcal{CL} \wedge \mathcal{N}(x_i) \neq \mathcal{N}(x_j) \Rightarrow \mathcal{CL} = \mathcal{CL} \cup (x_i, x_k) \end{aligned} \tag{3.15}$$

La procédure d'initialisation consiste ensuite à choisir les n_k centres initiaux respectant au mieux les contraintes *i.e.* tirés parmi les n_λ composantes connexes de $\mathcal{N}$:

- si $n_k \geq n_\lambda$ alors les prototypes initiaux sont choisis parmi les n_k composantes connexes les plus grandes en cardinalité ;
- si $n_k < n_\lambda$ alors les prototypes initiaux sont choisis parmi les n_λ composantes connexes, puis ensuite parmi les individus liés par une contrainte $\mathcal{CL}$ avec toutes les composantes connexes de $\mathcal{N}$. Enfin les centres initiaux éventuels restant à initialiser sont tirés aléatoirement.

Discussion

Algorithme 20 PCKM

ENTRÉES : $\mathcal{X}$, n_k, $\mathcal{ML}$, $\mathcal{CL}$, W
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation des n_k centres de groupes $\{c_1, ..., c_{n_k}\}$
2 : Mise à jour des groupes C_k en utilisant (3.12)
3 : Mise à jour des centres de groupe c_k en utilisant (3.13)
4 : Si Q_{PCKM} change alors aller en **2**

L'approche PCKM est une des premières approches exprimée explicitement comme la recherche d'une solution optimale à un critère objectif où l'intégration des contraintes est réalisée par une pénalisation de celui-ci. On peut reprocher à l'approche PCKM que les poids w_{ij} soient fixés à l'avance et que leur définition ne soit pas explicite. Or ces poids sont centraux dans la recherche d'une solution satisfaisant effectivement les contraintes. Une amélioration à envisager serait de les ré-estimer lors du déroulement de l'algorithme.

3.4.2 SSKM : les K-moyennes semi-supervisées

L'approche SSKM de [Kulis et al., 2005] reprend l'idée de PCKM mais propose un algorithme interprétable de façon complètement différente de ce dernier. Il s'agit d'une approche discriminative qui reprend l'objectif de PCKM en y incorporant des modifications mineures.

Objectif

Le problème est posé comme la minimisation du critère d'inertie de KM encore une fois réajusté par un terme relatif au respect des contraintes $\mathcal{ML}$ et $\mathcal{CL}$:

$$\begin{aligned} &\min_{c,\, C} Q_{\text{SSKM}}(c, C) \\ &= \min_{c,\, C} \sum_{k=1}^{n_k} \sum_{x_i \in C_k} ||x_i - c_k||_2^2 + \sum_{k=1}^{n_k} \sum_{\substack{(x_i,x_j)\in C_k^2 \\ (x_i,x_j)\in \mathcal{CL}}} \frac{w_{ij}}{|C_k|} - \sum_{k=1}^{n_k} \sum_{\substack{(x_i,x_j)\in C_k^2 \\ (x_i,x_j)\in \mathcal{ML}}} \frac{w_{ij}}{|C_k|} \end{aligned} \tag{3.16}$$

Le terme d'inertie de KM est cette fois pénalisé par le non respect des contraintes $\mathcal{CL}$, et récompensé par le respect des contraintes $\mathcal{ML}$.

Algorithme

[Kulis et al., 2005] ont montré que ce critère pouvait se ré-exprimer plus simplement en utilisant l'astuce du noyau (cf. section 2.4.3). Ainsi minimiser le critère Q_{SSKM} revient à minimiser le critère Q_{KM} pour lequel les individus sont projetés par l'application ϕ inconnue vers un espace de représentation P muni du produit scalaire $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$:

$$Q_{\text{SSKM}}(c, C) = Q_{\text{KKM}}(c, C) = \sum_{k=1}^{n_k} \sum_{x_i \in C_k} ||\phi(x_i) - c_k||_2^2$$

où $K_{ij} = \langle x_i, x_j \rangle + W_{ij}$ et W est construit par :

$$W_{ij} = \begin{cases} w_{ij} & \forall (x_i, x_j) \in \mathcal{ML} \\ -w_{ij} & \forall (x_i, x_j) \in \mathcal{CL} \end{cases}$$

L'algorithme de résolution (Algorithme 22) est alors connu et correspond à un simple KM à noyau, ou KKM, appliqué sur le noyau $K = S + W$ où S est la matrice des produits scalaires dans l'espace d'origine (avant projection par ϕ) : $S_{ij} = \langle x_i, x_j \rangle$. Il consiste alors, à partir d'une initialisation de prototypes de groupes tirés parmi les individus, à alterner :

1. l'étape d'affectation des individus à leur groupe le plus proche :

$$C_k^* = \{x_i \in \mathcal{X} | \underset{c \in \{c_1, ..., c_{n_k}\}}{arg\ min} ||\phi(x_i) - c||_2^2 = c_k\} \tag{3.17}$$

2. l'étape de mise à jour implicite des prototypes, par un calcul de leurs distances par rapport aux individus $d_P(x_i, c_k^*) = ||\phi(x_i) - c_k||_2^2$:

$$||\phi(x_i) - c_k||_2^2 = K_{ii} - 2\frac{\sum_{x_j \in C_k} K_{ij}}{|C_k|} + \frac{\sum_{x_j \in C_k} \sum_{x_l \in C_k} K_{jl}}{|C_k|^2} \tag{3.18}$$

L'algorithme revient donc à appliquer KM sur $\mathcal{X}$ où les distances entre individus sont altérées *a priori* pour se conformer aux contraintes $\mathcal{CL}$ et $\mathcal{ML}$.

Algorithme 21 SSKM

ENTRÉES : $\mathcal{X}$, n_k, $\mathcal{ML}$, $\mathcal{CL}$, W
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation des n_k centres de groupes $\{c_1, ..., c_{n_k}\}$
2 : Construire le noyau $K = S + W$
3 : Mise à jour des groupes C_k en utilisant (3.17)
4 : Mise à jour des distances aux centres $d_P(x_i, c_k^*)$ par (3.18)
5 : Si Q_{PCKM} change alors aller en **2**

Discussion

L'approche SSKM permet de faire le lien entre les approches de *clustering* semi-supervisé basé sur la pénalisation et celles basé sur l'altération de la proximité. En effet, les auteurs établissent que la recherche d'une solution optimale de leur critère pénalisé est obtenable au travers d'un *clustering* classique après que les mesures de distance entre les individus aient été redéfinies. Dans le contexte actuel des recherches pour le *clustering* semi-supervisé, on regrette l'imposition de l'algorithme KM, mais cela est nécessaire pour garantir un contrôle complet sur l'optimisation.

3.5 Approches par altération de la proximité

3.5.1 LLMA : adaptation localement linéaire de la métrique

L'approche d'adaptation localement linéaire de la métrique [Chang and Yeung, 2004] vise à trouver une projection de l'ensemble des individus de $\mathcal{X}$ telle que les individus devant être classés ensembles se retrouvent plus proches dans cet espace de projection. L'originalité de l'approche réside dans les propriétés de cette projection. En effet, les auteurs proposent de trouver une projection qui soit :

- localement linéaire, dans le sens où les individus impliqués dans les contraintes $\mathcal{ML}$ (de base ou induites par transitivité) ainsi que les individus proches de ceux-ci sont projetés linéairement dans un nouvel espace P;
- globalement non linéaire, dans le sens où tous les individus, et en particulier ceux qui ne sont pas concernés par des contraintes, sont projetés non linéairement dans P.

Objectif

La projection $\phi : \mathbb{R}^p \mapsto \mathbb{R}^p$ est linéaire et définie explicitement sous la forme :

$$\phi(x_l) = x_l + \sum_{(x_i,x_j)\in\mathcal{ML}} K_{1_{li}} b_i = x_l + BK_{1_l} \tag{3.19}$$

où

$$K_{1_{li}} = e^{-\frac{||x_i - x_l||_2^2}{2\sigma_1^2}} \tag{3.20}$$

modélise une similarité entre les individus x_i et x_l. Ainsi plus un individu x_l est loin des individus impliqués dans au moins une contrainte $\mathcal{ML}$ moins la projection altère x_l, et $\Phi(x_l) \mapsto x_l$. Le problème prend la forme d'un critère objectif pénalisé pour lequel la solution optimale correspond aux paramètres de la projection : la matrice $B = (b_1, \ldots, b_{n_\mathcal{C}})$. L'objectif est alors de minimiser la distance entre $\phi(x_i)$ et $\phi(x_j)$ $\forall(x_i, x_j) \in \mathcal{ML}$ tout en préservant les écarts entre x_i et x_j $\forall(x_i, x_j) \notin \mathcal{ML}$:

$$\begin{aligned} &\min_B \; Q_{\text{LLMA}}(B) \\ &= \min_B \sum_{(x_i,x_j)\in\mathcal{ML}} ||\phi(x_i) - \phi(x_j)||_2^2 + \eta \sum_{(x_i,x_j)\in\mathcal{X}^2} K_{2ij}\xi_{ij} \end{aligned} \tag{3.21}$$

où $n_\mathcal{C}$ correspond au nombre d'individus impliqués dans les contraintes $\xi_{ij} = (||\phi(x_i) - \phi(x_j)||_2 - ||x_i - x_j||_2)^2$ correspond à l'écart entre les distances avant et après projection entre les individus x_i et x_j. K_2 est une fonction de similarité prédéfinie gaussienne entre les individus x_i et x_j avant projection :

$$K_{2ij} = e^{-\frac{||x_i - x_j||_2^2}{\sigma_2^2}} \tag{3.22}$$

La valeur $K_{2_{ij}}$ joue le rôle de poids pour le second terme du critère Q_{LLMA}. Ainsi plus deux individus x_i et x_j seront proches au sens de la distance euclidienne, plus ils seront similaire au sens de K_2 et plus on privilégiera le fait de conserver cette valeur de distance après projection, sauf dans le cas où ces individus sont impliqués dans une contrainte $\mathcal{ML}$ donnée ou induite.

Algorithme

L'algorithme consiste à alterner différentes étapes afin de déterminer la projection optimale caractérisée par B^* :

- une mise à jour des paramètres de la mesure de similarité K_1 : σ_1 ;
- une mise à jour des paramètres de la mesure de similarité K_2 : σ_2 ;
- la mise à jour optimale des variables b_i ;
- la redéfinition de la position des individus dans l'espace.

Les paramètres des mesures de similarités K_1 et K_2 sont déterminés de manière heuristique par :

$$\sigma_1 = \lambda_1 \frac{V}{\sqrt{t}} \; ; \; \sigma_2 = \lambda_2 \sigma_1 \tag{3.23}$$

où $\lambda_1 \geq 0$, $\lambda_2 \geq 0$ sont des constantes données et V correspond à la valeur de distance moyenne entre individus projetés :

$$\frac{2}{n(n-1)} \sum_{\substack{(x_i,x_j)\in\mathcal{X}^2 \\ i<j}} ||\phi(x_i) - \phi(x_j)||_2^2$$

ainsi plus le nombre d'itérations est élevé, plus le paramètre de variance σ_1 diminue, entraînant également une diminution de σ_2. Au bout du compte les valeurs de similarité correspondantes $K_{1_{ij}}$ et $K_{2_{ij}}$ tendent vers les valeurs extrêmes 0 ou 1 pour toute paire d'individus $(x_i, x_j) \in \mathcal{X}^2$. Étant données de telles valeurs de σ_1 et σ_2 et la position courante des individus x_i, les paramètres B de la prochaine transformation sont calculés de manière optimale ou quasi optimale. Décrire les conditions d'optimalité de la solution $B^* = (b_1^*, \ldots, b_{n_\mathcal{C}}^*)$ *i.e.* $\nabla_B Q = \mathbf{0}$ ne permet pas d'obtenir une forme close de la solution. Cependant, les auteurs proposent d'approximer une telle solution en maintenant dans l'expression du critère une contrainte $\xi_{ij} = 0$. Dans ce contexte B^* peut être déterminé explicitement par :

$$B^* = -B_1 B_2^\dagger \tag{3.24}$$

avec

$$\begin{aligned} B_1 &= \sum_{(x_i,x_j)\in\mathcal{X}^2} \Big(\big(s_{ij} + \eta K_{2ij}(1 - \frac{||x_i - x_j||_2^2}{||\phi(x_i) - \phi(x_j)||_2^2})\big) \\ &\quad . (\phi(x_i) - \phi(x_j))(K_{1_{:i}} - K_{1_{:j}})^\top \Big) \\ B_2 &= \sum_{(x_i,x_j)\in\mathcal{X}^2} \Big(\big(s_{ij} + \eta K_{2ij}(1 - \frac{||x_i - x_j||_2^2}{||\phi(x_i) - \phi(x_j)||_2^2})\big) \\ &\quad . (K_{1_{:i}} - K_{1_{:j}})(K_{1_{:i}} - K_{1_{:j}})^\top \Big) \end{aligned}$$

et

$$s_{ij} = \begin{cases} 1 & \text{si } (x_i, x_j) \in \mathcal{ML} \\ 0 & \text{sinon} \end{cases}$$

Les auteurs proposent également un autre moyen d'optimiser leur critère sans faire l'hypothèse restrictive $\xi_{ij} = 0$ mais cette seconde procédure, reposant sur un principe de majoration itérative ne sera pas détaillée davantage.

Algorithme 22 LLMA

ENTRÉES : $\mathcal{X}$, X, n_k, $\mathcal{ML}$, W, t_f
SORTIES : X'
1 : Réaliser la clôture réflexive et transitive de $\mathcal{ML}$
2 : Initialiser $\phi(x_i) = x_i \; \forall x_i \in \mathcal{X}$, $t = 1$
3 : Mise à jour de σ_1 et σ_2 en utilisant (3.23)
4 : Mise à jour de K_1 et K_2 en utilisant (3.20) et (3.22)
5 : Mise à jour optimale de B par (3.24)
6 : Si $t = t_f$ alors $t = t + 1$ et aller en **3**

Discussion

L'approche LLMA est intéressante en ce qui concerne la gestion des contraintes $\mathcal{ML}$. Seuls les individus impliqués dans de telles contraintes sont effectivement projetés de telle sorte à être rapprochés. En revanche, l'approche ne permet pas la gestion de contrainte de type $\mathcal{CL}$ ce qui limite son applicabilité dans les contextes plus actuels. De plus l'approche souffre de quelques artefacts pour garantir l'obtention d'une solution optimale du problème d'optimisation ainsi qu'une convergence de l'algorithme associé. La décroissance programmée des variances associées aux gaussiennes K_1 et K_2 rappellent l'utilisation du paramètre de température dans les approches de type SOM (cf. section 1.3.2.2).

3.6 Approches indépendantes de l'algorithme de *clustering*

Les approches d'apprentissage de distances ou de similarités peuvent être vus comme des approches indépendantes de l'algorithme de *clustering*. L'intégration des contraintes se fait alors en amont de l'application du *clustering*. Une distinction est cependant faite dans la mesure où l'on s'intéresse à l'impact de la mesure de proximité apprise sur la performance de l'algorithme de *clustering* employé dans le but de corriger cet apprentissage de proximité pour que celle-ci soit en adéquation avec :

- les contraintes $\mathcal{ML}$ et $\mathcal{CL}$;
- l'amélioration de la performance de A ;
- la distribution naturelle des individus dans l'espace, X.

Le principe de fonctionnement de cette famille d'approches initiées par BOOSTCLUSTER [Liu et al., 2007] est de générer successivement un ensemble d'hypothèses H de *clustering* selon différentes mesures de proximité apprises de telle sorte à respecter les contraintes utilisateurs. A partir de cet ensemble d'hypothèses est construit le *clustering* C, qui dans ce contexte devra être de meilleure qualité qu'un *clustering* obtenu selon les techniques d'intégration simple dans la proximité.

3.6.1 BC : *BoostCluster*

L'approche BC [Liu et al., 2007] permet de s'abstraire de l'algorithme de *clustering* employé afin de trouver un bon partitionnement respectant les contraintes. Il propose d'intégrer des informations de semi-supervision de type $\mathcal{ML}$ et $\mathcal{CL}$ dans n'importe quel algorithme de *clustering* A selon des techniques empruntées à l'apprentissage de distances. En particulier, la distance apprise s'adapte à l'algorithme employé afin que celui-ci satisfasse le mieux possible les contraintes données.

Objectif

Le principe est d'apprendre une matrice de similarité K de sorte que celle-ci respecte les contraintes, ainsi :

- $(x_i, x_j) \in \mathcal{ML}$ doit induire une valeur de K_{ij} élevée ;
- $(x_i, x_j) \in \mathcal{CL}$ doivent induire une valeur de K_{ij} faible ;
- de plus, l'apprentissage de K doit être validé par l'algorithme de *clustering* A. Ainsi K réalise un compromis entre l'intégration optimale des contraintes et la satisfaction de celle-ci par A.

Le problème prend alors la forme d'un programme d'optimisation où il s'agit de trouver une *bonne* solution au problème :

$$\underset{K}{min}\ Q_{\text{BC}} = \underset{K}{min} \Big(\sum_{(x_i,x_j)\in\mathcal{ML}} e^{-K_{ij}} \Big) \Big(\sum_{(x_i,x_j)\in\mathcal{CL}} e^{K_{ij}} \Big) \tag{3.25}$$

La difficulté réside dans le fait qu'il n'est pas possible d'estimer à l'avance la satisfaction par A (car l'objectif de A n'est pas connu) des contraintes $\mathcal{ML}$ et $\mathcal{CL}$ étant donnée une valeur de K. Ainsi cette information ne peut être traduite directement dans l'expression du critère à optimiser. De plus, la matrice K^* optimale n'est pas unique, son expression est connue et elle ne correspond pas nécessairement à la meilleure matrice pour l'amélioration de la performance de A. La matrice K^* optimale est donnée par :

$$K_{ij} = \begin{cases} 1 & \forall (x_i, x_j) \in \mathcal{ML} \\ 0 & \forall (x_i, x_j) \in \mathcal{CL} \\ \alpha_{ij} & \forall (x_i, x_j) \notin \mathcal{ML} \cup \mathcal{CL} \end{cases}$$

où α_{ij} est une valeur arbitraire. Ainsi, l'objectif est d'améliorer le critère Q_{BC} en cherchant K tel que la performance de A soit améliorée au mieux. Ce faisant, K est alors une *bonne* solution.

Algorithme

L'algorithme proposé (algorithme 23) pour résoudre ce problème d'optimisation consiste à alterner trois étapes garantissant l'obtention d'un K améliorant son adéquation avec les contraintes et améliorant la performance de A sur le respect des contraintes. Soit $K^{(0)} = \mathbf{0}$ la valeur initiale de la matrice K, le K^* optimal est construit de manière incrémentale à l'issu de la convergence de la suite $(K^{(t)})_{t\in[1..t_f]}$ où $K^{(t)} = f(K^{(t-1)})$. La première étape consiste à proposer une transformation de X en X^* de sorte que :

- des individus x_i et x_j tels que $(x_i, x_j) \in \mathcal{ML}$ soient davantage rapprochés relativement aux autres paires d'individus, si leur valeur de similarité est faible $(cond\ 1)$;
- des individus x_i et x_j tels que $(x_i, x_j) \in \mathcal{CL}$ restent d'autant éloignés que leur valeur de similarité est forte $(cond\ 2)$.

Pour cela des poids w_{ij} sont calculés tels que :

$$w_{ij} = \begin{cases} \dfrac{e^{-K_{ij}}}{Z_{\mathcal{ML}}} & \forall (x_i, x_j) \in \mathcal{ML} \\ \\ -\dfrac{e^{K_{ij}}}{Z_{\mathcal{CL}}} & \forall (x_i, x_j) \in \mathcal{CL} \end{cases} \tag{3.26}$$

où $Z_{\mathcal{ML}}$ et $Z_{\mathcal{CL}}$ sont des facteurs de normalisation. Ainsi, les poids reflètent exactement les conditions $(cond\ 1)$ et $(cond\ 2)$.

Ces poids servent à déterminer un sous espace de projection $P^* \in \mathbb{R}^{p\times s}$ solution du problème d'optimisation :

$$\begin{array}{lll} \max\limits_{P} & trace(P^\top X^\top W X P) & \\ s.t. & P^\top P & = & Id_s \end{array} \tag{3.27}$$

où s est la dimension du sous-espace (fixé à l'avance dans BC) et W est la matrice des poids définit par $W_{ij} = w_{ij}$.

La nouvelle représentation X^* s'obtient alors en projetant X *via* P^* où $X^* = XP^*$. L'application de l'algorithme A sur X^* permet d'observer son comportement face à la nouvelle représentation. Soit $H^{(t)}$ le *clustering* produit par A :

$$H_{ij}^{(t)} = \begin{cases} 1 & \text{si } Link(x_i, x_j, A) \\ 0 & \text{si } \overline{Link}(x_i, x_j, A) \end{cases} \tag{3.28}$$

H prend la forme d'une hypothèse car dépendante de la valeur de similarité courante $K^{(t)}$. Cette hypothèse permet de réévaluer la valeur de similarité K selon la simple équation :

$$K^{(t)} = K^{(t-1)} + \alpha^{(t)} H^{(t)} \tag{3.29}$$

où $\alpha^{(t)} \geq 0$ quantifie le ratio du nombre de contraintes satisfaites sur le nombre de contraintes violées :

$$\alpha^{(t)} = \frac{1}{2} \log \left(\frac{\sum\limits_{\substack{(x_i,x_j)\in\mathcal{ML} \\ H_{ij}^{(t)}=1}} |w_{ij}|}{\sum\limits_{\substack{(x_i,x_j)\in\mathcal{ML} \\ H_{ij}^{(t)}=0}} |w_{ij}|} \times \frac{\sum\limits_{\substack{(x_i,x_j)\in\mathcal{CL} \\ H_{ij}^{(t)}=0}} |w_{ij}|}{\sum\limits_{\substack{(x_i,x_j)\in\mathcal{CL} \\ H_{ij}^{(t)}=1}} |w_{ij}|} \right) \tag{3.30}$$

Le premier terme entre parenthèses correspond à la part (pondérée) de contraintes $\mathcal{ML}$ satisfaites et le second, à la part (pondérée) de contraintes $\mathcal{CL}$ satisfaites.

Algorithme 23 BC

ENTRÉES : $\mathcal{X}$, n_k, $\mathcal{ML}$, $\mathcal{CL}$
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation de $K = \mathbf{0}$, $t = 0$
2 : Calcul de W par (3.26)
3 : Calcul de $X^* = XP^*$ après résolution de (3.27)
4 : Estimation de $H^{(t)}$ en appliquant A sur X^* par (3.28)
5 : Mise à jour de K selon (3.29)
6 : Si K ne converge pas faire $t = t + 1$ et aller en **2**
7 : $C =$ *clustering* de $\mathcal{X}$ par A en utilisant K^*

Discussion

L'approche BC permet d'améliorer la performance de n'importe quel algorithme de *clustering* A en fournissant à celui-ci une matrice de similarité K adaptée au comportement de A vis à vis de la satisfaction des contraintes $\mathcal{CL}$ et $\mathcal{ML}$. La matrice K est apprise à partir de la génération d'un ensemble de t_f espaces de représentations permettant d'en déduire t_f hypothèses de *clustering* de $\mathcal{X}$:

$$K^* = \sum_{t=1}^{t_f} \alpha^{(t)} H^{(t)} \tag{3.31}$$

Le nombre d'étape t_f de l'algorithme est, selon les auteurs, imposé. Néanmoins, on peut ne pas fixer ce paramètre et attendre d'observer les erreurs de l'algorithme A sur la satisfaction des contraintes. En effet, si dans l'expression de $\alpha^{(t)}$ (3.30) la quantité (pondérée) de contraintes violées exprimée par le dénominateur est plus grande que la quantité de contraintes satisfaites exprimée par le numérateur, alors l'expression de $\alpha^{(t)}$ est négative et contredit les hypothèses faites pour la construction itérative de K (3.29).

Un autre point que l'on peut soulever au regard des approches précédentes de *clustering* semi-supervisé, est que l'approche échoue par son critère objectif (3.25), à proposer une intégration de contraintes $\mathcal{ML}$ seules ou de $\mathcal{CL}$ seules, ce qui peut arriver régulièrement dans des cas concrets d'application. De plus, lors de la génération de chaque nouvelle représentation, celle-ci est déterminée uniquement selon les poids w_{ij} associés aux individus x_i et x_j impliqués dans les contraintes. Autrement dit, les individus qui ne sont impliqués dans aucune contraintes, ne sont pas considérés lors de la recherche du sous-espace de projection optimal P^* (3.27).

Enfin, on peut s'interroger sur la discontinuité entre (1) les résultats de *clusterings* intermédiaires obtenus lors du processus itératif *via* application de A sur un nouvel espace de représentation, et (2) le *clustering* final qui est obtenu, non pas par application sur un nouvel espace, mais par l'utilisation d'une nouvelle mesure de similarité.

3.7 Contributions

3.7.1 Motivation

Les contributions proposées reprennent les principes des approches indépendantes de l'algorithme dans la lignée de BC. Le concept est assez similaire dans le sens où les solutions

proposées sont des méta-algorithmes dont l'objectif est d'offrir à chaque étape un sous-espace de projection permettant à l'algorithme de *clustering* de respecter au mieux les contraintes $\mathcal{ML}$ et $\mathcal{CL}$. Nous avons vu que l'approche BC se focalise dans l'expression de la fonction objectif à optimiser, uniquement sur les paires d'individus impliqués dans les contraintes données. Ce choix offre des avantages, comme la faible complexité et le succès quant à l'obtention d'un sous espace dans lequel des individus devant être regroupés (resp. séparés) se retrouvent proches (resp. éloignés). Néanmoins, il est aussi limitant dans le contexte du *clustering* sous contraintes, dans la mesure où il ne réalise pas explicitement l'hypothèse que des individus proches d'autres individus impliqués dans les contraintes devraient se comporter de manière semblable vis à vis de ces contraintes. Plus formellement :

$$\begin{aligned} \forall (x_i, x_j) \in \mathcal{X}^2,\ (x_i \ proche\ de\ x_j \wedge (x_j, x_l) \in \mathcal{ML}) &\Rightarrow (x_i, x_l) \in \mathcal{ML} \\ \forall (x_i, x_j) \in \mathcal{X}^2,\ (x_i \ proche\ de\ x_j \wedge (x_j, x_l) \in \mathcal{CL}) &\Rightarrow (x_i, x_l) \in \mathcal{CL} \end{aligned}$$

Cette hypothèse est centrale dans les travaux de [Klein et al., 2002] (cf. section 3.3.2). Ici, on ne cherche pas explicitement à imposer ces contraintes. En revanche, on aimerait qu'elles soient naturellement identifiées lors de la détermination du sous-espace de projection des données. L'hypothèse émise est qu'alors un sous-espace de projection respectant au mieux la représentation d'origine des données permettra cette identification. En effet si nous pouvons nous assurer qu'un individu x_i proche d'un individu x_j impliqué dans une contrainte, dans l'espace d'origine, reste proche de lui dans le sous-espace de projection, nous réalisons l'hypothèse. Nous identifions alors deux principes clés que nous chercherons à respecter en vue d'obtenir une nouvelle représentation favorisant le respect des contraintes par l'algorithme de *clustering* :

- la **cohérence** vis à vis de la représentation d'origine des données. La nouvelle représentation devra être fidèle à la représentation d'origine.
- la **consistance** sur le respect des contraintes données par l'utilisateur. Dans la nouvelle représentation, des individus impliqués dans une contrainte $\mathcal{ML}$ (resp. $\mathcal{CL}$) devront être proches (resp. éloignés).

Les deux approches proposées et présentées par la suite diffèrent sur la manière de modéliser et d'intégrer ces deux principes ainsi que sur la manière d'intégrer l'observation de la performance de A, vue dans cette famille d'approche comme un évaluateur de la proximité apprise.

Une approche de type *boosting*

La première approche que nous proposons reprend un formalisme de type *boosting* dans un cadre non supervisé. L'idée du *boosting* est apparu dans le contexte de l'apprentissage supervisé. L'objectif est de guider l'entraînement d'un classifieur dit faible car fournissant un ensemble d'hypothèses assez erronées mais se comportant mieux qu'un classifieur aléatoire, en vue de l'améliorer. Il s'agit d'un méta-algorithme qui consiste itérativement à apprendre un modèle à partir des données *via* le classifieur faible, en tenant compte, pour chaque modèle, des erreurs commises par le modèle précédent. Cette prise en compte est réalisée au moyen de poids que l'on associe aux exemples d'apprentissage. L'idée étant qu'un poids fort sera associé à un exemple sur lequel le classifieur s'est précédemment trompé, et un poids faible est associé aux exemples bien classés. Ainsi à chaque étape, et *via* la pondération sur l'ensemble des exemples, un nouveau modèle est appris, réalisant des erreurs différentes au fur et à mesure des itérations. L'objectif étant d'obtenir un classifieur de meilleure qualité sur les données d'entraînement, celui-ci devra tenir compte de chaque classifieur appris à chaque étape du méta-algorithme de *boosting*. Le classifieur final est obtenu au moyen d'un vote pondéré par les confiances accordées

aux différents classifieurs, confiances relatives aux erreurs réalisées par ceux-ci. L'algorithme BoC calque le principe du *boosting* dans le contexte du *clustering* semi-supervisé. Le principe est de tenir à jour une distribution des poids sur l'ensemble des paires d'individus impliqués dans les contraintes (les exemples pondérés sont les paires d'individus). Nous augmentons le poids associé à une paire d'individus si A ne respecte pas la contrainte ($\mathcal{ML}$ ou $\mathcal{CL}$) correspondante à cette paire d'individus et nous diminuons le poids associé à une paire d'individus impliqués dans une contrainte satisfaite par A.

Une approche basée sur l'optimisation numérique

La deuxième approche proposée quant à elle, même si elle est extrêmement proche de la précédente de part l'expression de l'objectif, est sensiblement différente sur la résolution. Nous choisissons de nous inspirer de l'optimisation numérique pour trouver une solution optimale au problème posé. Dans cet algorithme l'idée est d'apprendre à chaque étape une nouvelle représentation des individus meilleure que la précédente dans le sens où A doit parvenir de mieux en mieux à satisfaire les contraintes données par l'utilisateur. La différence profonde concernant la résolution est qu'alors *via* cette approche il n'est pas nécessaire de réaliser un vote consensuel entre l'ensemble des différentes hypothèses obtenues à chaque étape du méta-algorithme mais de n'en conserver que les dernières. En ce qui concerne les ressemblances avec la précédente approche, la construction de la nouvelle représentation des individus est réalisée également à travers l'utilisation d'une distribution de poids sur l'ensemble des paires d'individus impliqués dans les contraintes. En revanche les poids ne sont pas mis à jour à la manière du *boosting* mais sont estimés de manière adaptée et par optimisation, pour satisfaire le principe de consistance. De plus, ils servent à pénaliser un objectif visant à satisfaire le principe de cohérence qui lui doit être optimisé. Le concept général de ces deux approches est schématisé dans la figure 3.3.

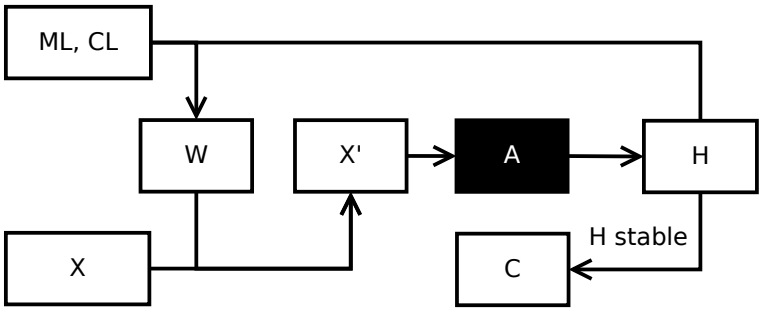

FIGURE 3.3 — Schéma général du déroulement des méta-algorithmes pour le *clustering* semi-supervisé. W désigne la matrice des poids, et X' la représentation optimale obtenue à partir du calcul du sous-espace P^* optimal.

Les algorithmes proposés reposent sur l'optimisation d'un critère objectif. Ce critère doit intégrer la volonté de satisfaire simultanément les deux principes que sont la cohérence et la consistance. La solution optimale pour ce critère objectif doit alors correspondre à un sous-espace réalisant, après projection de l'ensemble des individus dans celui-ci, un compromis entre :

- le respect de la représentation d'origine d'une part ;
- l'adéquation avec les contraintes utilisateurs d'autre part.

Le respect de la représentation d'origine : la cohérence

En ce qui concerne le premier point, les deux approches proposées reposent sur la même technique bien connue et éprouvée par les communautés issues de la *Statistique* et de *l'Analyse de Données* : l'*analyse en composante principale* ou ACP. L'idée de cet outil est d'offrir un moyen de représenter de manière optimale un ensemble d'individus décrits dans un espace vectoriel de dimension p, dans un sous-espace vectoriel de dimension $s < p$. La nouvelle représentation est optimale dans le sens ou elle préserve le maximum d'information présente dans la représentation d'origine. L'information préservée est la variance du nuage des individus, ce qui correspond à la dispersion de l'ensemble des individus relativement à leur centre de gravité. Dans la suite de ce chapitre, la métrique d correspondra à la métrique euclidienne $||.||_2$. Si on considère l'ensemble d'individu centré, où le nuage est translaté de sorte que le centre de gravité coincide avec l'origine du repère (0), le critère se formalise de la façon suivante :

$$Q_{\text{COH}}(P) = \sum_{(x_i,x_j)\in\mathcal{X}^2} d_P^2(x_i,x_j) = 2n \sum_{x_i\in\mathcal{X}} d_P^2(0,x_i) \tag{3.32}$$

et le problème d'optimisation associé à la recherche de cohérence est alors :

$$\max_P \sum_{(x_i,x_j)\in\mathcal{X}^2} d_P^2(x_i,x_j)$$

Le choix de l'ACP comme moyen d'obtenir un nouvel espace de représentation cohérent avec la représentation d'origine se justifie pleinement par l'optimalité de la solution puisqu'elle offre intuitivement un sous-espace dans lequel la distribution des individus projetés est la plus proche possible de la distribution des individus dans l'espace d'origine. Le respect de la représentation d'origine correspond au principe de cohérence.

Le respect des connaissances : la consistance

Le problème est maintenant de modéliser la volonté de respecter les connaissances représentées par les contraintes $\mathcal{ML}$ et $\mathcal{CL}$. L'intégration proposée se fonde sur les approches de type PCKM et SSKM. Plutôt que de pénaliser le critère objectif d'un algorithme de *clustering* particulier à l'image des approches précédentes (Q_{KM}), nous pénalisons le critère Q_{COH} par un terme pénalisant devant traduire le non respect des connaissances. La performance de A sur la satisfaction de ces contraintes n'étant pas prédictible, une expression analytique ne peut être écrite pour constituer un tel terme pénalisant. La modélisation proposée doit donc se fonder sur des hypothèses qui elles peuvent être traduites analytiquement, et qui, si elles sont vérifiées, devraient permettre d'atteindre l'objectif initial :

- si $(x_i,x_j) \in \mathcal{ML}$, alors plus les individus sont proches dans la nouvelle représentation, plus A aura de chance de satisfaire la contrainte $\mathcal{ML}$;
- si $(x_i,x_j) \in \mathcal{CL}$, alors plus les individus sont éloignés dans la nouvelle représentation, plus A aura de chance de satisfaire la contrainte $\mathcal{CL}$.

C'est sur ce point, l'intégration de l'objectif de la recherche de consistance, que les différentes contributions proposées diffèrent.

3.7.2 BoC : *boosting* de *clustering*

L'approche BoC suggère d'associer un critère objectif modélisant la recherche de consistance. Le critère proposé est le suivant :

$$Q_{\text{CST}}(P) = \sum_{(x_i,x_j)\in\mathcal{CL}} w_{ij} d_P^2(x_i,x_j) - \sum_{(x_i,x_j)\in\mathcal{ML}} w_{ij} d_P^2(x_i,x_j) \tag{3.33}$$

De part ce critère à maximiser selon $P \in \mathbb{R}^{p \times s}$ et $s < p$, paramétré notamment par les poids $w_{ij} > 0$, il est possible de réaliser les hypothèses précédentes. Ainsi :

- si $(x_i, x_j) \in \mathcal{CL}$, alors plus w_{ij} est grand, plus la distance dans l'espace de projection $d_P^2(x_i, x_j)$ devra être élevée.
- si $(x_i, x_j) \in \mathcal{ML}$, alors plus w_{ij} est grand, plus la distance dans l'espace de projection $d_P^2(x_i, x_j)$ devra être faible.

Les poids w_{ij} constituent alors un moyen de réaliser l'hypothèse en forçant la recherche d'une topologie en adéquation avec les contraintes $\mathcal{ML}$ et $\mathcal{CL}$.

Objectif

L'objectif global de l'approche est d'apprendre de manière itérative un ensemble de représentations de $\mathcal{X}$ en observant la performance de A sur la satisfaction des contraintes $\mathcal{ML}$ et $\mathcal{CL}$, permettant ainsi à A de produire un ensemble H d'hypothèses de *clustering*. Cet objectif ne peut être formalisé tel quel, dû à l'absence de connaissances sur A. L'idée est alors de proposer un formalisme :

- adapté pour permettre la recherche d'une représentation optimale X^* ;
- paramétré pour pouvoir intégrer un encodage de la performance de A.

Le critère proposé prend alors la forme d'un compromis :

$$\begin{aligned} Q_{\text{BoC}}(P) &= \frac{1-\eta}{n^2} Q_{\text{COH}}(P) + \frac{\eta}{m} Q_{\text{CST}}(P) \\ &= Q_{\text{COH}}(P) + reg_1(\eta) Q_{\text{CST}}(P) \\ &= \sum_{(x_i,x_j) \in \mathcal{X}^2} d_P^2(x_i, x_j) \\ &\quad + reg_1(\eta) \Big(\sum_{(x_i,x_j) \in \mathcal{CL}} w_{ij} d_P^2(x_i, x_j) - \sum_{(x_i,x_j) \in \mathcal{ML}} w_{ij} d_P^2(x_i, x_j) \Big) \end{aligned}$$

où $reg_1(\eta)$ permet de moduler entre la recherche de cohérence ou de consistance :

$$reg_1(\eta) = \frac{n^2 \eta}{(1-\eta) m}$$

avec $\eta \in [0..1]$ un paramètre associé à la pondération de chaque terme. Les facteurs n^2 et $m = |\mathcal{ML} \cup \mathcal{CL}|$ permettent d'avoir des ordres de grandeurs comparables entre les termes de cohérence et de consistance.

Le problème d'optimisation consiste alors à maximiser la variance des individus projetés en respectant la consistance sur les contraintes $\mathcal{CL}$ et $\mathcal{ML}$ données :

$$\begin{aligned} \max_P \; & Q_{\text{BoC}}(P) \\ s.t. \quad & P^\top P = Id_s \end{aligned} \tag{3.34}$$

où les poids w permettent d'intégrer la performance de A, liant ainsi l'apprentissage de P^* à l'algorithme de *clustering* A. Avant de représenter plus en détail la résolution du problème d'optimisation, il est utile de rappeler quelques résultats notamment autour de l'ACP.

ACP. Soit $X \in \mathbb{R}^{n \times p}$ la représentation matricielle de $\mathcal{X}$ centrée, la matrice $X^\top X \in \mathbb{R}^{p \times p}$ représente la matrice de corrélations (ou covariances, selon la procédure de normalisation appliquée aux données) empirique entre les variables descriptives, attributs ou propriétés.

La variance dans l'espace d'origine est définie (dans le cas où les données sont centrées et réduites) par :

$$Variance(X) = \frac{1}{n} trace(X^\top X)$$

Ainsi, soit $X' = XP$ une nouvelle représentation de X, la variance des individus dans l'espace de projection, qui correspond exactement à l'expression optimale du critère Q_{ACP} devient :

$$Variance(X') = \frac{1}{n} trace(X'^\top X') = \frac{1}{n} trace((XP)^\top XP) = \frac{1}{n} trace(P^\top X^\top XP)$$

Ainsi, on peut remarquer que

$$\max_P \ Variance(X') \equiv \max_P \ trace(P^\top X^\top XP)$$

Dans ce contexte, on peut poser :

$$Q_{\text{ACP}}(P) = trace(P^\top X^\top XP)$$

L'intérêt de présenter le critère de l'ACP sous cette forme réside dans la résolution du problème d'optimisation. Soit $X^* = XP^*$, l'obtention de la représentation optimale passe par la recherche de la matrice de projection optimale P^* solution du problème :

$$\begin{array}{l} \max_P \ Q_{\text{ACP}}(P) \\ \quad s.t. \quad P^\top P \ = \ Id_s \end{array} \tag{3.35}$$

où la contrainte $P^\top P = Id_s$ est là pour garantir l'orthonormalité de P^* assurant $[\![rang(X^*) = s]\!]$. Ceci permet de garantir une indépendance entre les s nouveaux descripteurs caractérisant $\mathcal{X}$ au travers de X^*.

La résolution de ce problème d'optimisation convexe est un résultat bien connu de l'algèbre linéaire, les s colonnes de P^* sont les s vecteurs propres associés aux s plus grandes valeurs propres de la matrice des corrélations/covariances $X^\top X$.

Algorithme

Dans BoC, le problème global d'obtention du *clustering* optimal C^* découle ainsi d'un processus itératif comprenant :

1. la résolution du problème d'apprentissage de X^* ;
2. l'adaptation des poids par mesure du respect de la consistance de A sur X^*.

L'algorithme employé pour résoudre le premier problème (3.34) suit le principe de résolution de l'ACP. En effet le critère (à maximiser) Q_{BoC} associé à la recherche de X^* par l'intermédiaire de P^* peut être réécrit :

$$Q_{\text{BoC}}(P) = \sum_{x_i \in \mathcal{X}} d_P^2(0, x_i) - reg_2(\eta) \sum_{(x_i, x_j) \in \mathcal{ML} \cup \mathcal{CL}} W_{ij} d_P^2(x_i, x_j) \tag{3.36}$$

où $reg_2(\eta) = \frac{1}{2n} reg_1(\eta)$ et avec :

$$W_{ij} = \begin{cases} -w_{ij} & \forall (x_i, x_j) \in \mathcal{CL} \\ w_{ij} & \forall (x_i, x_j) \in \mathcal{ML} \end{cases} \tag{3.37}$$

Soit $X \in \mathbb{R}^{n \times p}$ la représentation matricielle des données, et soient

- $[\mathcal{ML} \cup \mathcal{CL}]$ une représentation tabulaire indicée par l de l'ensemble $\mathcal{ML} \cup \mathcal{CL}$. $[\mathcal{ML} \cup \mathcal{CL}]_l$ est le l-ième couple $(x_i, x_j) \in \mathcal{ML} \cup \mathcal{CL}$ correspondant à une contrainte à satisfaire ;
- $Y^+, Y^- \in \mathbb{R}^{m \times p}$, les matrices telles que :

$$\begin{array}{rcll} Y^+_{l:} & = & (reg_2(\eta)|W_{ij}|)^{\frac{1}{2}}(x_i - x_j) & \text{avec } (x_i, x_j) = [\mathcal{ML} \cup \mathcal{CL}]_l \\ Y^-_{l:} & = & sign(W_{ij})(reg_2(\eta)|W_{ij}|)^{\frac{1}{2}}(x_i - x_j) & \text{avec } (x_i, x_j) = [\mathcal{ML} \cup \mathcal{CL}]_l \end{array}$$

$Y^+_{l:}$ et $Y^-_{l:}$ correspondent respectivement aux l-ièmes lignes des matrices Y^+ et Y^- représentant la différence régularisée entre les vecteurs x_i et x_j tels que le couple (x_i, x_j) constitue la l-ième contrainte ($\mathcal{ML}$ ou $\mathcal{CL}$).

Soient $Y^{+\prime} = Y^+ P$ et $Y^{-\prime} = Y^- P$, le critère Q_{BoC} peut alors être réécrit sous forme matricielle par :

$$\begin{array}{rcl} Q_{\text{BoC}}(P) & = & trace(X'^\top X') - trace(Y^{+\prime\top} Y^{-\prime}) \\ & = & trace(P^\top X^\top X P) - trace(P^\top Y^{+\top} Y^- P) \\ & = & trace(P(X^\top X - Y^{+\top} Y^-) P^\top) \end{array}$$

Le problème d'optimisation (3.34) se résout alors comme dans le cadre de l'ACP en diagonalisant la matrice $M = X^\top X - Y^{+\top} Y^-$. Le sous-espace optimal P^* correspond alors aux s vecteurs propres associées aux s valeurs propres les plus grandes de cette matrice. L'algorithme A est ensuite appliqué sur X^* de sorte à proposer une hypothèse de *clustering* H définie sur toutes les paires d'individus :

$$H_{ij} = \begin{cases} 1 & \text{si } Link(x_i, x_j, A) \\ -1 & \text{si } \overline{Link}(x_i, x_j, A) \end{cases} \tag{3.38}$$

où par défaut, $H_{ii} = 1$.

Le second problème à résoudre est l'intégration de la performance de A sur X^*. Celle-ci est réalisée en modifiant la distribution des poids w, modifiant ainsi les paramètres du premier problème pour une résolution ultérieure. Les poids sont ré-estimées de manière heuristique en suivant les principes du *boosting*, dans le sens où si A ne parvient pas à regrouper x_i et x_j tel que $(x_i, x_j) \in ML$ (respectivement $(x_i, x_j) \in CL$) alors les poids w_{ij} du couple correspondant (x_i, x_j) doivent croître (respectivement décroître). Cette adaptation doit inciter A à s'améliorer sur le *clustering* concernant ces paires d'individus, en lui proposant une représentation X^* adéquat. Dans un premier temps, l'erreur ϵ de A est calculée comme la proportion de contraintes $\mathcal{ML}$ et $\mathcal{CL}$ violées :

$$\epsilon = \frac{\overline{m}}{m} \tag{3.39}$$

où $\overline{m}$ est le nombre de contraintes non satisfaites par A.

À partir de cette erreur, une confiance α est alors associée au *clustering* produit par A :

$$\alpha = \frac{1}{2} \ln \left(\frac{1 - \epsilon}{\epsilon} \right) \tag{3.40}$$

Soit la matrice E correspondant aux hypothèses attendues, définie par :

$$E_{ij} = \begin{cases} 1 & \forall (x_i, x_j) \in \mathcal{ML} \\ -1 & \forall (x_i, x_j) \in \mathcal{CL} \\ 0 & \forall (x_i, x_j) \in \mathcal{X} \setminus (\mathcal{ML} \cup \mathcal{CL}) \end{cases}$$

Les poids sont finalement mis à jour de façon à respecter le principe de *boosting* :

$$w^*_{ij} = w_{ij} \frac{e^{-\alpha_{ij} E_{ij} H_{ij}}}{Z} \; \forall (x_i, x_j) \in \mathcal{ML} \cup \mathcal{CL} \tag{3.41}$$

où Z est un facteur de normalisation. Les poids sont alors augmentés si $E_{ij} \neq H_{ij}$ ce qui correspond à une erreur de clustering par A vis-à-vis des contraintes données.

On remarque qu'une erreur $\epsilon \geq \frac{1}{2}$ implique une confiance $\alpha \leq 0$ causant alors un échec vis à vis de l'objectif visé. L'algorithme de *clustering* n'est alors plus capable de satisfaire globalement les contraintes $\mathcal{ML}$ et $\mathcal{CL}$. On dit dans ce contexte que A ne remplit plus la condition d'être un classifieur non supervisé faible, et qu'il n'est plus raisonnable de le *booster*. Dans ce cas l'algorithme BoC s'arrête et une synthèse des différentes hypothèses obtenues est réalisée par un vote à la majorité, pour donner le *clustering* final des individus C.

Algorithme 24 BoC

Entrées : $\mathcal{X}$, n_k, $\mathcal{ML}$, $\mathcal{CL}$, t_f, A
Sorties : $C = \{C_1, ..., C_{n_k}\}$, X^*, P^*
1 : Initialisation des $w_{ij} = \frac{1}{m} \; \forall (x_i, x_j) \in \mathcal{ML} \cup \mathcal{CL}$ et $t = 0$
2 : Calculer P^* en résolvant (3.34) et déterminer $X^* = PX$
3 : Appliquer A sur X^*
4 : Mesurer ϵ par (3.39) et α par (3.40)
5 : Mise à jour de w par (3.41)
6 : Si $t < t_f$ ou $\epsilon < \frac{1}{2}$ alors aller en **2**
7 : $T = min\ (t, t_f)$
8 : $C = Vote(\{H^{(t)}\}_{1 \leq t \leq T})$

Construction de la partition finale

Le vote à la majorité permettant d'obtenir C, à partir de l'ensemble $\{H^{(t)}\}_{1 \leq t \leq T}$ des hypothèses de clustering sur les paires d'individus, peut être réalisé de différentes façons :

1. Selon le *boosting*, l'hypothèse finale, ici C, peut être construite à partir d'une combinaison linéaire H^* des différentes hypothèses apprises au cours du méta-algorithme. H^* est alors défini par :

$$H^*_{ij} = \sum_{t=1}^{T} \alpha^{(t)} H^{(t)}$$

Une matrice C de *clustering* peut alors être construite en observant la signature de la matrice H :

$$C_{ij} = \begin{cases} 1 & \text{si } H^*_{ij} > 0 \\ 0 & \text{si } H^*_{ij} < 0 \end{cases} \tag{3.42}$$

Néanmoins il n'est pas garantit que la matrice C ainsi définit corresponde effectivement à un *clustering*. Si l'on interprète C tel un graphe, une approche par partitionnement de graphe (comme SC) peut être employée pour couper un nombre minimum d'arêtes afin de

constituer n_k composantes connexes, puis une complétion en clique de ces composantes connexes nous permet d'obtenir une matrice C composée de n_k blocs de 1, correspondant davantage à un *clustering*. Une autre façon de procéder serait de considérer H^* comme une matrice de similarité et de l'utiliser comme telle si A est applicable sur une matrice similarité, ou d'en dériver une distance en considérant H^* comme une matrice de produit scalaire, et ensuite appliquer A en considérant cette distance.

2. Selon le même genre de principe de vote, mais en utilisant les divers paramètres appris lors de l'algorithme, il est possible d'estimer de nouveaux poids $\tilde{w}$ reflétant les différentes étapes du *boosting*.
$$\tilde{w}_{ij} = \sum_{t=1}^{T} \alpha^{(t)} w_{ij}^{(t)}$$
Les poids $\tilde{w}$ correspondent à une moyenne pondérée des poids utilisées lors de la génération successive des différentes représentation optimales. Ces poids permettent alors de résoudre (3.34) (où $w = \tilde{w}$) afin de trouver une nouvelle représentation X^* sur laquelle appliquer A pour déterminer C.
3. Une troisième piste envisagée pour produire C par un consensus entre les différents résultats de chaque étape du processus de *boosting* est de directement concaténer les différentes représentations des individus en pondérant chacune d'elle par l'efficacité qu'elle apporte en terme de *clustering*. Ce qui nous intéresse étant la distance entre les individus, ce type de fusion revient à réaliser une moyenne pondérée par α des distances entre individus décrits par les représentations optimales respectives :
$$d^2(x_i, x_j) = \sum_{t=1}^{T} \alpha^{(t)} (x_i - x_j)^\top P^{(t)^\top} P^{(t)} (x_i - x_j)$$
L'algorithme A est alors appliqué en utilisant d comme mesure de distance. Pour les algorithmes se fondant sur une mesure de similarité, un noyau peut être appris de manière similaire.

Discussion

L'approche proposée est très proche dans l'esprit de BC mais diverge sur plusieurs aspects. Tout d'abord, les deux approches se proposent de *booster* l'algorithme A en intégrant une mesure de la performance de A pour le calcul d'un espace de représentation optimal X^*. L'objectif de la discussion suivante est de traiter les similitudes et les différences entre ces deux approches. Dans un premier temps, l'intégration de la performance de A est traitée, puis dans un second temps, les détails du calcul du sous espace optimal sont développés.

Intégration de la performance de A

L'intégration de la performance de A est réalisée par l'intermédiaire des poids w. Ces poids sont mis à jour de façon différentes dans les deux approches. Dans BC, les poids w (noté brièvement w_{BC}) sont normalisés indépendamment selon le type de contraintes $\mathcal{ML}$ ou $\mathcal{CL}$, alors qu'ils sont normalisés relativement à l'ensemble des contraintes dans le cas de BoC (w_{BoC}). En particulier, soit W_{BC} et W_{BoC} les matrices des poids correspondants aux approches, l'initialisation est différente :
$$W_{\text{BC}ij} = \begin{cases} \dfrac{1}{m^+} & \forall (x_i, x_j) \in \mathcal{ML} \\ -\dfrac{1}{m^-} & \forall (x_i, x_j) \in \mathcal{CL} \end{cases} \tag{3.43}$$

$$W_{\text{BoC}ij} = \begin{cases} \dfrac{1}{m} & \forall (x_i, x_j) \in \mathcal{ML} \\ -\dfrac{1}{m} & \forall (x_i, x_j) \in \mathcal{CL} \end{cases} \tag{3.44}$$

Soit la mise à jour des poids de BC (3.26) :

$$W^{(t)}_{\text{BC}ij} = \begin{cases} \dfrac{e^{-K^{(t)}_{ij}}}{Z_{\mathcal{ML}}} & \forall (x_i, x_j) \in \mathcal{ML} \\ -\dfrac{e^{K^{(t)}_{ij}}}{Z_{\mathcal{CL}}} & \forall (x_i, x_j) \in \mathcal{CL} \end{cases} \tag{3.45}$$

Si on utilise le fait que K est construite durant le processus itératif par l'équation :

$$K^{(t)} = K^{(t-1)} + \alpha^{(t)} H^{(t)}$$

alors le calcul des poids se réécrit :

$$W^{(t)}_{\text{BC}ij} = \begin{cases} \dfrac{e^{-K^{(t-1)}_{ij} - \alpha^{(t)} H^{(t)}_{ij}}}{Z_{\mathcal{ML}}} = W^{(t-1)}_{\text{BC}_{ij}} \dfrac{e^{-\alpha^{(t)} H^{(t)}_{ij}}}{Z_{\mathcal{ML}}} & \forall (x_i, x_j) \in \mathcal{ML} \\ -\dfrac{e^{K^{(t-1)}_{ij} + \alpha^{(t)} H^{(t)}_{ij}}}{Z_{\mathcal{CL}}} = W^{(t-1)}_{\text{BC}_{ij}} \dfrac{e^{\alpha^{(t)} H^{(t)}_{ij}}}{Z_{\mathcal{ML}}} & \forall (x_i, x_j) \in \mathcal{CL} \end{cases} \tag{3.46}$$

Sous cette forme la mise à jour des poids de BC est très similaire à celle de BoC, dans la mesure où les hypothèses H_{ij} sont à valeurs dans $\{0, 1\}$ pour BC et dans $\{-1, 1\}$ pour BoC. En particulier :

- Pour une contrainte $\mathcal{ML}$ non violée, *i.e.* $(x_i, x_j) \in \mathcal{ML}$ et $H_{ij} = 1$, le poids associé $W_{\text{BC}ij}$ diminue, ce qui entraîne par la normalisation, une augmentation de la valeur des poids associés aux contraintes $\mathcal{ML}$ respectées.
- Pour une contrainte $\mathcal{CL}$ violée, *i.e.* $(x_i, x_j) \in \mathcal{CL}$ et $H_{ij} = 1$, le poids associé $W_{\text{BC}ij}$ augmente directement (dans les négatifs, car $W_{\text{BC}ij} < 0 \ \forall (x_i, x_j) \in \mathcal{CL}$), entraînant par la normalisation, une diminution de la valeur des poids associés aux contraintes $\mathcal{CL}$ respectées.

Seule diffère l'expression de la confiance $\alpha^{(t)}$ (équation (3.30) dans BC et (3.40)), mais elle reste dans les deux cas une mesure relative à l'erreur de A dans la satisfaction des contraintes $\mathcal{CL}$ et $\mathcal{ML}$. Cette erreur est explicite dans BoC mais non dans BC.

Calcul de la représentation optimale X^*

Le second point important des approches BC et BoC est la génération d'une nouvelle représentation consciente des lacunes de A sur le respect des contraintes. Cette nouvelle représentation vise à améliorer globalement les performances de A. Les deux approches visent à diagonaliser une matrice de corrélations mais c'est sur le calcul de cette corrélation qu'elle diffère : $X^\top W_{\text{BC}} X$ pour BC, et $X^\top X - {Y^+}^\top Y^-$ pour BoC. Les critères objectifs associés aux recherche des sous-espaces de projections optimaux respectifs sont :

pour BC : $trace(P^\top X^\top W_{\text{BC}} X P)$

pour BoC : $trace(P^\top (X^\top X - {Y^+}^\top Y^-) P)$

Soit $X' = XP$ et $\langle x_i, x_j \rangle_P$ le produit scalaire entre x_i et x_j projetés dans P. Le critère Q_{BC} peut être réécrit pour dégager une similitude forte avec la recherche de consistance réalisée par BoC :

$$\begin{aligned} Q_{\text{BC}}(P) &= trace(P^\top X^\top W_{\text{BC}} X P) \\ &= trace(X'^\top W_{\text{BC}} X) \\ &= \sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} \langle x_i, x_j \rangle_P \\ &= \frac{1}{2} \sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} (\langle x_i, x_i \rangle_P + \langle x_j, x_j \rangle_P - d_P^2(x_i, x_j)) \end{aligned}$$

Ce critère est équivalent en maximisation à :

$$\begin{aligned} Q_{\text{BC}}(P) &= \sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} (\langle x_i, x_i \rangle_P + \langle x_j, x_j \rangle_P - d_P^2(x_i, x_j)) \\ &= \sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} (\langle x_i, x_i \rangle_P + \langle x_j, x_j \rangle_P) - \sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} d_P^2(x_i, x_j) \end{aligned}$$

Comme $Q_{\text{CST}}(P) = -\sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BoC}ij} d_P^2(x_i, x_j)$ et après l'analogie constatée entre W_{BC} et W_{BoC}, on peut réécrire :

$$Q_{\text{BC}}(P) \approx \sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} (\langle x_i, x_i \rangle_P + \langle x_j, x_j \rangle_P) + Q_{\text{CST}}(P)$$

De cette façon, on peut rapprocher les deux objectifs en constatant :

$$\max_P Q_{\text{BC}}(P) \approx \max_P Q_{\text{CST}}(P)$$

Il reste alors l'expression :

$$\sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} (\langle x_i, x_i \rangle_P + \langle x_j, x_j \rangle_P) = \sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} (||x_i P||_2^2 + ||x_j P||_2^2)$$

qui reste difficile à interpréter. En particulier, comme :

$$\begin{aligned} &\sum_{(x_i,x_j)\in\mathcal{X}^2} W_{\text{BC}ij} (||x_i P||_2^2 + ||x_j P||_2^2) \\ &= \sum_{(x_i,x_j)\in\mathcal{ML}} w_{\text{BC}ij} (||x_i P||_2^2 + ||x_j P||_2^2) + \sum_{(x_i,x_j)\in\mathcal{CL}} w_{\text{BC}ij} (||x_i P||_2^2 + ||x_j P||_2^2) \end{aligned}$$

alors plus le poids associé à une contrainte $\mathcal{ML}$ augmentera ($w_{\text{BC}} > 0$), plus P sera tel que les normes des individus impliqués dans ces contraintes soient préservées dans la nouvelle représentation. Plus le poids associé à une contrainte $\mathcal{CL}$ augmentera dans les négatifs ($w_{\text{BC}} < 0$), plus P sera tel que la somme des normes des individus impliqués dans ces contraintes soient minimisée, ce qui intuitivement revient à les rapprocher et est contradictoire avec l'objectif.

L'approche proposée permet à l'image de BC, de calculer une représentation X^* à chaque étape, optimale pour des valeurs de poids fixés. BoC propose différentes façons de produire une hypothèse finale H^* interprétable comme un *clustering* des données C, et celles-ci seront discutés dans la section des expérimentations. Cependant le facteur limitant de la contribution BoC est le problème de la convergence et l'arbitraire de l'intégration de la performance de A. Dans la perspective de palier à ce problème, les approches UzaBoC et AdaUzaBoC, fondées sur des techniques d'optimisation numérique, ont été développées et éprouvées empiriquement. Leurs descriptions détaillées font l'objet de la prochaine section.

3.7.3 UZABOC et ADAUZABOC : *boosting* simple et adaptatif de *clustering* par optimisation

Le critère de l'ACP utilisé par BOC est indépendant de l'intégration de la performance de A. De ce fait, UZABOC se fonde sur le même critère pour modéliser la cohérence. En revanche, l'approche suggère d'intégrer la mesure de performance de A par l'intermédiaire de contraintes au problème d'optimisation posé simplement par la recherche de cohénrece. Ainsi, en conservant l'hypothèse de BOC *i.e.* la volonté de rapprocher des individus impliqués dans une contrainte $\mathcal{ML}$ et de tenir éloignés des individus impliqués dans une contrainte $\mathcal{CL}$, les hypothèses suivantes sont émises :

- si $(x_i, x_j) \in \mathcal{ML}$ alors il existe une constante $\xi_{ij} \geq 0$ la plus grande possible telle que ξ_{ij} borne supérieurement la distance entre x_i et x_j dans le sous-espace :

$$(x_i, x_j) \in \mathcal{ML} \Rightarrow \exists \xi_{ij} \geq 0,\ d_P^2(x_i, x_j) \leq \xi_{ij}$$

- si $(x_i, x_j) \in \mathcal{CL}$ alors il existe une constante $\xi_{ij} \geq 0$ la plus petite possible telle que ξ_{ij} borne inférieurement la distance entre x_i et x_j dans le sous-espace :

$$(x_i, x_j) \in \mathcal{CL} \Rightarrow \exists \xi_{ij} \geq 0,\ d_P^2(x_i, x_j) \geq \xi_{ij}$$

Objectif

L'intégration de ces hypothèses comme contraintes au problème de recherche de cohérence permet de formuler le problème d'optimisation suivant :

$$\begin{array}{rrcll} \max\limits_P Q_{\text{COH}}(P) & = & \max\limits_P trace(P^\top X^\top X P) & & \\ s.t. & P^\top P & = & Id_s & \\ & d_P^2(x_i, x_j) & \leq & \xi_{ij} & \forall (x_i, x_j) \in \mathcal{ML} \quad (cs1) \\ & d_P^2(x_i, x_j) & \geq & \xi_{ij} & \forall (x_i, x_j) \in \mathcal{CL} \quad (cs2) \end{array} \tag{3.47}$$

Chaque contrainte $\mathcal{ML}$ ou $\mathcal{CL}$ est associée à une contrainte d'optimisation $(cs1)$ ou $(cs2)$. Résoudre ce problème pour obtenir une représentation optimale P^* tel qu'il est posé ne permet à aucun moment d'intégrer le retour de A sur la génération de X^*. L'idée pour résoudre ce problème est de se servir de ξ pour rendre compte de la performance de A. Si A appliqué à $X^* = XP^*$ ne parvient pas à satisfaire les contraintes $\mathcal{ML}$ et $\mathcal{CL}$ alors que les contraintes d'optimisation $(cs1)$ et $(cs2)$ sont satisfaites, ces dernières ne sont pas suffisamment adaptées. Dans ce cas, la solution P^* n'est pas adaptée, et les bornes ξ_{ij} correspondantes doivent être réévaluées afin de restreindre l'espace des solutions réalisables. Cela permet, à la suite d'une nouvelle optimisation, d'améliorer les chances d'obtenir un optimum P^* adapté aux contraintes.

Algorithme

L'algorithme développé (dont la trame est exposée figure 3.4) pour résoudre le problème de la recherche de la représentation permettant le respect au mieux des contraintes $\mathcal{ML}$ et $\mathcal{CL}$ par A, se décline en différents sous problèmes :

- la recherche d'une représentation optimale par résolution de (3.47) ;
- l'intégration du retour de A pour tendre vers une adéquation entre l'algorithme de *clustering* et la représentation optimale.

En supposant connues les valeurs de ξ pour toutes les contraintes, le problème (3.47) peut être résolu grâce à l'optimisation lagrangienne. La contrainte d'optimisation $[\![P^\top P = Id_s]\!]$ peut être décomposée en s contraintes d'optimisation, en constatant à la fois :

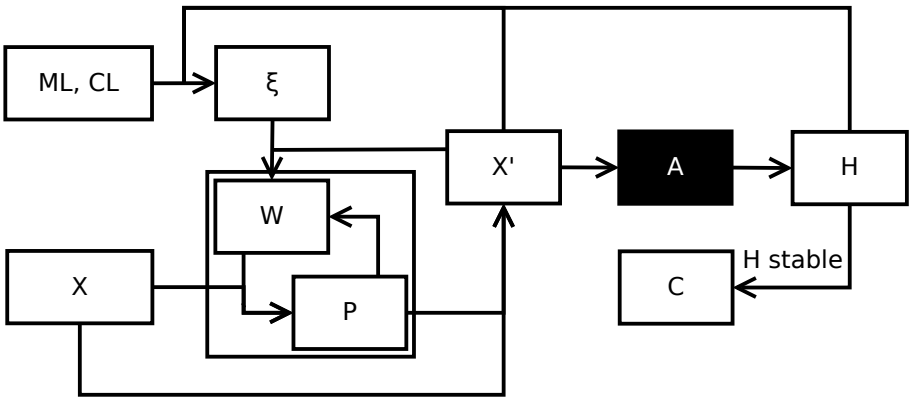

FIGURE 3.4 — Schéma du déroulement d'UZABOC.

- $P^\top P = Id_s \Leftrightarrow (P^\top P)^2 = Id_s$;
- $(P^\top P)^\top_{:i}(P^\top P)_{:i} = 1 \Leftrightarrow P^\top P = Id_s$.

Le lagrangien associé $\mathcal{L}(P, w, \lambda)$ est donné par la formule :

$$\begin{aligned} &\mathcal{L}(P, w, \lambda) \\ &= trace(P^\top X^\top X P) - \lambda^\top diag((P^T P)^2 - Id_s) \\ &\quad - \sum_{(x_i,x_j)\in\mathcal{ML}} w_{ij}\left(d^2_P(x_i, x_j) - \xi_{ij}\right) + \sum_{(x_i,x_j)\in\mathcal{CL}} w_{ij}\left(d^2_P(x_i, x_j) - \xi_{ij}\right) \end{aligned} \tag{3.48}$$

où $diag(M)$ est le vecteur constitué des éléments diagonaux de M.
$w = \{w_{ij}\}_{\substack{i\in[1..n]\\ j\in[1..n]}}$ et $\lambda^\top = (\lambda_1, \ldots, \lambda_s)$ représentent les multiplicateurs de lagrange. En particulier, les multiplicateurs de lagrange w sont analogues aux poids w du critère de BOC : Q_{BOC}. En posant W telle que :

$$W_{ij} = \begin{cases} -w_{ij} & \forall (x_i, x_j) \in \mathcal{CL} \\ w_{ij} & \forall (x_i, x_j) \in \mathcal{ML} \end{cases} \tag{3.49}$$

En reprennant la notation de BOC ($X \in \mathbb{R}^{n\times p}$ la matrice des données,et $[\mathcal{ML} \cup \mathcal{CL}]$ la représentation tabulaire indicée par l de l'ensemble $\mathcal{ML} \cup \mathcal{CL}$) et en réintroduisant les matrices $Y^+ \in \mathbb{R}^{m\times p}$ et $Y^- \in \mathbb{R}^{m\times p}$ les matrices telles que :

$$\begin{aligned} Y^+_{l:} &= |W_{ij}|^{\frac{1}{2}}(x_i - x_j) && \text{si } (x_i, x_j) = [\mathcal{ML} \cup \mathcal{CL}]_l \\ Y^-_{l:} &= sign(W_{ij})|W_{ij}|^{\frac{1}{2}}(x_i - x_j) && \text{si } (x_i, x_j) = [\mathcal{ML} \cup \mathcal{CL}]_l \end{aligned}$$

$Y^+_{l:}$ et $Y^-_{l:}$ correspondent respectivement aux l-ièmes lignes des matrices Y^+ et Y^- représentant la différence pondérée entre les vecteurs x_i et x_j tels que le couple (x_i, x_j) constitue la l-ième contrainte ($\mathcal{ML}$ ou $\mathcal{CL}$).

Le lagrangien peut être reformulé :

$$\begin{aligned} &\mathcal{L}(P, w, \lambda) \\ &= trace(P^\top(X^\top X - {Y^+}^\top Y^-)P) - \lambda^\top diag((P^T P)^2 - Id_s) \\ &\quad - \sum_{(x_i,x_j)\in\mathcal{ML}} W_{ij}\xi_{ij} - \sum_{(x_i,x_j)\in\mathcal{CL}} W_{ij}\xi_{ij} \end{aligned} \tag{3.50}$$

Si P^* est un optimum de 3.47, alors il existe un unique couple (w^*, λ^*) tel que P^*, W^* et λ^* satisfont les conditions du premier ordre (CPO) suivantes:

$$\begin{cases} \nabla_{P_i^*}\mathcal{L}(P^*, w^*, \lambda^*) & = & \mathbf{0} & (cond\ 1) \\ \dfrac{\partial\mathcal{L}(P^*, w^*, \lambda^*)}{\partial_{w_{ij}^*}} & = & 0 & (cond\ 2) \\ \nabla_{\lambda^*}\mathcal{L}(P^*, w^*, \lambda^*) & = & \mathbf{0} & (cond\ 3) \end{cases}$$

Les différentes dérivées partielles dans $(cond\ 1)$, $(cond\ 2)$ et $(cond\ 3)$ mènent respectivement aux expressions:

$$\begin{aligned} \nabla_{P_i^*}\mathcal{L}(P^*, w^*, \lambda^*) &= 2(X^\top X - Y^{+\top}Y^-)(P_{:i}^*) - 2\lambda_i P^{*\top}P^*(P_{:i}^*) \\ \frac{\partial\mathcal{L}(P^*, w^*, \lambda^*)}{\partial_{w_{ij}^*}} &= \begin{cases} \xi_{ij} - d^2_{P^*}(x_i, x_j) & \forall(x_i, x_j) \in \mathcal{ML} \\ d^2_{P^*}(x_i, x_j) - \xi_{ij} & \forall(x_i, x_j) \in \mathcal{CL} \end{cases} \\ \nabla_{\lambda^*}\mathcal{L}(P^*, w^*, \lambda^*) &= P^{*\top}P^* - Id_s \end{aligned}$$

Si on étudie alors les différentes conditions du premier ordre, on remarque que :

- Sous réserve de connaître les valeurs des multiplicateurs de lagrange w (et en utilisant $(cond\ 3)$), la satisfaction de $(cond\ 1)$ traduit le fait que P^* correspond exactement à la solution optimale de l'ACP où la matrice de corrélation correspondante aux données à approximer est la matrice $M = X^\top X - Y^{+\top}Y^-$. La matrice de rang s approximant le mieux cette matrice corrélation s'obtient par diagonalisation et sélection des s vecteurs propres de M correspondants aux s valeurs propres les plus grandes.
- une expression sous forme close de w_{ij}^* ne peut être déterminée analytiquement pour garantir la satisfaction de $(cond\ 2)$ car $\forall(x_i, x_j) \in \mathcal{ML} \cup \mathcal{CL}$, $d^2_{P^*}(x_i, x_j)$ dépend de W_{ij}^*.

Ces observations suggèrent une procédure algorithmique afin d'isoler les recherches de P^* et de w^*. L'idée est de proposer un moyen d'approcher de manière itérative, au travers d'une suite les multiplicateurs de lagrange w^* optimaux, et P^*, par observation respectivement d'un sous espace P courant et de multiplicateurs w courants. étant donnés l'observation d'un sous-espace de projection P fixé. L'approche UZABOC se fonde alors sur l'algorithme d'Uzawa adapté à l'optimisation numérique d'un critère objectif sous contraintes pour lesquels les multiplicateurs de lagrange ne peuvent être déterminés par une expression close. L'algorithme d'Uzawa propose de construire une suite $(W^{(t)})_t$ convergente vers W^*. À chaque valeur $W^{(t)}$ connue, un sous espace optimal $P^{(t)}$ est obtenu directement par diagonalisation.

Calcul de la nouvelle représentation X^*. Le calcul de la nouvelle représentation optimale X^* est réalisé par une projection linéaire de X sur P^* :

$$X^* = XP^*$$

où P^* est obtenu comme la limite de la suite $(P^{(t)})_t$ issue de la résolution itérative (par Uzawa) du système émanant des conditions KKT, permettant d'obtenir également les multiplicateurs optimaux w^*. Partant d'une initialisation nulle des multiplicateurs $w = 0$, la mise à jour de P, pour w fixé, est déterminée par :

$$\begin{aligned} P^{(t)} &= \underset{P}{arg\,max}\ trace(P^\top(X^\top X - Y^{+\top}Y^-)P) \\ s.t. \quad P^\top P &= Id_s \end{aligned} \tag{3.51}$$

Les multiplicateurs de lagrange w sont eux, mis à jour par :

$$w_{ij}^{(t)} = \begin{cases} max\left(0, w_{ij}^{(t-1)} + \rho \times (d^2_{P^{(t)}}(x_i, x_j) - \xi_{ij})\right) & \forall (x_i, x_j) \in \mathcal{ML} \\ max\left(0, w_{ij}^{(t-1)} + \rho \times (\xi_{ij} - d^2_{P^{(t)}}(x_i, x_j))\right) & \forall (x_i, x_j) \in \mathcal{CL} \end{cases} \tag{3.52}$$

où ρ est un pas d'optimisation fixé à l'avance, et paramétrable mais constant, dans le cas de l'application d'Uzawa.

Intégration de la performance de A. Une fois le couple optimal (P^*, w^*) approché par Uzawa, la représentation optimale X^* est calculée et A est appliqué sur X^*. En cas d'erreurs sur la satisfaction des contraintes $\mathcal{ML}$ et $\mathcal{CL}$ par A un nouvel espace de représentation doit être déterminé. La règle de mise à jour (3.52) donne une indication sur le moyen de contrôler les mises à jour de w pour corriger la recherche d'un nouvel X^* (par la recherche d'un nouveau couple (P^*, w^*)) permettant à A de mieux satisfaire les contraintes $\mathcal{ML}$ et $\mathcal{CL}$. Ainsi, UZABOC propose d'influer directement sur les bornes ξ_{ij} des contraintes d'optimisation $(cs1)$ et $(cs2)$.

Soit H la matrice des hypothèses de *clusterings* issues de l'application de A sur X^* :

$$H_{ij} = \begin{cases} 1 & \text{si } Link(x_i, x_j, A) \\ -1 & \text{si } \overline{Link}(x_i, x_j, A) \end{cases} \tag{3.53}$$

Deux cas peuvent se produire pour chacun des types de contraintes $\mathcal{ML}$ et $\mathcal{CL}$ lorsqu'elles ne sont pas satisfaites :

- Soit $(x_i, x_j) \in \mathcal{ML}$ et $\overline{Link}(x_i, x_j, A)$ (la contrainte $\mathcal{ML}$ n'est pas respectée) :
  - si la contrainte $(cs1)$ n'est pas satisfaite, alors les multiplicateurs de lagrange augmentent naturellement, imposant ainsi un poids plus fort sur le couple (x_i, x_j) lors de la recherche de la prochaine représentation optimale ;
  - si la contrainte $(cs1)$ est satisfaite, alors les multiplicateurs de lagrange devraient naturellement diminuer, or l'objectif étant de le faire augmenter car la contrainte $\mathcal{ML}$ associée est violée. Nous proposons d'exercer un contrôle en durcissant la contrainte d'optimisation $(cs1)$, en diminuant la valeur de ξ_{ij}. Ainsi la diminution naturelle des poids est amortie et la difficulté de satisfaire la contrainte d'optimisation $(cs1)$ ultérieurement est accrue.
- Soit $(x_i, x_j) \in \mathcal{CL}$ et $Link(x_i, x_j, A)$ (la contrainte $\mathcal{CL}$ n'est pas respectée) :
  - si la contrainte $(cs2)$ n'est pas satisfaite, alors les multiplicateurs de lagrange augmentent naturellement, imposant ainsi un poids plus fort sur (x_i, x_j) lors de la recherche de la prochaine représentation optimale ;
  - si la contrainte $(cs2)$ est satisfaite, alors pour amortir la diminution naturelle des multiplicateurs, on propose d'adapter cette contrainte d'optimisation en augmentant la valeur de ξ_{ij}.
- Dans tous les autres cas, si les contraintes $\mathcal{CL}$ et $\mathcal{ML}$ sont satisfaites, les paramètres ξ correspondant sont suffisants et n'ont pas besoin d'être réévalués. De plus, les poids diminuent également naturellement jusqu'à devenir éventuellement nuls.

Ce principe de contrôle des mises à jour des multiplicateurs de lagrange est donc réalisée par une adaptation au préalable des paramètres ξ. Ainsi, partant d'une initialisation des ξ_{ij} tels que les contraintes d'optimisation $(cs1)$ et $(cs2)$ soient infalsifiables, une suite convergente $(\xi_{ij}^{(t)})_t$ est

construite de manière heuristique par :

$$\xi_{ij}^{(t)} = \begin{cases} \dfrac{d_P^2(x_i, x_j)}{2} & \forall (x_i, x_j) \in \mathcal{ML}, \overline{Link}(x_i, x_j, A) \wedge cs1(x_i, x_j) \\ \\ \dfrac{(d_P^2(x_i, x_j) + d^2(x_i, x_j))}{2} & \forall (x_i, x_j) \in \mathcal{CL}, Link(x_i, x_j, A) \wedge cs2(x_i, x_j) \end{cases} \tag{3.54}$$

où $cs1(x_i, x_j)$ indique que la contrainte d'optimisation $(cs1)$ est satisfaite pour le couple (x_i, x_j) (*idem* pour $(cs2)$).

Algorithme 25 UZABOC

ENTRÉES : $\mathcal{X}$, n_k, $\mathcal{ML}$, $\mathcal{CL}$, t_f
SORTIES : $C = \{C_1, ..., C_{n_k}\}$, X^*, P^*
1 : Initialisation des $w_{ij} = 0 \; \forall (x_i, x_j) \in \mathcal{CL} \cup \mathcal{ML}$
2 : Initialisation des $\xi_{ij} = 0 \; \forall (x_i, x_j) \in CL$ et $\xi_{ij} = d_P^2(x_i, x_j) \; \forall (x_i, x_j) \in \mathcal{ML}$
3 : $t = 0$. Calculer $P^{(t)}$ en résolvant (3.47) et déterminer $X^{(t)} = XP^{(t)}$
4 : Mise à jour des w_{ij} par (3.52)
5 : Si $\mathcal{L}(P^*, w*, \lambda^*)$ ne converge pas alors $t = t + 1$ aller en **3**. $X^* = X^{(t)}$ et $P^* = P^{(t)}$
6 : $C =$ Appliquer A sur X^*
7 : Mise à jour de ξ par (3.54)
8 : Si $t < t_f$ et UZABOC ne converge pas alors aller en **3**
9 : Si $t < t_f$ alors $t_f = t$

Discussion

L'algorithme UZABOC est relativement proche de BOC. Par une formalisation sous forme d'optimisation sous contraintes, on peut dégager une similitude forte entre les multiplicateurs de lagrange de UZABOC et les poids de BOC. L'avantage de UZABOC sur BOC est que l'adaptation des poids à la satisfaction des contraintes $\mathcal{ML}$ et $\mathcal{CL}$ par A est moins arbitraire, car reposant sur un algorithme d'optimisation numérique adapté.

Enfin un autre avantage de l'approche UZABOC est que la distribution naturelle des poids est apprise par l'algorithme d'optimisation de sorte que l'algorithme tend asymptotiquement à produire la meilleure (au sens du point-selle) représentation permettant de satisfaire cohérence et consistance selon l'algorithme A employé. Les approches fondées sur le *boosting* reposent quant à elles sur une combinaison linéaire d'hypothèses produites par la distribution des poids à chaque étape, normalisée et adaptée pour apprendre successivement des hypothèses indépendantes les unes des autres.

À travers cet aspect se règle également la question de la convergence. Là où les approches par *boosting* convergent difficilement vers une solution qui n'est pas le résultat attendu et nécessitent une procédure finale pour produire un *clustering* des individus en satisfaisant les contraintes, l'approche par optimisation cherche le sous-espace optimal réalisant un compromis entre le terme de cohérence représenté par l'objectif, et le terme de consistance représenté par le terme de pénalisation introduit dans le lagrangien. L'algorithme d'Uzawa cherche alors à approximer le point selle de ce lagrangien, correspondant intuitivement à une solution optimale P^* maximisant la part de cohérence et minimisant la part pénalisante associée à la consistance. Le point selle du lagrangien $\mathcal{L}(P^*, w^*, \lambda^*)$ est caractérisé par :

$$\mathcal{L}(P, w^*, \lambda^*) \leq \mathcal{L}(P^*, w^*, \lambda^*) \leq \mathcal{L}(P^*, w, \lambda) \tag{3.55}$$

Dans notre contexte, l'étape de calcul de P^* est associée également au calcul de λ^*, ainsi on ne peut garantir la maximisation de la borne inférieure du point selle. La mise à jour des multiplicateurs w permet de réduire la valeur de l'objectif du dual en adaptant les multiplicateurs au respect des contraintes d'optimisation. Ainsi pour des valeurs de ξ fixés, on ne peut garantir que l'algorithme UZABOC converge vers ce point selle s'il existe, mais nous pouvons alors observer empiriquement l'écart entre les valeurs des lagragien après mise à jour des différentes variables (P du primal, et λ et w du dual). La différence entre les deux bornes est appelée ici le saut de dualité, et celui-ci doit tendre vers 0 à mesure que les contraintes d'optimisation se stabilisent, caractérisant ainsi l'atteinte d'une solution optimale en dualité forte. Pour finir, l'approche UZABOC est globalement convergente, puisque les suites $(\xi_{ij}^{(t)})_t \ \forall (x_i, x_j) \in \mathcal{ML}$ sont décroissantes et minorées par 0, et les suites $(\xi_{ij}^{(t)})_t \ \forall (x_i, x_j) \in \mathcal{CL}$ sont croissantes et majorées par $d^2(x_i, x_j)$. L'approche converge alors vers une solution optimale lorsque le saut de dualité s'annule, et converge vers une solution sous-optimale en cas de dualité faible, solution pour laquelle un écart à l'optimum (un certificat) peut-être calculé. Ces différentes observations laissent entrevoir deux variantes, simple et adaptative, pour l'algorithme :

- la variante simple UZABOC consiste à approcher complètement le point selle du lagrangien pour chaque réévaluation des paramètres ξ;
- la variante adaptative ADAUZABOC consiste à approcher le point selle tout en adaptant pendant la recherche les valeurs de ξ modifiant ainsi en ligne les contraintes du problème (et la valeur du lagrangien) et réduisant ainsi l'espace des solutions qui leur est associé.

Ainsi, la variante simple (cf. algorithme 25), pour ξ fixé, applique complètement et jusqu'à convergence l'algorithme Uzawa pour obtenir un sous espace P^*. A est appliqué sur $X^* = XP^*$ et les erreurs de A sur le respect des contraintes $\mathcal{ML}$ et $\mathcal{CL}$ mettent à jour les paramètres ξ de manière à guider davantage la recherche d'une *meilleure* solution de *clustering*. Cette procédure est alors réappliquée avec les nouvelles valeurs de ξ.

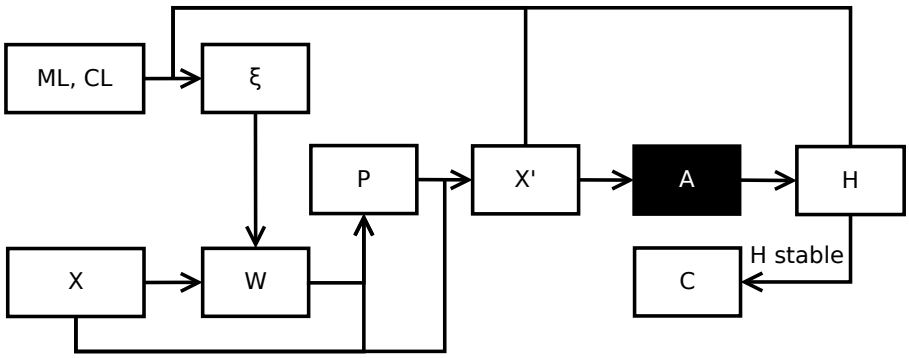

FIGURE 3.5 — Schéma du déroulement d'ADAUZABOC.

Partant d'une initialisation de ξ et des poids W, la variante adaptative (Fig. 3.5 et algorithme 26) recherche P^* en cherchant à améliorer la borne inférieure du problème de point selle (3.55) tout en réévaluant λ^*. A est ensuite appliqué sur $X^* = XP^*$ et ξ est mis à jour afin de tenir compte des erreurs de A sur $\mathcal{ML}$ et $\mathcal{CL}$. La mise à jour des poids W n'est alors plus exactement celle qui permet de réduire la borne supérieure du lagrangien, mais une nouvelle direction de mise à jour est considéré afin de tenir compte immédiatement du retour de A. Cette variante se comporte plus comme l'approche par *boosting*, dans la mesure où chaque itération permet

d'adapter la distribution des poids en insistant davantage sur les paires d'individus correspondant aux contraintes $\mathcal{ML}$ et $\mathcal{CL}$ non satisfaites. L'absence de normalisation de ces poids permet d'obtenir à la fin, une solution réalisant une adéquation entre l'intégration des contraintes $\mathcal{ML}$ et $\mathcal{CL}$ et leur satisfaction, et ainsi ne nécessite pas de procédure de vote à la majorité.

Algorithme 26 ADAUZABOC

ENTRÉES : $\mathcal{X}$, n_k, $\mathcal{ML}$, $\mathcal{CL}$, t_f
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialisation des $w_{ij} = 0 \; \forall (x_i, x_j) \in \mathcal{CL} \cup \mathcal{ML}$
2 : Initialisation des $\xi_{ij} = 0 \; \forall (x_i, x_j) \in CL$ et $\xi_{ij} = d_P^2(x_i, x_j) \; \forall (x_i, x_j) \in \mathcal{ML}$
3 : $t = 0$. Calculer $P^{(t)}$ en résolvant (3.47) et déterminer $X^{(t)} = XP^{(t)}$
4 : $C =$ Appliquer A sur $X^{(t)}$
5 : Mise à jour de ξ par (3.54)
6 : Mise à jour des w_{ij} par (3.52)
7 : Si $t < t_f$ et ADAUZABOC ne converge pas alors $t = t + 1$ et aller en **3**. $X^* = X^{(t)}$ et $P^* = P^{(t)}$
8 : Si $t < t_f$ alors $t_f = t$

Ces deux variantes sont illustrées dans la figure 3.6 pour la recherche d'une solution optimale. Elles seront discutées davantage dans l'évaluation empirique.

3.8 Évaluation

3.8.1 Données

Les jeux de données utilisés pour l'évaluation expérimentale des différentes contributions BOC, UZABOC et ADAUZABOC proviennent tous de la base UCI [1]. Il s'agit des jeux de données *Iris*, *Wine*, *Parkinson* et *WDBC*. Les caractéristiques principales de ces jeux de données sont résumés dans le tableau 3.1.

Jeu	Nb. Individus	Nb. Attributs	Nb. classes
Iris	150	4	3
Wine	178	13	3
Parkinson	195	22	2
WDBC	569	30	2

TABLEAU 3.1 — Caractéristiques des jeux de données utilisés pour le *clustering* semi-supervisé.

- Le jeu de donnée *Iris* correspond à un ensemble de 150 fleurs représentant 3 variétés d'iris présentes en quantités homogènes, soient 50 Iris par classe.
- Le jeu *Wine* correspond à différents vins d'Italie et sont représentés par leurs constituants chimique ou descripteurs sensoriels (taux d'acidité, alcool, magnésium, intensité de la couleur, *etc.*).
- *Parkinson* est un jeu de donnée dans lequel 195 enregistrements vocaux de 31 patients sont représentés par des descripteurs numériques issus de techniques de traitement du signal (fréquence fondamentale minimum, maximum, moyenne, mesures de variation d'amplitude, *etc.*).

1. *http://archive.ics.uci.edu/ml/*

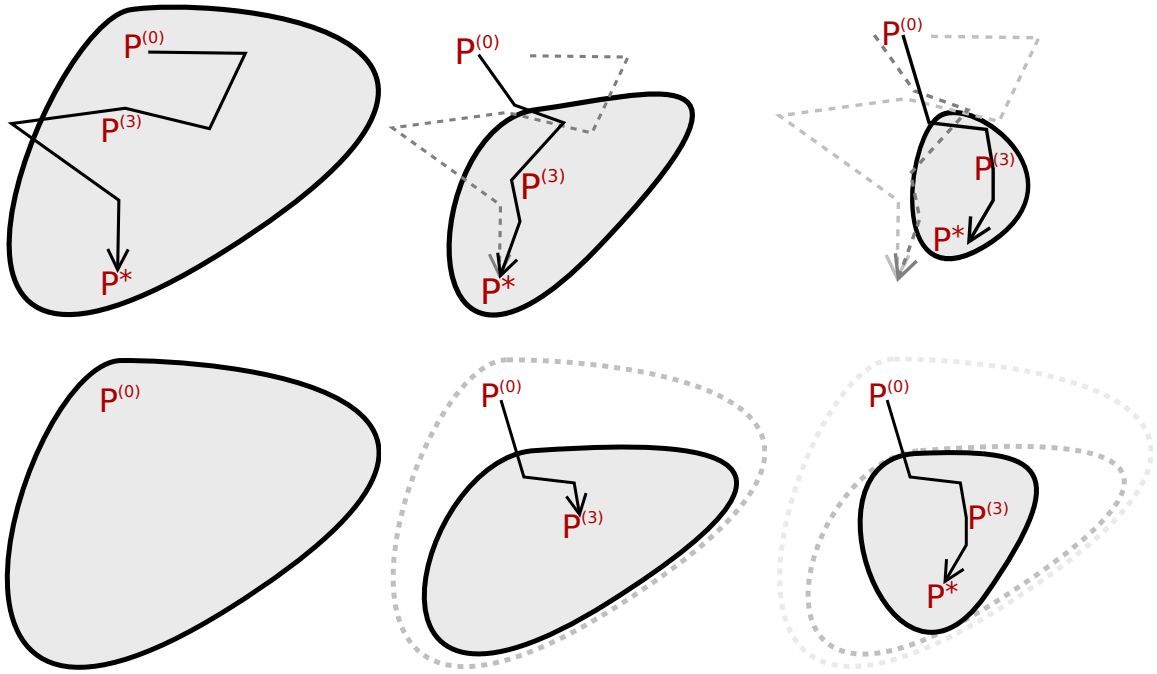

FIGURE 3.6 — Illustration des méthodes de recherche du sous espace optimal P^* par UZABOC et ADAUZABOC. La première ligne se réfère à la recherche de P^* par UZABOC. Pour ξ fixé, les contraintes sont fixées et l'ensemble des solutions réalisables est défini. UZABOC recherche alors le P^* satisfaisant ces contraintes. Selon la performance de A, les contraintes sont modifiées par modification des bornes ξ. Cela se traduit par une réduction de l'ensemble des solutions réalisables, une nouvelle recherche du P^* conforme aux contraintes est alors lancée. Ces opérations sont renouvellées jusqu'à ce que les bornes cessent d'évoluer. La deuxième ligne montre l'évolution de la recherche de P^* par ADAUZABOC, où dans ce contexte, l'espace des solutions réalisables évolue pendant la recherche de P^*.

– Les données de *WDBC* concernent le diagnostic de cancer du sein. 569 images de seins sont numérisées et décrites par différents attributs géométriques (périmètre, aire, concavité, compacité, rayon *etc.*.) ainsi que des attributs de variations de niveaux de gris, dans le but de repérer des masses cancéreuses.

3.8.2 Protocole expérimental

Le protocole expérimental suivi fixe les différents paramètres pour l'étude comparative des approches BOC, ADAUZABOC et BC. Les différentes approches à évaluer repose sur une construction et une diagonalisation d'une matrice de corrélations entre les variables des données (ACP). Afin de respecter des principes de base de l'analyse de données, des pré-traitements ont été réalisés. Les jeux de données on tous été centrés et des expériences ont été conduites sans ou avec réduction afin d'attribuer une importance équitable à tous les descripteurs. Dans le même esprit et concernant la recherche du sous-espace optimal pratiqué par BOC, UZABOC, ADAUZABOC et BC, le nombre de dimensions du sous-espace peut :

– être fixé et constant pendant tout le processus d'amélioration de A,

– évoluer au fil des itérations selon l'heuristique consistant à ne sélectionner que les vecteurs propres correspondant aux valeurs propres positives.

Les approches ont également été éprouvées selon différents algorithmes de *clustering* boîte noire afin de valider l'amélioration des performances de ces algorithmes. Les différents algorithmes A testés [2] sont :
– K-MEANS (cf. section 1.3.1.1) ;
– SPECTRAL CLUSTERING (cf. section 1.3.1.2) sur le graphe des 15 plus proches voisins avec le laplacien L_{rw} ;
– CLINK (cf. section 1.2.2).

Pour ces différents algorithmes de *clustering*, le nombre de groupes à déterminer correspond au nombre de classes $n_k = n_c$. Ensuite, différentes stratégies ont été envisagées pour générer différentes informations de semi-supervision à partir des données. Comme il s'agit de données pour lesquelles on peut obtenir les classes des individus, cette information sert à générer des contraintes valides par rapport à l'objectif d'amélioration de performance. Celles-ci ont été générées aléatoirement.

Cependant, dans l'optique d'observer l'amélioration des contributions, à nombre de contraintes données augmentant, plusieurs modes de génération peuvent être considérés. Les expériences présentées ont été réalisées selon la stratégie suivante [3] : partant d'un ensemble de contraintes $\mathcal{ML}$ et $\mathcal{CL}$, celles-ci sont conservées et enrichies par de nouvelles, jusqu'à atteindre un nombre de contraintes fixé. De plus, les contraintes sont tirées de telle sorte à conserver un nombre équilibré de $\mathcal{ML}$ et de $\mathcal{CL}$.

Dans l'optique d'étudier la robustesse des contributions, une partie des expériences a été renouvelée en introduisant du bruit dans les contraintes, dans le sens où certaines contraintes $\mathcal{ML}$ ou $\mathcal{CL}$ sont incohérentes avec les classes d'origine. Le pourcentage de contraintes bruitées est fixé à 20%.

Ensuite, différents choix d'initialisation peuvent être réalisés sur A afin de (1) placer les approches comparatives dans une posture d'égalité vis à vis de l'instabilité inhérente à A lorsque celui-ci est par nature non déterministe (KM, SC), ou au contraire (2) d'étudier la robustesse des approches au regard de cette instabilité :
– une même initialisation peut être apportée à l'algorithme A pour toutes les exécutions des approches comparatives. Ceci permet d'observer la stabilité de ces approches pour l'amélioration d'une boîte noire A rendue déterministe ;
– une même initialisation (par exécution) peut être considérée et identique pour toutes les approches comparatives. Ceci permet de mettre les approches sur un pied d'égalité et dans ce contexte, d'observer leur robustesse face à différents comportements de A ;
– une initialisation différente peut être envisagée pour toutes les approches et à chaque fois que A est sollicité pour produire un *clustering*. Ce cas permet d'observer la robustesse des contributions et de BC face à une boîte noire A plus instable.

Dans les expériences présentées, les algorithmes de *clustering* employés ont été initialisés selon la seconde stratégie. Pour finir, concernant l'approche BOC uniquement, le paramètre η permettant de moduler entre la cohérence et la consistance de la solution est affecté à différentes valeurs dans l'intervalle $[0..1]$ pour observer le comportement de la méta-heuristique selon ce paramètre.

2. les approches FKM, ALINK, SLINK, DBSCAN, KKM, KFKM et EM ont également été implémentées mais ne sont pas incluses dans ces tests.
3. une stratégie de génération aléatoire a également été implémentée mais n'est pas inclue dans ces tests.

Concernant le nombre d'itération maximum, pour ξ fixé, UZABOC réalise au plus 50 itérations pour approximer le point selle. Le nombre d'itération global autorisant les modifications de ξ est fixé à 20, de même que pour le nombre d'étape de *boosting* pour BOC.

3.8.3 Évaluation interne

Les comportements des différentes contributions ont été observés en parallèle sur *Iris* pour une exécution des méta-algorithmes, selon deux angles et pour deux approches de clustering différentes : KM et CLINK. La première observation consiste à étudier le phénomène de convergence des approches. UZABOC et ADAUZABOC approximent le point selle du lagrangien avec (UZABOC) ou sans (ADAUZABOC) variation sur les contraintes lors de l'apprentissage d'un sous-espace optimal. Ainsi, le saut de dualité doit tendre vers 0, ce qui caractérise l'optimalité de la solution au regard de la satisfaction des contraintes du problème d'optimisation. À défaut, une *meilleure* approximation du point selle est obtenue, pour un saut de dualité positif. Une autre manière de voir cette convergence est d'observer la variation des poids entre deux étapes du méta algorithme, celui-ci devant tendre vers 0 à mesure que la convergence est approchée. Ce critère a été retenu pour observer la convergence de BOC, qui n'est pas exprimé explicitement comme la recherche d'un point selle. Ces deux critères sont couplés à l'observation de la satisfaction des contraintes utilisateurs $\mathcal{CL}$ et $\mathcal{ML}$. Cette observation permet de corréler la validité de la modélisation associée à la satisfaction des contraintes au regard de l'objectif initial. Enfin, comme indice de qualité du méta-algorithme employé, le critère externe d'information mutuelle normalisée (NMI 1.24) est indiqué à titre indicatif. Cela permet de mesurer l'impact sur la qualité du *clustering* de chaque étape du méta-algorithme. Ces différentes observations sont présentées dans les graphiques 3.9 à 3.14.

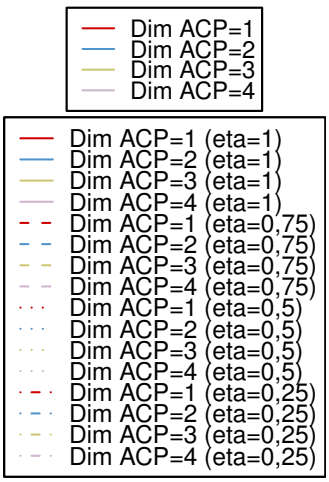

FIGURE 3.7 — Légende de l'évaluation interne pour UZABOC et ADAUZABOC (à gauche), et BOC (à droite).

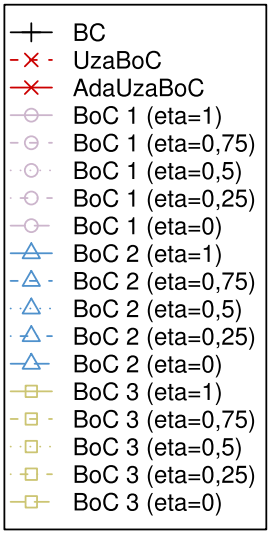

FIGURE 3.8 — Légende de l'évaluation externe de BOC, UZABOC et ADAUZABOC relativement à BC.

Étude empirique de la convergence

On remarque en premier lieu sur la figure 3.9 que, pour l'exécution concernée, les approches BOC convergent vers une stabilisation de la variation des valeurs de poids entre deux étapes.

Chaque étape de *boosting* permet d'obtenir des solutions très variées et on observe en général que plus l'on cherche à satisfaire la consistance ($\eta = 1$), plus les solutions obtenues satisfont les contraintes. De plus, si l'on observe la corrélation avec l'évolution de la mesure d'évaluation externe, on constate que les performances sur l'ensemble des jeux de données sont complètement corrélées avec la satisfaction des contraintes tirées au hasard, quelque soit leur nombre. La performance finale est déterminée uniquement par la décision induite par le type de fusion employé pour BoC.

Concernant UzaBoC (Fig. 3.10) et AdaUzaBoC (Fig. 3.11), on constate cette fois en premier lieu que les deux approches tendent à converger vers une annulation du saut de dualité. Ceci est plus flagrant sur l'approche AdaUzaBoC, étant donné qu'elle converge plus rapidement que UzaBoC (les contraintes s'adaptant pendant la résolution du problème par Uzawa). De plus, les évolutions des méta-algorithmes tendent à produire des solutions satisfaisant davantage les contraintes $\mathcal{ML}$ et $\mathcal{CL}$. Cette satisfaction progressive des contraintes est encore une fois corrélée quelque soit l'approche, à une amélioration de la performance relative au critère d'évaluation externe.

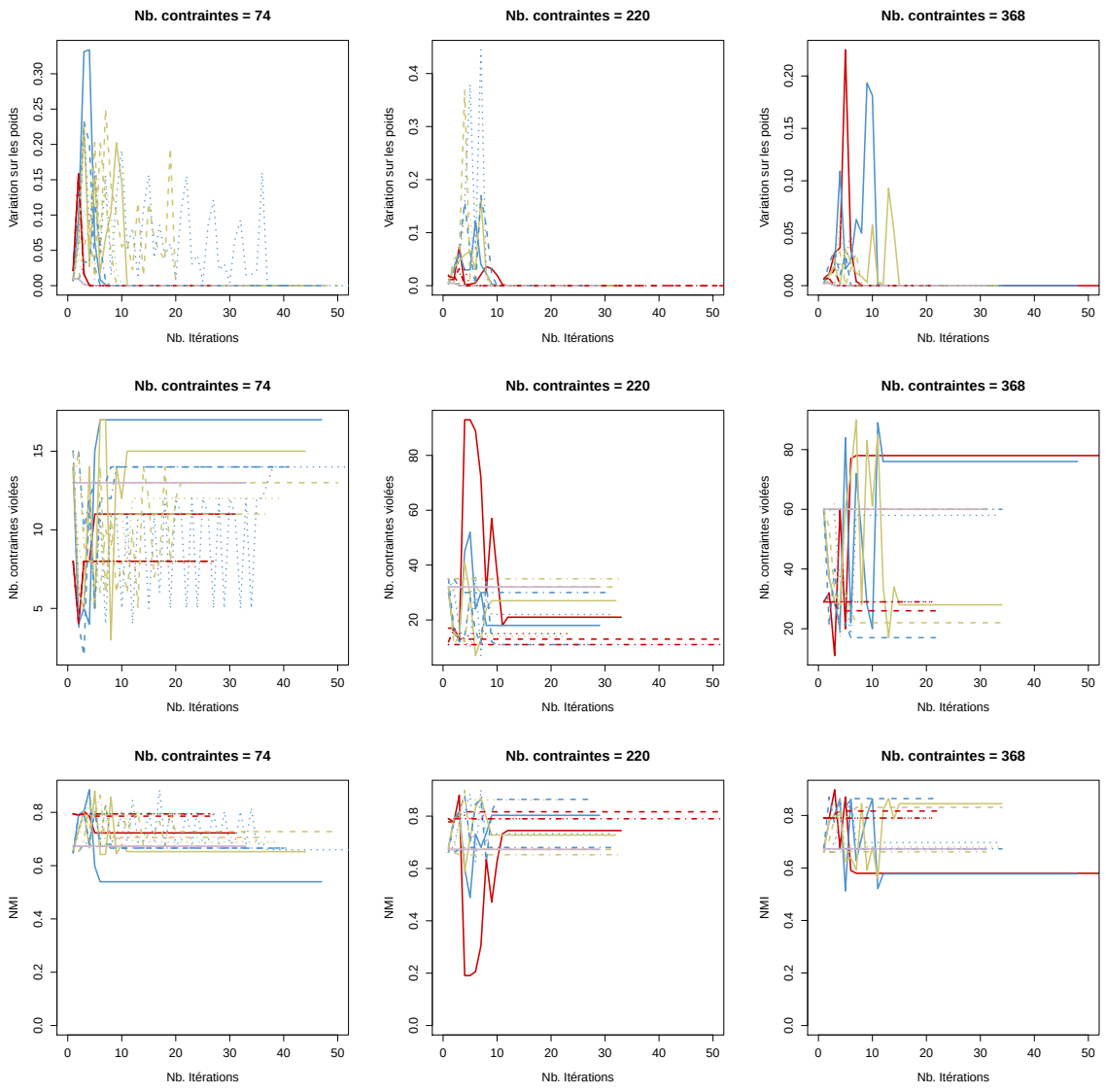

FIGURE 3.9 — Convergence empirique de BoC avec KM étudiée en observant la variation sur les poids sur *Iris* centré et réduit.

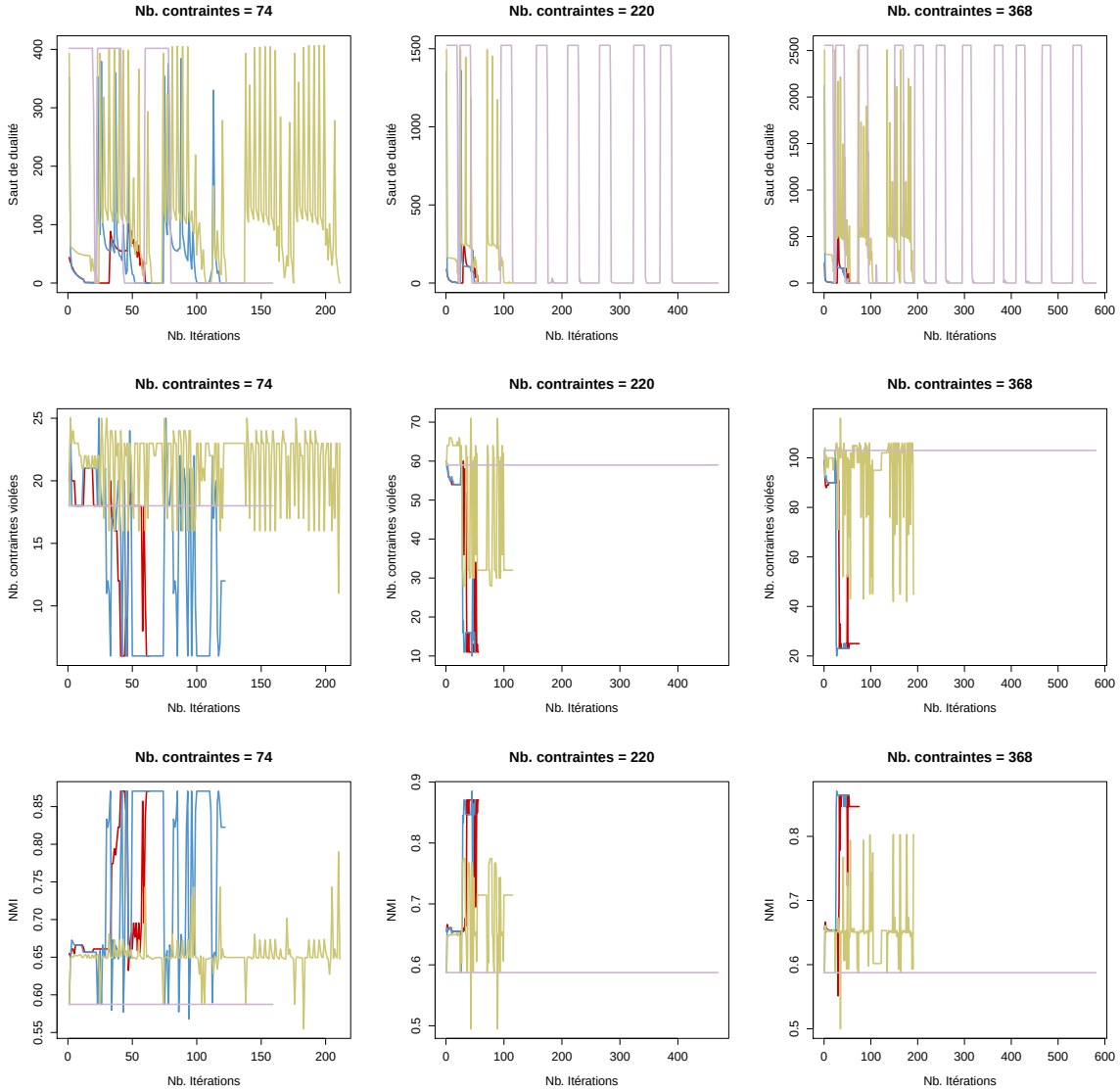

FIGURE 3.10 — Convergence empirique de UZABOC avec KM étudiée en observant le saut de dualité sur *Iris* centré et réduit.

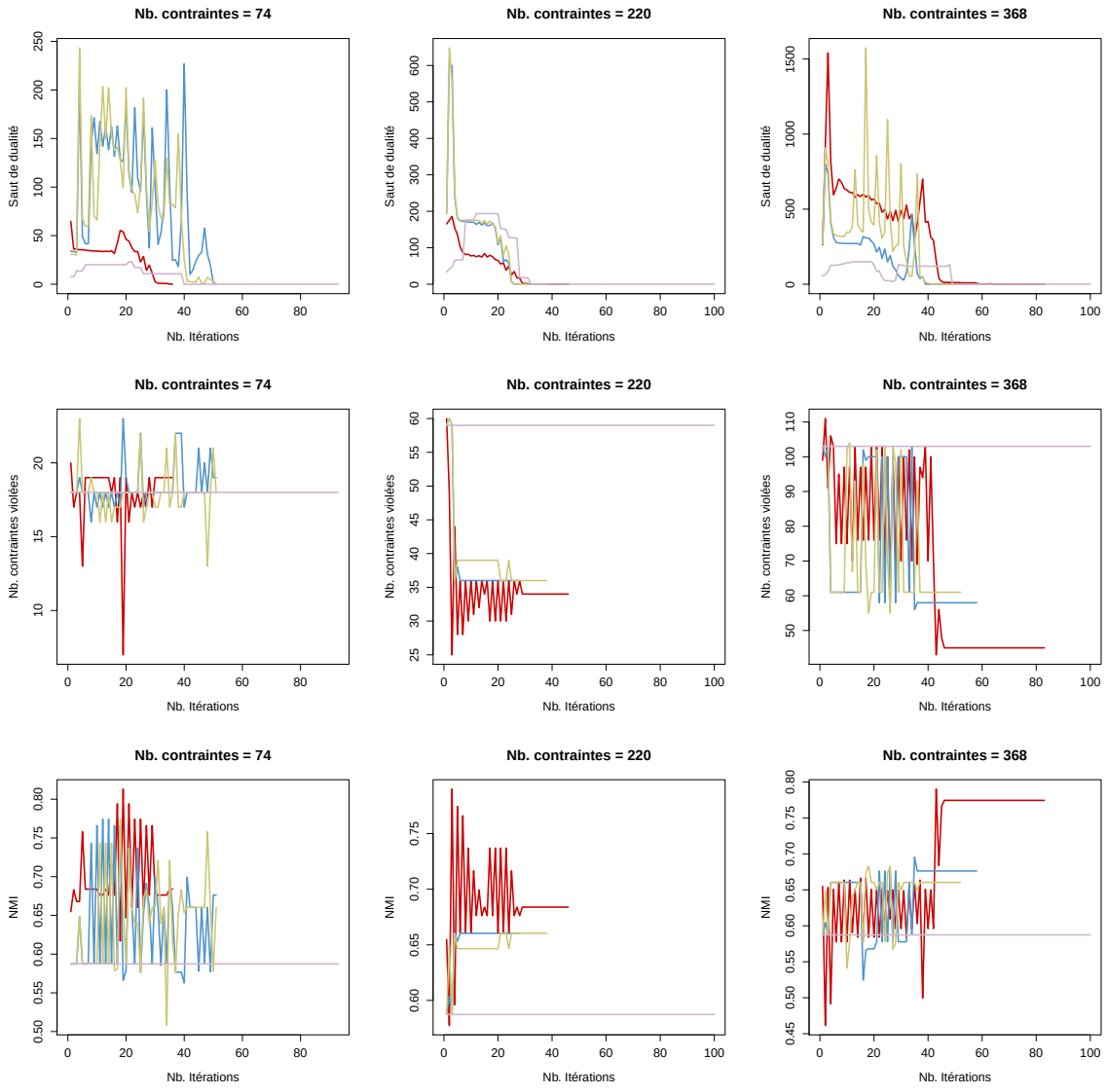

FIGURE 3.11 — Convergence empirique de ADAUZABOC avec KM étudiée en observant le saut de dualité sur *Iris* centré et réduit.

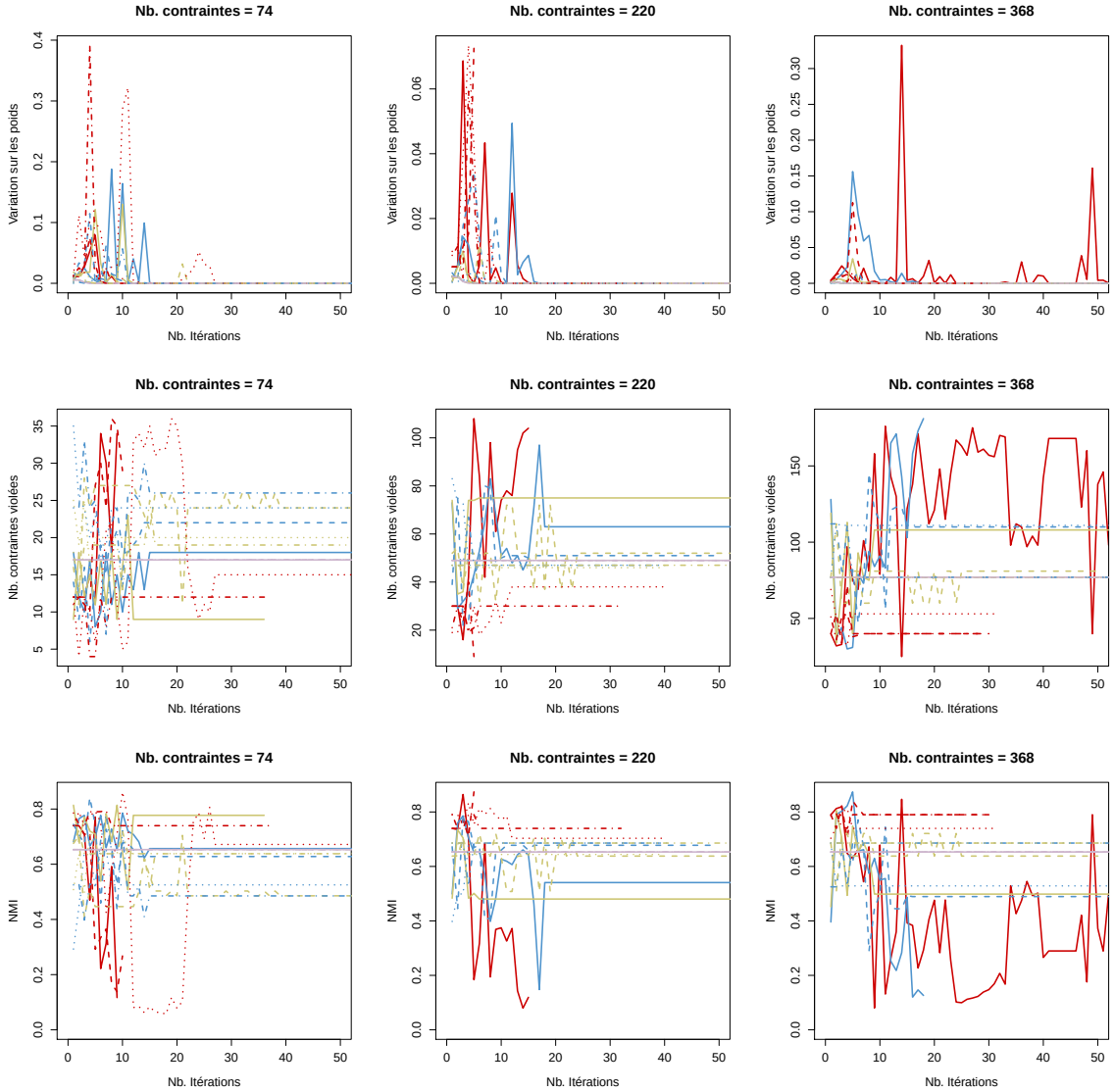

FIGURE 3.12 — Convergence empirique de BoC avec CLINK étudiée en observant le saut de dualité sur *Iris* centré et réduit.

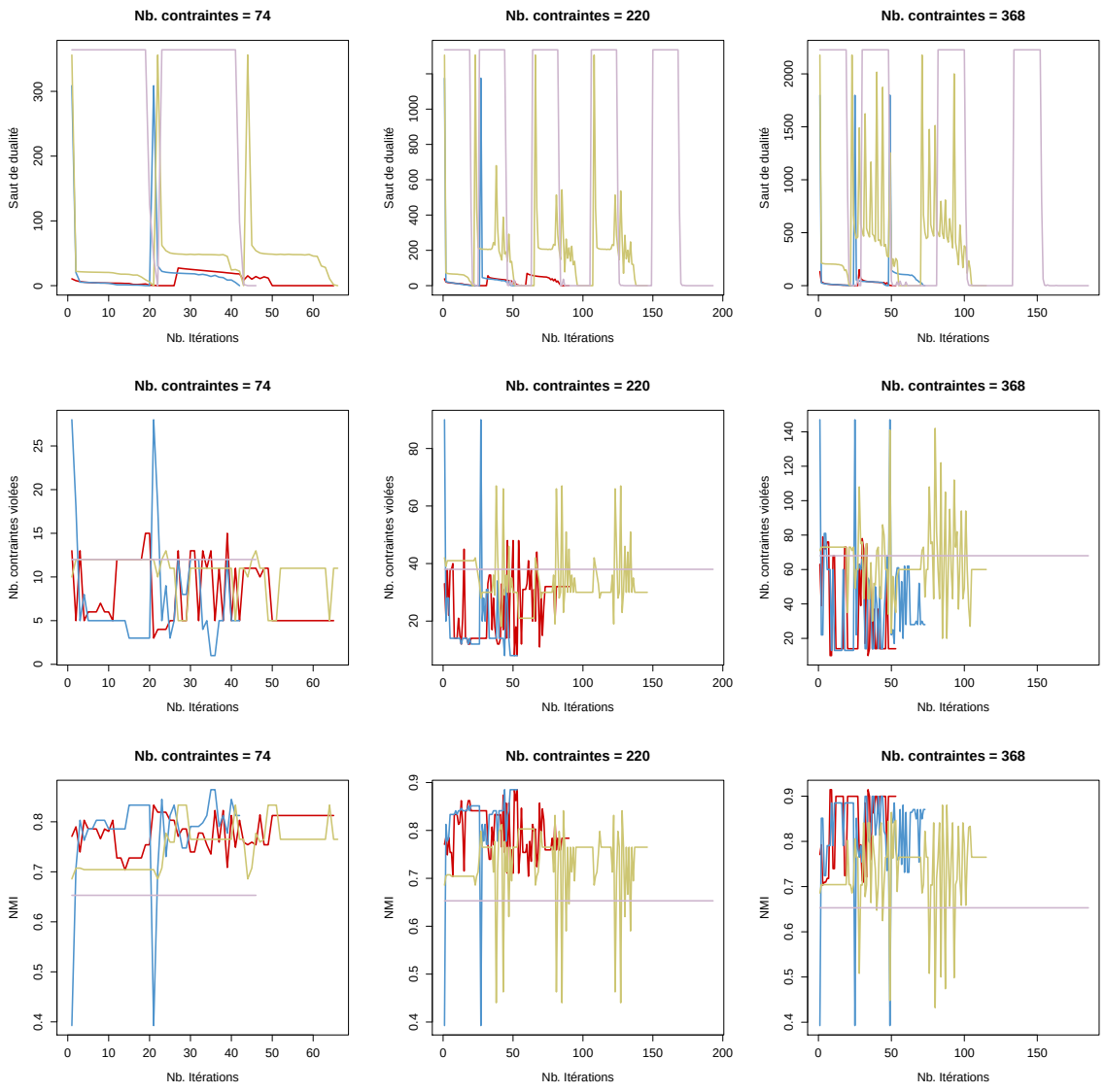

FIGURE 3.13 — Convergence empirique de UZABOC avec CLINK étudiée en observant le saut de dualité sur *Iris* centré et réduit.

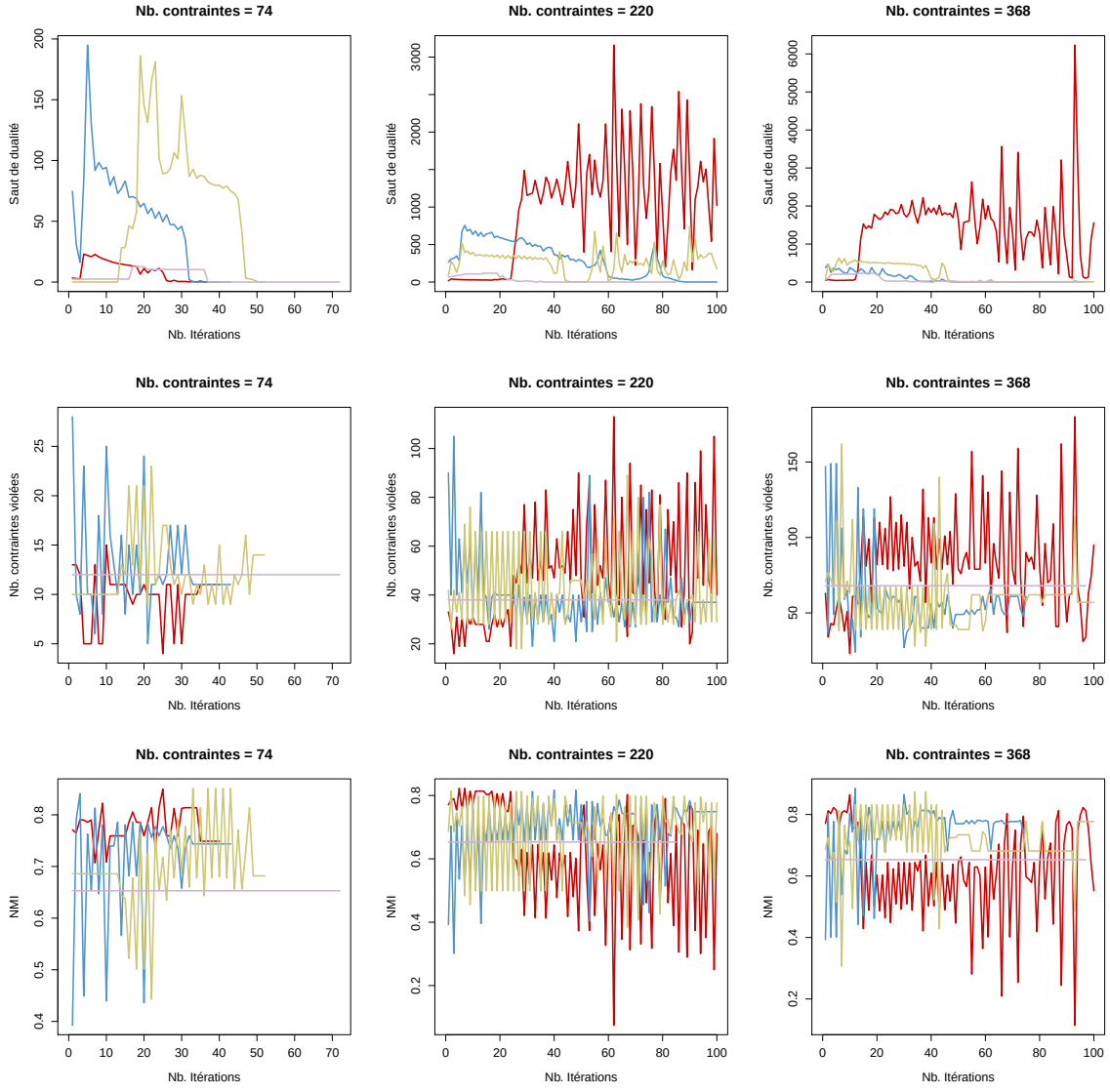

FIGURE 3.14 — Convergence empirique de ADAUZABOC avec CLINK étudiée en observant le saut de dualité sur *Iris* centré et réduit.

3.8.4 Évaluation externe

Les approches BoC, UzaBoC et AdaUzaBoC ont été évaluées empiriquement dans le but de mesurer leur performance relativement à l'évolution du nombre de contraintes, décrite dans le protocole précédent. BoC a été testé selon différentes valeurs de η (eta) et selon différent types de fusions finales pour obtenir un *clustering* à partir des différentes hypothèses produites durant le processus de *boosting*. Les différentes instances de l'approche BoC sont désignées par :

BoC 1 : des poids moyens $\tilde{w}_{ij}$ sont déterminés pour toutes les paires d'individus impliqués dans les contraintes :

$$\tilde{w}_{ij} = \sum_{t=1}^{t_f} \alpha^{(t)} w_{ij}^{(t)}$$

Ces poids servent pour obtenir une nouvelle représentation optimale des individus sur laquelle appliquer A. Cette forme de fusion est suggéré dans le paragraphe 4.6.2.§ 2.

BoC 2 : une matrice de similarité $\tilde{K}$ (noyau) est construite à partir d'une moyenne pondérée par les confiances des hypothèses de *clustering* sur les paires d'individus :

$$\tilde{K} = \sum_{t=1}^{t_f} \alpha^{(t)} H^{(t)}$$

Cette matrice noyau sert directement de matrice de similarité, ou à redéfinir une distance, utilisée ensuite par A pour obtenir un *clustering* des individus. Ce type de construction de C est semblable à celle employée par BC, et est suggéré dans le paragraphe 4.6.2.§ 1.

BoC 3 : une matrice de similarité $\tilde{K}$ est construite à partir d'une somme pondérée des similarités entre individus obtenues dans les différentes représentations optimales :

$$\tilde{K} = \sum_{t=1}^{t_f} \alpha^{(t)} X^{(t)} X^{(t)\top}$$

Ce type de construction de $\tilde{K}$ et son utilisation comme matrice de produit scalaire pour définir une distance, revient à calculer la matrice moyenne des distances entre individus à chaque étape de *boosting*. Cela revient également à calculer une distance à partir de la concaténation des différentes représentations optimales obtenues lors du processus itératif, comme suggéré au paragraphe 4.6.2.§ 3.

Les résultats présentés dans les graphiques 3.15 à 3.26 permettent d'étudier les différentes approches selon le jeu de donnée et les algorithmes de *clustering* employés. Chaque série de graphiques présente l'évolution de la performance des algorithmes de *clustering* KM, SC et CLink, relativement au nombre de contraintes, pour chaque jeu de données et dans des configurations différentes. Ces expériences nous permettent de discuter de :

- l'apport des méta-algorithmes sur la qualité des groupes produits par les différents algorithmes de *clustering* ;
- l'impact du paramètre η (eta) sur BoC, et d'établir par ce biais l'impact de la recherche de cohérence sur la performance ;
- la performance relative des contributions par rapport à BC;
- l'impact de la normalisation des données.

Chaque série de graphiques est constituée de deux lignes de trois graphiques. La première ligne concerne l'évolution du F-score et la seconde, l'évolution de l'information mutuelle normalisée. Dans chaque ligne, les trois graphiques concernent, dans l'ordre, KM, SC et CLink.

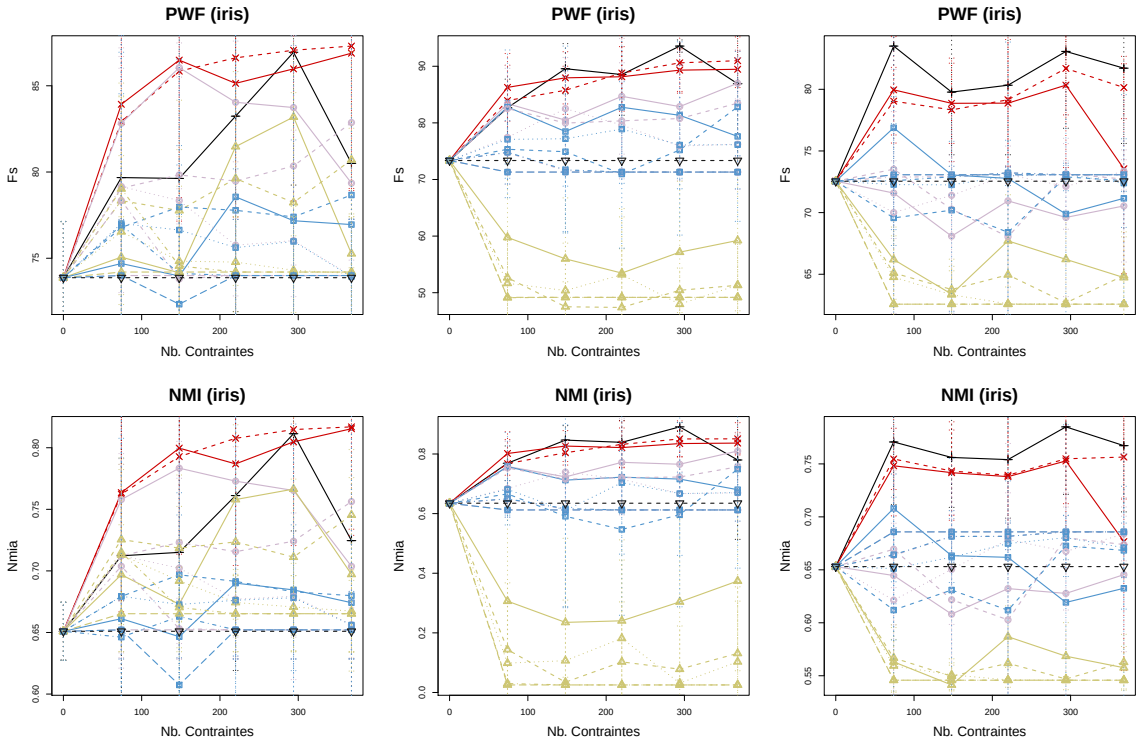

FIGURE 3.15 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur *Iris* centré et réduit.

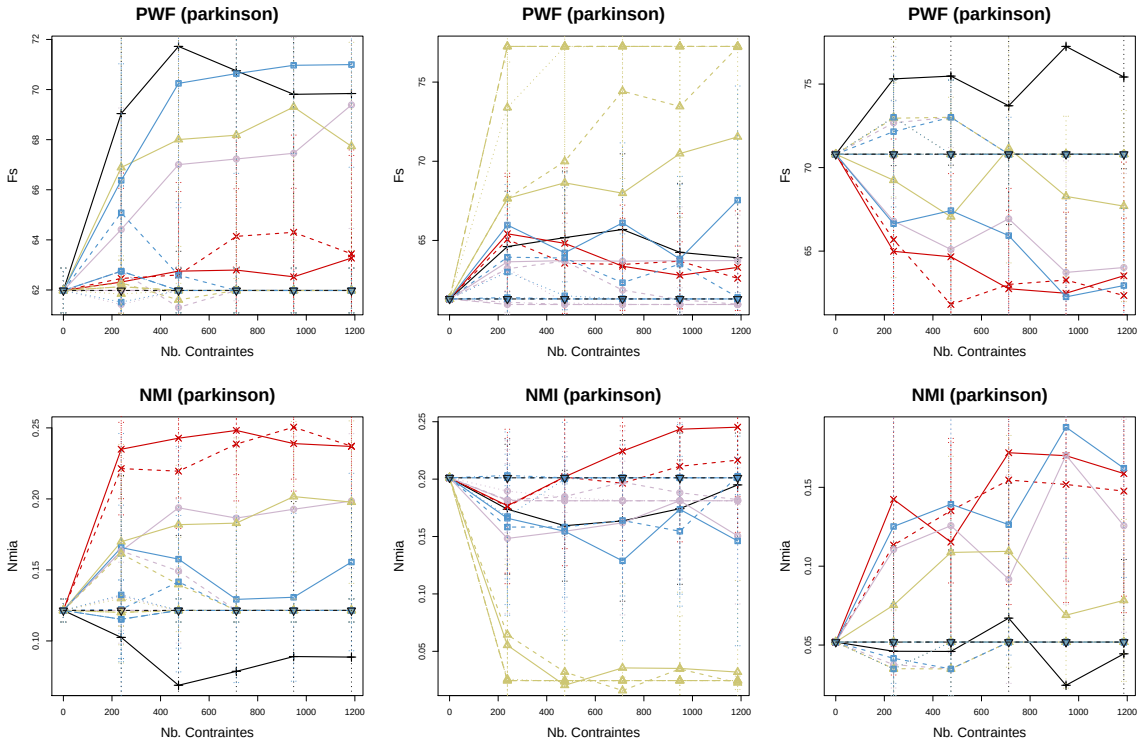

FIGURE 3.16 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur *Parkinson* centré et réduit.

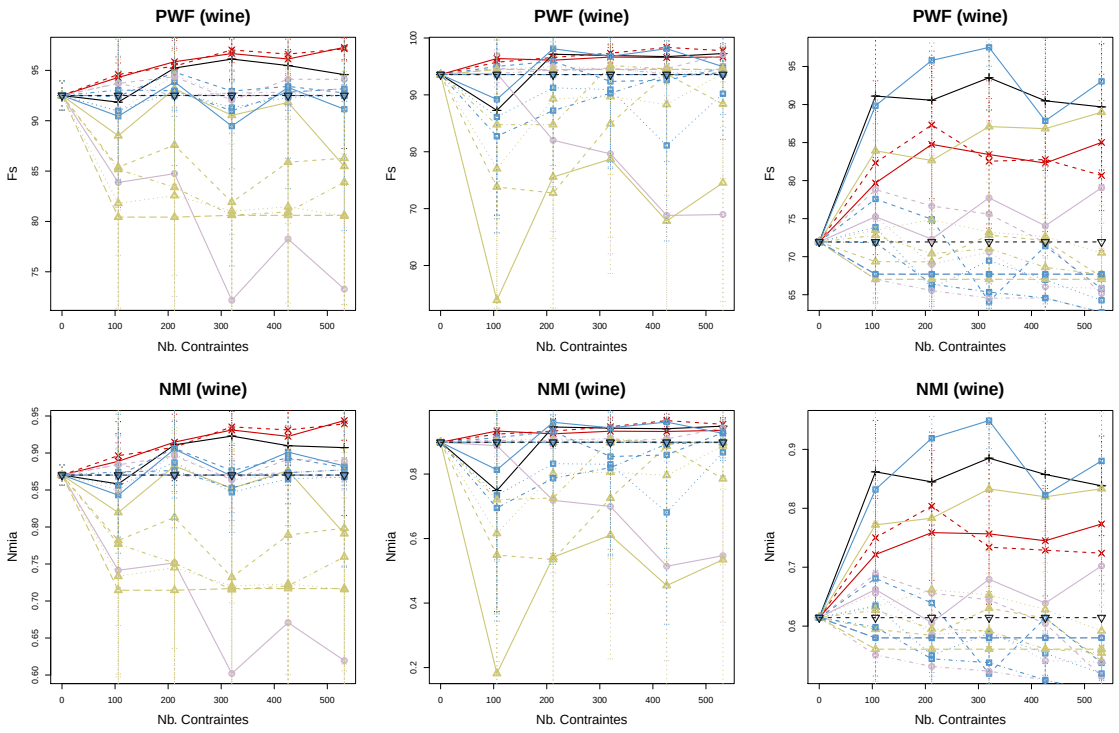

FIGURE 3.17 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur *wine* centré et réduit.

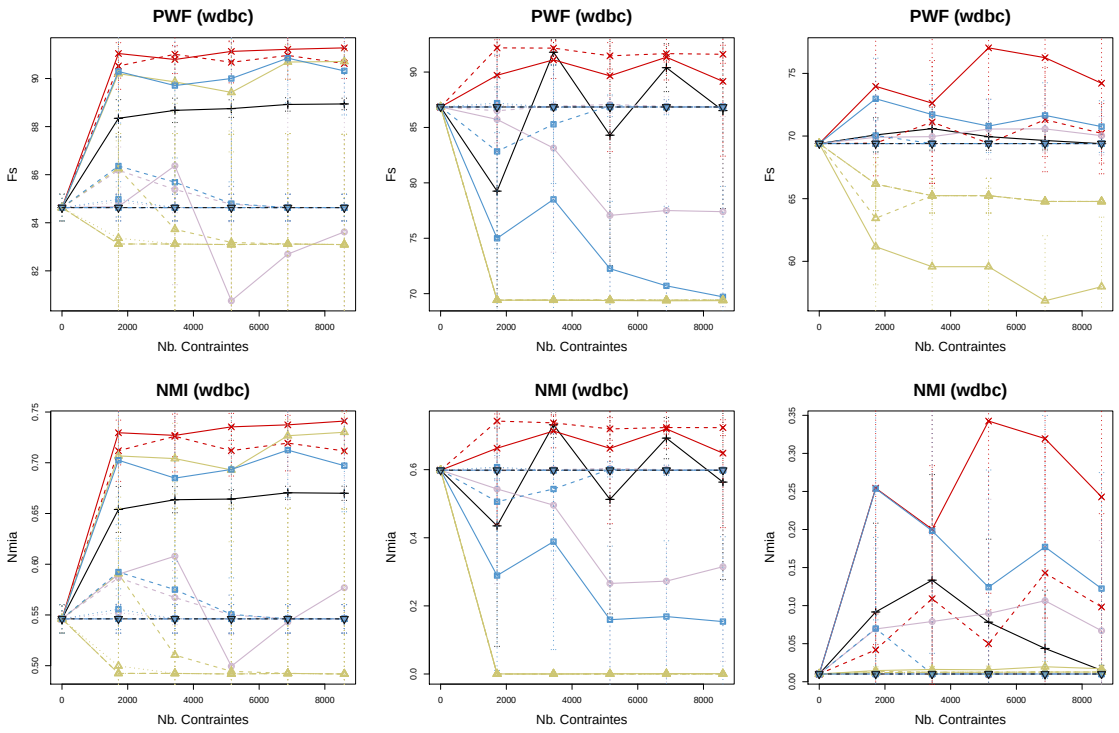

FIGURE 3.18 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur *WDBC* centré et réduit.

Amélioration de la performance des algorithmes de *clustering*

Globalement, comme on peut le constater sur la quasi-intégralité des données centrées et réduites (Fig. 3.15 à Fig. 3.18), les contributions UZABOC et ADAUZABOC permettent systématiquement d'améliorer la performance des trois algorithmes de *clustering* employés. Le cas où l'amélioration ne semble pas être réalisée (pour le jeu de donnée *parkinson* (Fig. 3.16)) est relatif à la mesure de F-score, l'amélioration est observable selon l'information mutuelle normalisée. Ceci s'explique par l'obtention d'une solution moins dégénérée, dans le sens où un groupe devient plus important en taille que les autres, ce qui favorise le rappel et *a fortiori* le F-score. L'approche BOC est quant à elle plus instable.

Amélioration de la qualité relativement à l'état de l'art

On constate également que sur la grande majorité des jeux de données, les approches UZABOC et ADAUZABOC surpassent l'approche BC. Sur *Iris*, L'écart de performance est plus mince, et sur *Parkinson* ces écarts sont relatifs à la mesure d'évaluation, notamment à la faiblesse du F-score. Seul CLINK semble être davantage amélioré par BC que par UZABOC ou ADAUZABOC. Concernant les variantes de BOC et les différentes valeurs du paramètre η, les résultats sont mitigés. On remarque que BOC 3 a un plus mauvais comportement dans le cas général que les versions BOC 1 et BOC 2. En revanche dans tous les cas, on constate que plus la prise en compte de la cohérence est importante, plus la performance se dégrade, ce qui semble contredire l'intuition de départ concernant la volonté de préserver au mieux la distribution d'origine des données. Néanmoins, il est normal d'observer de tels résultats relativement aux mesures d'évaluation externe, car plus la part de consistance est importante, plus on a de chances de réussir à satisfaire les contraintes, et ainsi à retrouver une bonne part de la classification de référence. Une évaluation alternative serait de ne mesurer par évaluation externe, que le résultat de *clustering* sur les individus non impliqués dans une contrainte. De plus, les approches UZABOC et ADAUZABOC, qui dominent les différentes approches envisagées, reposent sur la maximisation du critère de cohérence régularisé.

Impact du bruit dans les informations externes

L'impact du bruit a également été observé sur les différents jeux de données (Fig. 3.19 à Fig. 3.22). La constatation principale que l'on peut faire dans ce contexte est que hormis pour le jeu de données *WDBC*, les contributions sont en général moins robustes que BC. De plus l'observation des différentes variantes de BOC indique cette fois que la recherche uniquement de consistance fait chuter l'amélioration de la performance, ce qui donne du crédit à la recherche de cohérence. Cependant, il est très difficile d'améliorer ne serait-ce que l'algorithme de base employé sur la représentation d'origine, dans la mesure où les approches de type BOC s'arrêtent souvent brutalement par non réalisation de l'hypothèse du classifieur faible. En effet si le jeu de donnée se prête aux approche de *clustering* semi-supervisées indépendante de l'algorithme, alors si celui-ci parvient à retrouver naturellement une bonne classification, il réalisera des erreurs sur les contraintes bruitées.

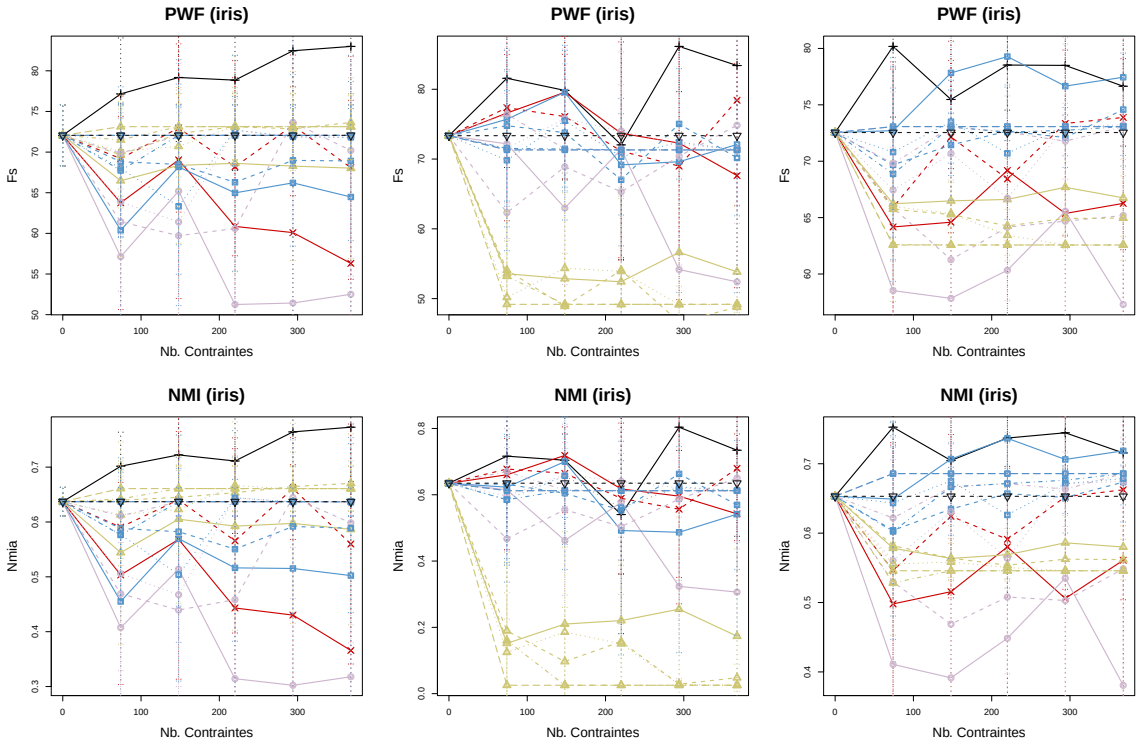

FIGURE 3.19 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur le jeu *Iris* centré et réduit avec contraintes bruitées.

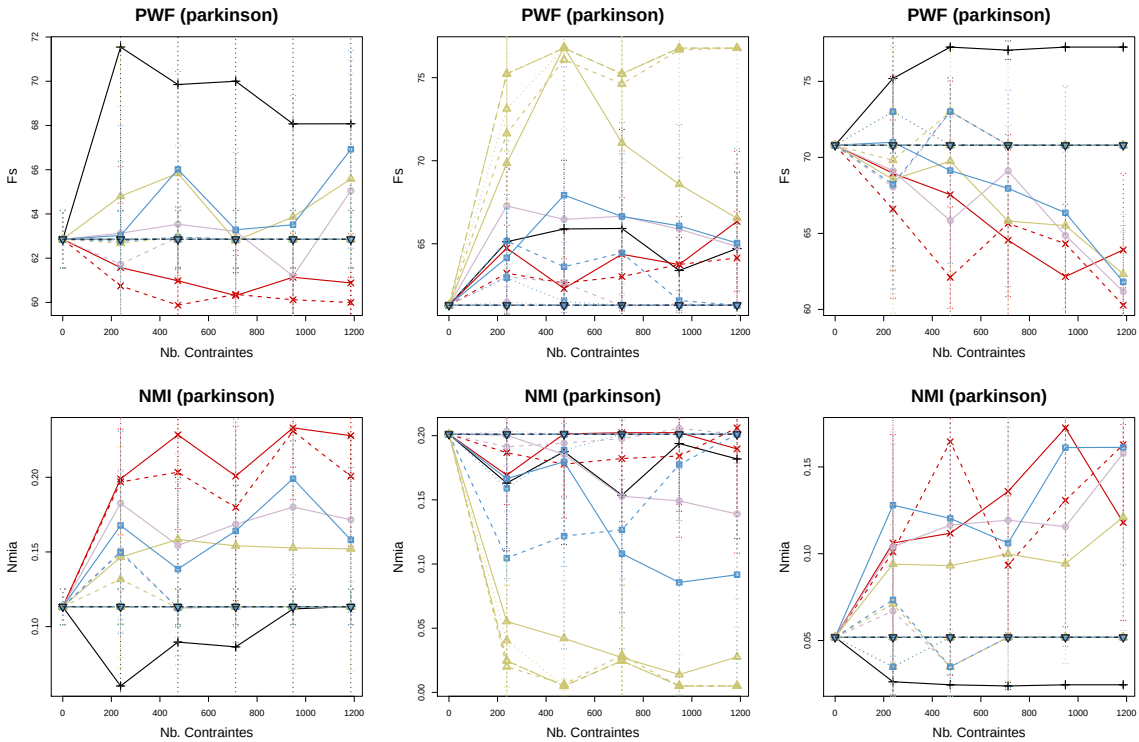

FIGURE 3.20 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur le jeu *Parkinson* centré et réduit avec contraintes bruitées.

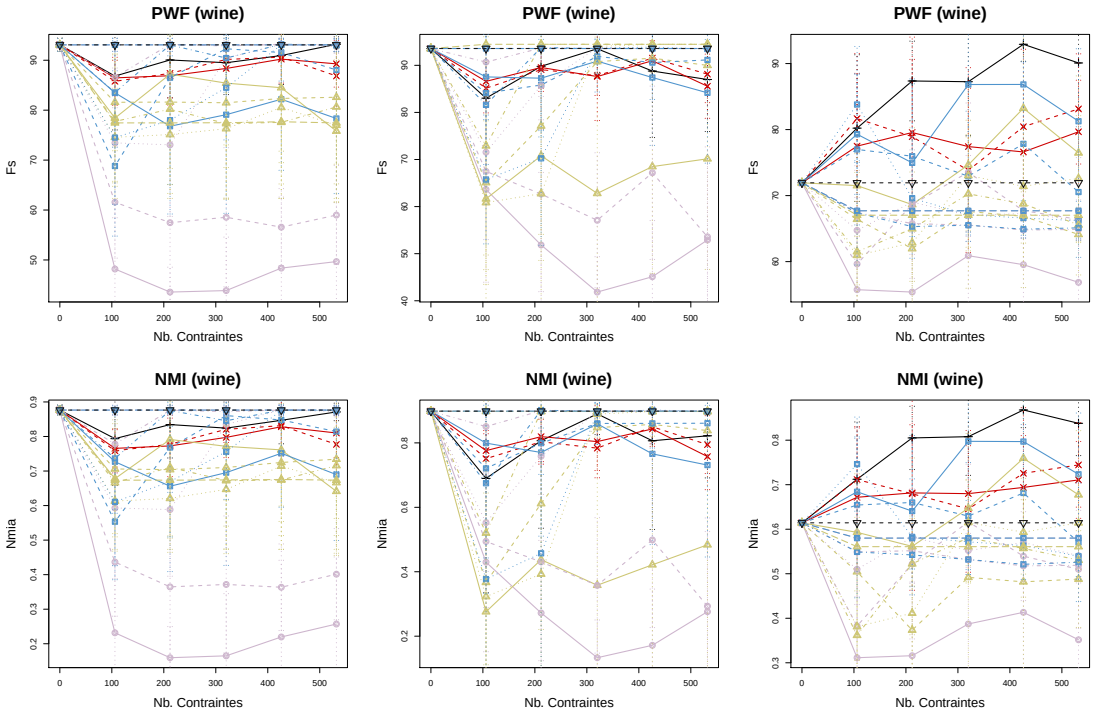

FIGURE 3.21 — Comparaison des approches BOC, UZABOC, ADAUZABOC et BC sur le jeu *wine* centré et réduit avec contraintes bruitées.

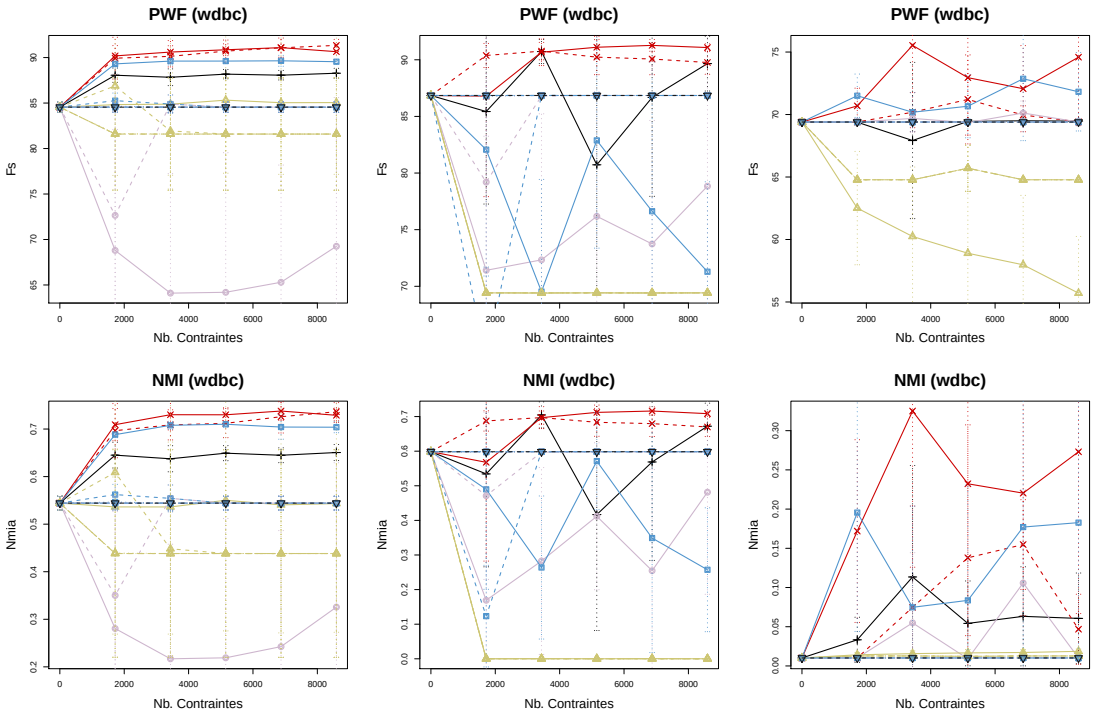

FIGURE 3.22 — Comparaison des approches BOC, UZABOC, ADAUZABOC et BC sur le jeu *WDBC* centré et réduit avec contraintes bruitées.

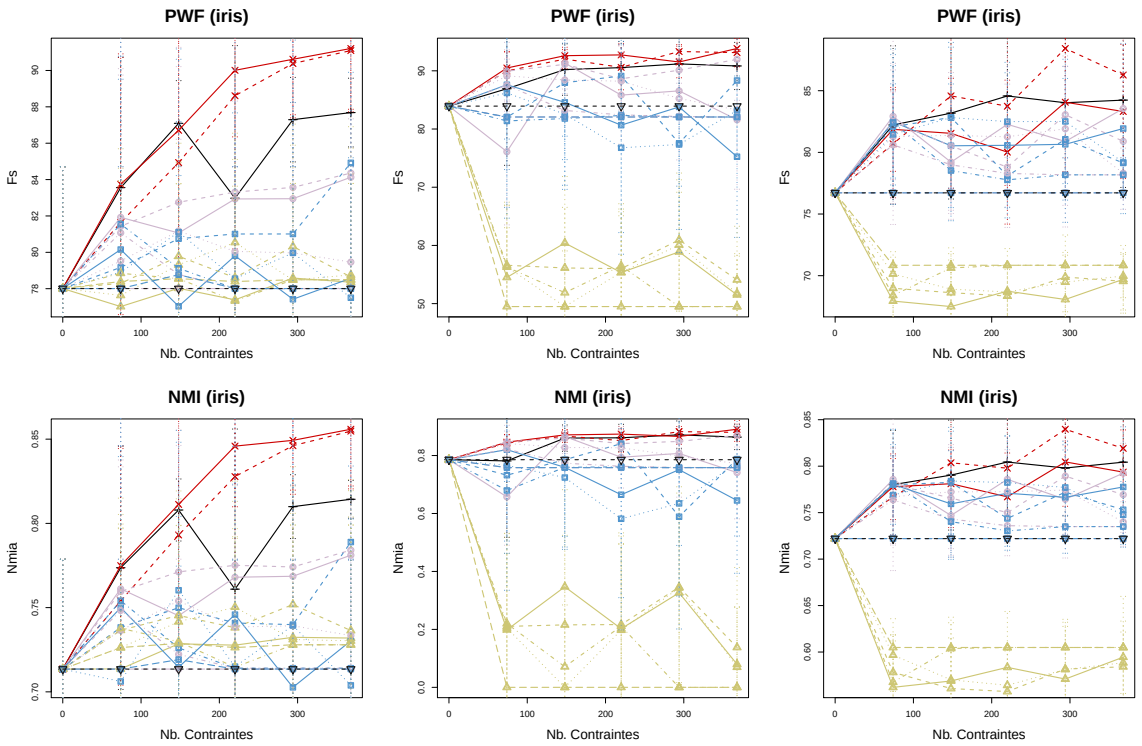

FIGURE 3.23 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur le jeu *Iris* centré.

Impact du pré-traitement des données sur l'efficacité des approches

Les différents comportement ont également été observés selon différents pré-traitements. Les résultats de la figure 3.15 à la figure 3.18 représentent le cas où les données sont centrées et réduites, alors que les résultats de la figure 3.23 à la figure 3.26 correspondent aux données centrées uniquement. L'opération de centrage des variables ou attributs est nécessaire de par la modélisation considérée du problème et la formalisation du critère de l'ACP. L'opération de réduction des variables à une variance unitaire avant tout traitement de type ACP permet de rétablir une équité entre les différentes variables. Cependant, si les variables de variance élevée sont très discriminatives, au sens ou la dispersion des individus selon ces variables permettent de retrouver naturellement les classes d'individus, alors il peut être bon de conserver davantage l'information portée par elles dans la définition de la nouvelle représentation optimale sur laquelle effectuer le *clustering*.

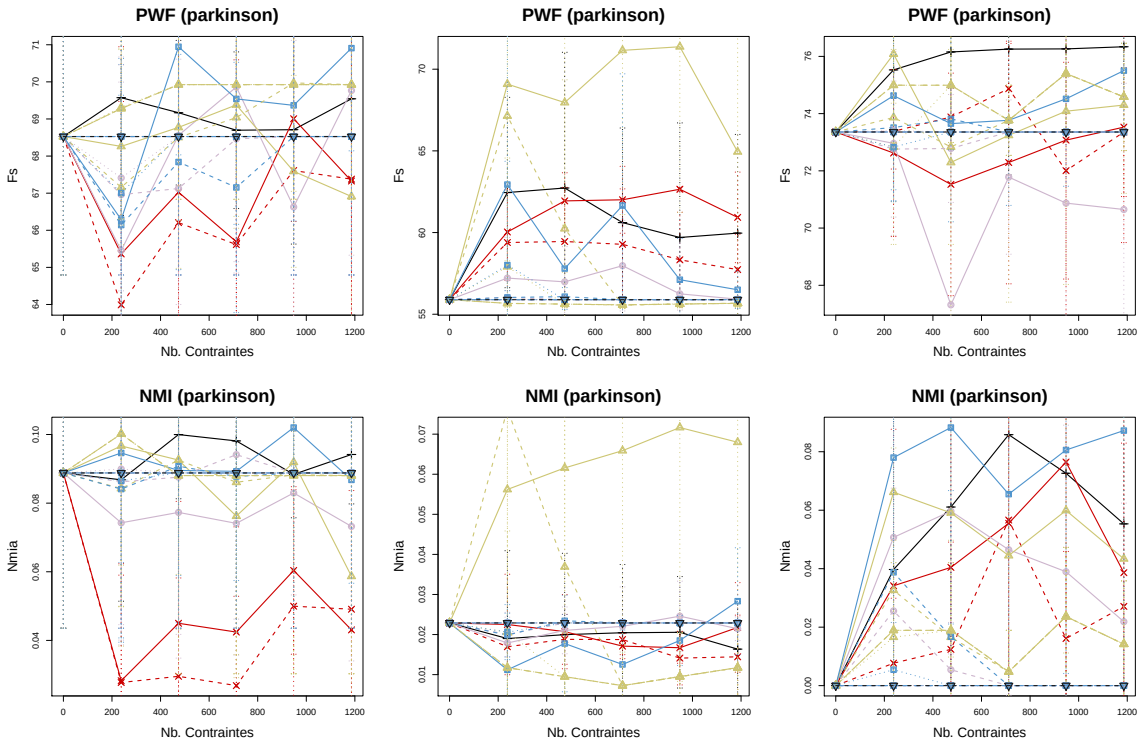

FIGURE 3.24 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur le jeu *Parkinson* centré. Celles-ci sont évaluées selon le F-score (en haut) et l'information mutuelle normalisé (en bas) pour KM, SC et CLink (dans l'ordre, de gauche à droite).

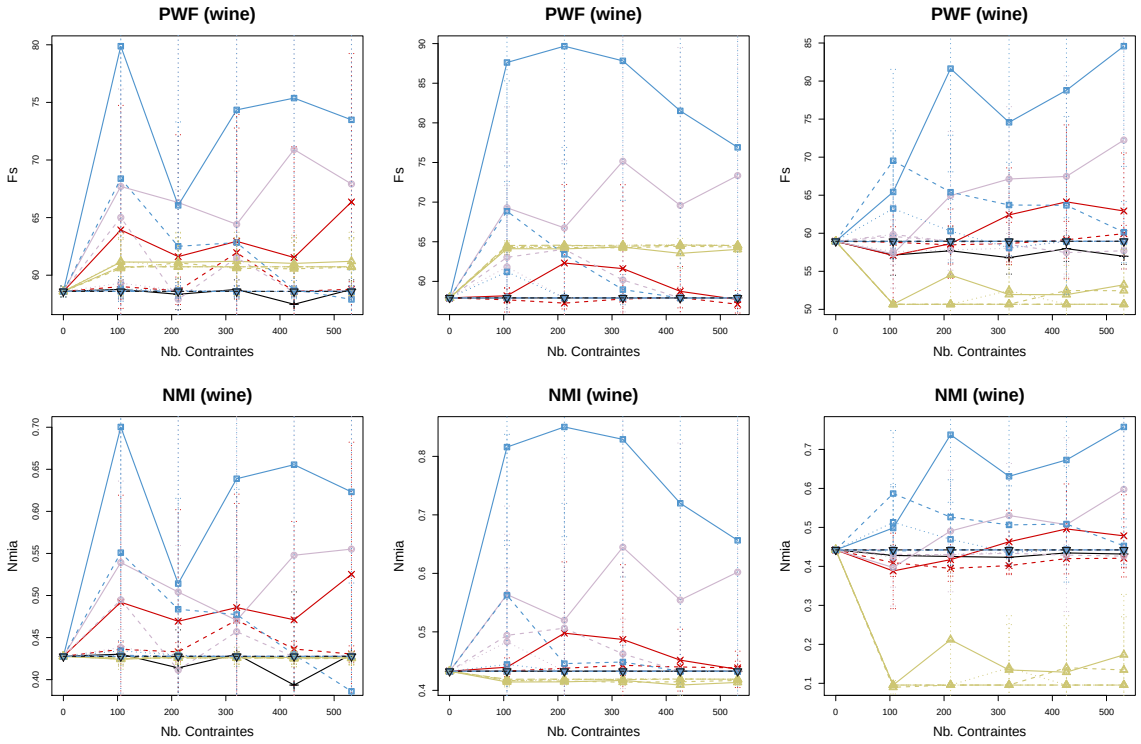

FIGURE 3.25 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur le jeu *wine* centré.

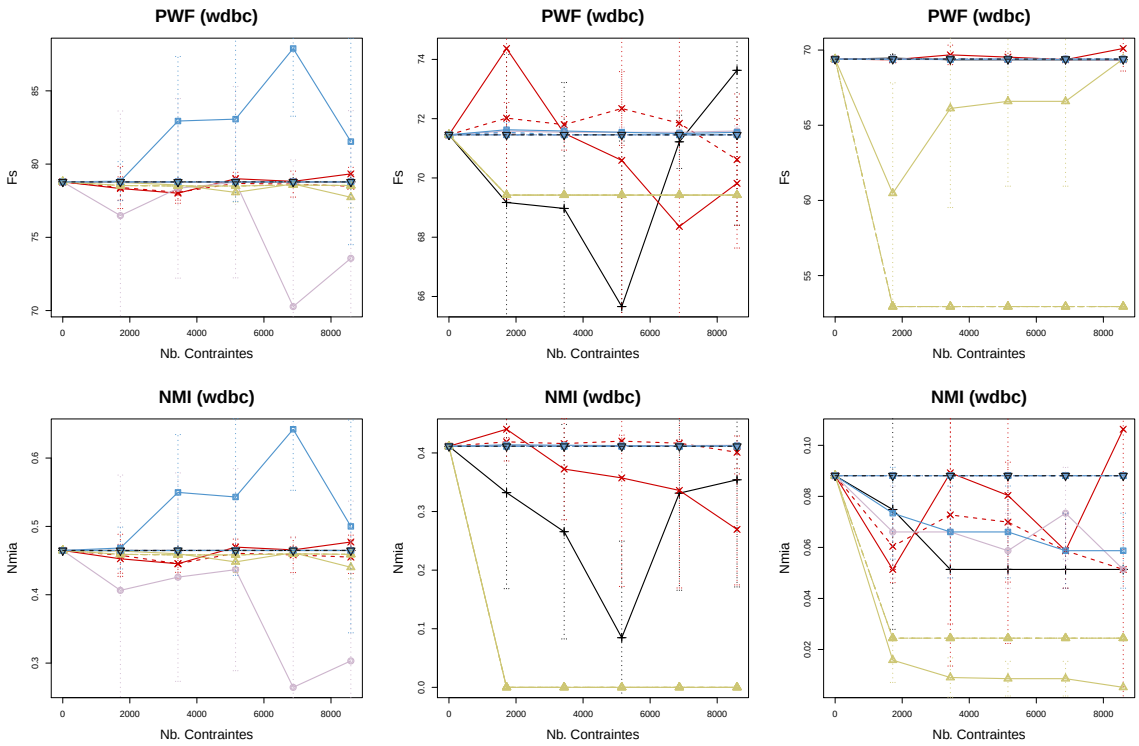

FIGURE 3.26 — Comparaison des approches BoC, UzaBoC, AdaUzaBoC et BC sur le jeu *WDBC* centré.

3.9 Discussion

Les contributions BoC , UzaBoC et AdaUzaBoC reprennent les travaux de [Liu et al., 2007] sur le développement de BC, et proposent des extensions afin de respecter les différentes propriétés introduites : la cohérence et la consistance. L'analogie entre BoC et BC permet d'argumenter sur les différentes possibilités pour réaliser un *boosting* d'un algorithme quelconque de *clustering* en vue d'en améliorer la performance. Une similitude forte a ensuite été dégagée entre BoC et UzaBoC, et sa variante adaptative AdaUzaBoC. Cependant, ces dernières permettent de s'abstraire d'un processus de fusion finale indispensable aux approches orientées *boosting*. Ceci est dû notamment au fait que la normalisation de la distribution de poids dans ces approches est telle qu'accentuer la satisfaction d'une partie des contraintes utilisateurs implique un relâchement des autres contraintes. Le méta-algorithme BoC souffre alors dans ce contexte d'un problème d'oscillation dans la satisfaction des contraintes et s'en remet à la décision finale modulée par les différents paramètres de confiance.

Les différentes approchent ont le défaut d'être limitées par le fait qu'une projection linéaire est réalisée pour déterminer à chaque étape la représentation optimale. Dans le cas général, il peut exister des contraintes $\mathcal{CL}$ impliquant des individus se situant entre d'autres individus impliqués eux dans une contrainte $\mathcal{ML}$, et tels que tous ces individus soient alignés. Un tel scénario rend la satisfaction des contraintes impossible car aucun sous espace ne peut rapprocher les individus $\mathcal{ML}$ sans rapprocher les individus $\mathcal{CL}$. Ainsi, la grande majorité des algorithmes de *clustering*, si ils parviennent à regrouper ces individus $\mathcal{ML}$ regrouperont alors les individus $\mathcal{CL}$. Une perspective envisageable serait de réaliser une projection non linéaire de l'ensemble des individus. Néanmoins cette solution est en général plus coûteuse au sens de la complexité algorithmique.

3.10 Conclusion

Ce chapitre a permis de présenter la problématique du *clustering* semi-supervisé. Un historique des différentes approches clés a été développé avant de présenter le socle des contributions proposées. Celles-ci se fondent sur l'approche BC proposée par [Liu et al., 2007] et proposent de l'étendre en introduisant des propriétés devant être satisfaites par les approches de type méta-algorithme indépendantes de tout algorithme de *clustering*. L'approche BoC fondée sur le *boosting* se rapproche de BC et permet de trouver un ensemble de solutions de *clustering* satisfaisant chacune au mieux une partie des contraintes. Différentes procédure de décision du *clustering* final ont été proposées afin de combiner ces différents résultats. L'approche UzaBoC est plus élégante puisqu'elle permet, au travers d'une procédure d'optimisation numérique convergente, de déterminer à chaque étape une nouvelle représentation meilleure que la précédente. Les choix de modélisation proposés ont été éprouvés empiriquement, et des résultats prometteurs ont été obtenus notamment avec la variante AdaUzaBoC. Ces diverses contributions ne sont pas sans défauts et des améliorations pourront leur être apportées. Cependant, afin de résoudre les différents problèmes liés à la multiplicité des données autour du *clustering*, l'approche AdaUzaBoC a été retenue pour être utilisée dans le cadre du *clustering* collaboratif, proche du *clustering ensemble* ou consensus de partition, ainsi qu'au problème de recherche de *clustering* alternatifs. Ces différentes problématiques sont traitées simultanément dans la prochaine partie.

Classification non supervisée collaborative

4

Sommaire

4.1 Introduction **146**

4.2 Contexte **147**

4.3 Approches de type ensemble de *clusterings* **149**

4.3.1 *Clustering* consensus par ensemble de *clusterings* 149

4.3.2 Consensus de partitions 151

4.4 Approches collaboratives **154**

4.4.1 SAMARAH : système d'apprentissage multi-agents de raffinement automatique de hiérarchies 154

4.4.2 MOCLE : *clustering* d'ensemble multi-objectif 156

4.5 Approches alternatives **158**

4.5.1 COALA : *clustering* hiérarchique alternatif 158

4.5.2 ADFT : apprentissage de distance alternative 160

4.5.3 CAMI : estimation d'un mélange de modèles alternatifs 161

4.6 Contributions **163**

4.6.1 Motivation 163

4.6.2 CoBoC : *boosting* collectif et collaboratif pour la recherche de consensus 166

4.6.3 AlterBoC : *boosting* collectif et collaboratif pour la recherche d'alternatives 170

4.7 Évaluation **172**

4.7.1 Protocole expérimental 173

4.7.2 Évaluation interne 174

4.7.3 Évaluation externe 183

4.8 Discussion **205**

4.9 Conclusion **207**

4.1 Introduction

Ce chapitre introduit de nouvelles techniques pour obtenir un ou plusieurs regroupements d'individus décrits par plusieurs représentations, les approches CoBoC et AlterBoC. Ces algorithmes ont pour objectif de répondre à deux problématiques duales :

- CoBoC pour le *clustering* d'ensemble et le *clustering* collaboratif, ou la recherche d'une partition, ou de plusieurs partitions consensus à partir d'un ensemble (appelé aussi profil) de partitions ;
- AlterBoC pour l'*alternative clustering* ou la recherche de plusieurs partitions optimales, de bonne qualité et dissimilaires entre elles.

Dans un premier temps les approches typiques pour la résolution de ces problématiques sont présentées ainsi que les principes de base régissant les différentes contributions proposées. Dans un second temps, ces dernières seront détaillées. Elles sont fondées sur une forme de co-apprentissage (*co-training*) pour l'apprentissage simultané de solutions de *clusterings* répondant à ces problématiques. Le co-apprentissage est maîtrisé et mené *via* un partage d'informations entre les algorithmes de *clusterings* appliqués localement. Ce partage est réalisé au travers d'heuristiques de génération de contraintes puis d'intégration de celles-ci dans chacun des algorithmes de *clustering* réalisant leur tâche locale, dont le cœur correspond à l'approche AdaUzaBoC développée au chapitre 3.

L'objectif des approches de *clustering ensemble* étendues au cadre multi-vues est de produire une unique partition à partir d'un ensemble d'individus munis d'un ensemble de représentations. Cette partition correspond à une recherche de consensus entre plusieurs partitions locales, obtenues naturellement dans chaque vue par un algorithme de *clustering* adapté. La notation suivante permet de comprendre les formalismes des différentes approches proposées :

NOTATION	
n :	le nombre d'individus à regrouper.
$n_p^{(r)}$:	le nombre d'attributs décrivant les individus dans la vue r.
n_k :	le nombre de groupes à identifier.
n_c :	le nombre de classes associé aux données.
$\mathcal{X} = \{x_1, ..., x_n\}$:	l'ensemble des n individus à partitionner.
$X^{(r)} \in \mathbb{R}^{n \times n_p^{(r)}}$:	la représentation matricielle de $\mathcal{X}$ dans la vue r.
$x_i^{(r)} \in \mathbb{R}^{n_p^{(r)}}$:	la représentation vectorielle de l'individu x_i dans la vue r.
$C = \{C_1, ..., C_{n_k}\}$:	la structure de *clustering* en n_k groupes à construire.
$\Pi = \{C^{(1)}, ..., C^{(n_r)}\}$:	l'ensemble des n_r *clusterings* locaux dans chaque vue.
$C^{(r)} = \{C_1^{(r)}, ..., C_{n_k}^{(r)}\}$:	l'ensemble des n_k groupes du *clustering* dans la vue r.
$\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{n_c}\}$:	l'ensemble des n_c classes d'individus à retrouver.
$\mathcal{D} = \{\mathcal{D}_0, ..., \mathcal{D}_n\}$:	la structure de dendrogramme associée aux données.
$d_{(r)}(x_i, x_j)$:	la distance au sens général entre deux individus x_i et x_j dans r.
$\|\|x_i^{(r)} - x_j^{(r)}\|\|_p$:	la distance de Minkowski entre deux individus x_i et x_j dans r.
$\mathcal{ML}^{(r)}$:	l'ensemble des $(x_i, x_j) \in \mathcal{X}^2$ devant être regroupés dans r.
$\mathcal{CL}^{(r)}$:	les $(x_i, x_j) \in \mathcal{X}^2$ devant être séparés dans r.
$A^{(r)}$:	l'algorithme de *clustering* employé pour obtenir $C^{(r)}$.
$Link^{(r)}(x_i, x_j)$:	x_i et x_j sont regroupés par $A^{(r)}$ ou dans $C^{(r)}$.
$\overline{Link}^{(r)}(x_i, x_j)$:	x_i et x_j sont séparés par $A^{(r)}$ ou dans $C^{(r)}$.
$H^{(r)} \in \{0, 1\}^{n \times n}$:	la matrice de *clustering* associée à $C^{(r)}$

4.2 Contexte

le *clustering* d'ensemble

La problématique du *clustering* d'ensemble peut être définie ainsi : À partir d'un ensemble de partitions d'un même ensemble d'individus $\mathcal{X}$, trouver une partition consensus de l'ensemble d'individus. Le partition consensus est telle qu'elle doit être proche de chaque élément du profil (ou de l'ensemble) de partitions donné. Les algorithmes de la famille *clustering* d'ensemble ou *consensus clustering* visent simultanément plusieurs objectifs :

la **réutilisation des connaissances** et des outils de *clustering* existants lorsque d'une part on a à disposition plusieurs *clusterings* concernant l'ensemble d'individus $\mathcal{X}$ (émanant potentiellement de plusieurs vues différentes) que l'on souhaite utiliser sans réanalyser les données, et les combiner pour obtenir une solution plus robuste. D'autre part, si les *clusterings* ne sont pas connus, il est possible d'utiliser les algorithmes existants sur plusieurs vues des données, contenant un ensemble plus petits de descripteurs, et pour lesquelles les algorithmes classiques employés ont prouvé leur efficacité (KM, SOM, DBSCAN, *etc.*) ;

la **décentralisation des calculs** concerne le cas où les données sont effectivement décentralisées, *i.e.* réparties sur plusieurs sites. Dans ce contexte il peut être préférable d'effectuer les *clusterings* en parallèle sur chaque site, notamment si il n'est pas possible de réunir les différentes parties des données à analyser en raison de limites de stockage ou de réseau.

le **respect de la confidentialité** des données notamment lorsque les données sont décentralisées selon les variables descriptives ou attributs. Dans ce contexte, il est important que chaque partie des variables ne soit observée que par l'algorithme de *clustering* local employé, et inaccessible des autres algorithmes de *clusterings*. Seul l'information local d'appartenance des individus aux groupes peut alors être utilisé pour obtenir une solution consensus.

La littérature est marquée par la proposition de [Strehl and Ghosh, 2003] qui a permis de bien resituer la problématique du *clustering* d'ensemble dans les contextes applicatifs récents tels que présentés précédemment. La thématique a été par ailleurs considérablement étudiée et les approches, enrichies [Vega-Pons and Ruiz-Shulcloper, 2011]. En réalité le problème tel qu'il est formulé, est adressé depuis bien plus longtemps, notamment par la communauté francophone et les travaux de Simon Régnier [Regnier, 1965] sur la recherche de partition médiane. Ces travaux ont également été réactualisés par la même communauté au travers par exemple, la contribution de [Guénoche, 2011].

le *clustering* collaboratif

Différents chercheurs se sont également intéressés au problème semblable mais dont on peut faire la distinction du *clustering* collaboratif pour lequel on s'autorise à modifier les différents *clusterings* de base du profil afin de les enrichir et d'améliorer leur qualité en les combinant, comme l'ont proposé [Wemmert et al., 2000]. Enfin, d'autres approches ont été développées dans le même esprit afin d'obtenir un ensemble de *clusterings* consensus en combinant les différents *clusterings* de base, tout en assurant une certaine dissimilarité entre les *clusterings* de l'ensemble produit [Faceli et al., 2009]. Cette dernière approche notamment permet d'introduire la deuxième problématique à laquelle les contributions de ce chapitre apportent une solution : l'*alternative clustering*.

l'*alternative clustering*

La problématique de l'*alternative clustering* est la suivante : À partir d'un tableau relationnel ou descriptionnel sur l'ensemble d'individu $\mathcal{X}$, trouver un ensemble de *clusterings* de $\mathcal{X}$ tel que :

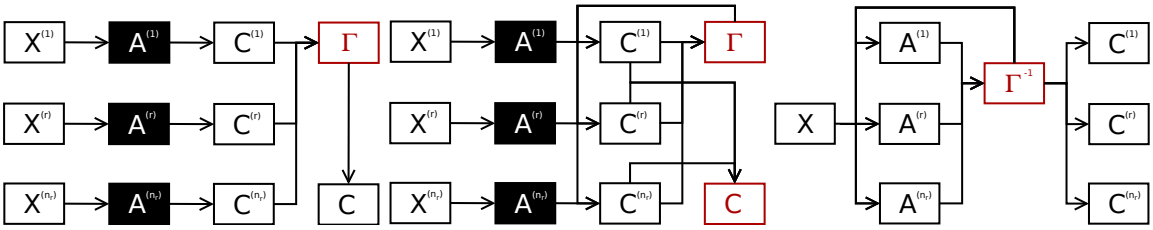

FIGURE 4.1 — Les différents paradigmes du *clustering* d'ensemble, *clustering* collaboratif et *alternative clustering*. Dans l'ordre ci-dessus, (1) la recherche d'un *clustering* consensus (contrôlée par une fonction ou un algorithme de consensus Γ) à partir d'un ensemble de *clusterings* issus d'algorithmes quelconque $A^{(r)}$, (2) la recherche d'un *clustering* consensus à partir d'un mécanisme de collaboration Γ remettant en cause les différents *clusterings* des données et enfin (3) la recherche d'un ensemble de *clusterings* alternatifs contrôlé par une stratégie (fonction ou algorithme) de divergence Γ^{-1}, à partir d'un jeu de donnée mono-vue.

– chaque *clustering* soit de bonne qualité, au sens d'une mesure de qualité usuelle (inertie de KM, vraisemblance pour EM, *etc.*) ;
– chaque *clustering* soit dissimilaire des autres au sens d'une mesure de similarité ou dissimilarité particulière.

Les algorithmes de la famille *alternative clustering* ont pour objectif d'offrir à un utilisateur un plus vaste choix de résultats pour l'analyse exploratoire dans un contexte purement applicatif. Ces approches permettent également d'identifier des structures de groupes différentes et potentiellement intéressantes dans l'analyse de données de grande dimensionnalité.

Les approches se sont majoritairement développées ces dernières années et utilisent des principes aussi vaste que pour le *clustering* simple. Les approches proposées reposent sur des adaptations d'algorithmes de *clustering* hiérarchique [Bae and Bailey, 2006], de modèles de mélanges [Dang and Bailey, 2010] ou bien encore sur des techniques indépendantes de l'algorithme de *clustering* en réalisant un apprentissage de distance adapté [Davidson and Qi, 2008].

Les différentes contributions proposées répondant aux problématiques peuvent être schématisés comme dans la figure 4.1. Les contributions proposées sont des instanciations particulières d'une plateforme générale permettant la combinaison d'algorithmes de *clusterings* et capable de déterminer :
– un *clustering* consensus pour des données multi-vues, ou pour des données mono-vue explorées par des algorithmes différents ainsi qu'un ensemble de distances adaptées ou différentes combinaison linéaires des variables descriptives des données permettant d'atteindre ce consensus ;
– un ensemble de *clusterings* alternatifs pour un jeu de données mono-vue, ou éventuellement multi-vues, ainsi que les distances ou combinaisons linéaires des variables descriptives correspondantes.

L'approche proposée est générique et ne nécessite pas de connaître les algorithmes de *clusterings* employés. De plus, contrairement à la quasi-intégralité des méthodes présentées précédemment, elle exploite les représentations vectorielles des individus lorsqu'elles sont disponibles. Enfin, elle se décline en deux versions, CoBoC et AlterBoC répondant aux deux problématiques posées.

4.3 Approches de type ensemble de *clusterings*

4.3.1 *Clustering* consensus par ensemble de *clusterings*

L'approche de *clustering ensemble* (CE) [Strehl and Ghosh, 2003] est une approche algorithmique, conçue pour obtenir un *clustering* unique consensus à partir d'un profil de partitions $\Pi = \{C^{(r)}\}_{r\in[1..n_r]}$ d'un même ensemble d'individus $\mathcal{X}$. Les auteurs proposent à la fois une mesure de comparaison entre *clusterings* fondée sur des éléments de théorie de l'information : l'information mutuelle normalisée, qu'un moyen heuristique d'optimiser un critère reposant sur cette comparaison pour trouver le *clustering* consensus .

Objectif

L'objectif est de construire un *clustering* C^* des individus, le plus proches possible de chaque partition du profil Π, au sens de l'information mutuelle normalisée (cf. section 1.5.3.2) :

$$C^* = arg\,\underset{C}{max}\, Q_{\text{CE}}(C, \Pi)$$

Avec

$$Q_{\text{CE}}(C, \Pi) = \frac{1}{n_r}\sum_{r=1}^{n_r} NMI(C, C^{(r)})$$

Soit Γ l'heuristique permettant de trouver un optimum du critère précédent. Les auteurs proposent trois heuristiques différentes correspondant à Γ : CSPA, HGPA et MCLA. Ceux-ci détermine l'algorithme appliqué (algorithme 27).

Algorithme

CSPA. La première heuristique développée consiste à compter en moyenne pour chaque paire d'individu $(x_i, x_j) \in \mathcal{X}^2$, le nombre de fois où ceux-ci sont regroupés parmi toutes les partitions disponibles. Ainsi, les valeurs obtenues sont comprises entre 0 et 1 et la fonction associée se comporte comme une mesure de similarité K, une forte valeur de K_{ij} correspondant au fait que x_i et x_j soient fréquemment regroupés dans les différents *clusterings* du profil. Soit $H^{(r)}$ la matrice du r-ième *clustering*, la fonction K de similarité ainsi produite est définie par :

$$K^{(r)} = \frac{1}{n_r}\sum_{r=1}^{n_r} H^{(r)}$$

Une fois ces valeurs de similarité établies entre les individus, les auteurs proposent d'appliquer un algorithme de *clustering* adapté capable de produire un unique *clustering* à partir d'une matrice de similarité, comme l'algorithme METIS [Karypis and Kumar, 1998], adapté au partitionnement de graphes en groupes de tailles homogènes.

HGPA. La seconde heuristique développée propose de construire un hyper-graphe à partir des différentes partitions. Dans chaque partition, chaque groupe $C_k^{(r)}$ correspond à une hyper-arête qui relie simultanément les individus membres de ce groupe. Dans l'hyper-graphe, un individu x_i est alors relié *via* r hyper-arêtes, à n_r groupes potentiellement différents. L'objectif de HGPA est alors, à partir de l'hyper-graphe, d'identifier un nombre minimal d'hyper-arêtes à enlever afin de déconnecter l'hyper-graphe en n_k groupes disjoints, éliminant ainsi les recouvrements induits par l'appartenance de certains individus à des groupes différents dans chaque vue. Ce problème est résolu *via* les approches de *clustering* d'hyper-graphe. Les auteurs proposent d'utiliser pour ce faire l'algorithme HMETIS.

MCLA. La dernière heuristique proposée correspond à une approche algorithmique de *clustering* de groupes. L'objectif est d'identifier parmi les différents groupes présents dans toutes les partitions ceux qui sont proches, et de les regrouper par *clustering* afin de déterminer globalement les k *meta*-groupes les plus représentatifs. De plus, les auteurs proposent un moyen de définir pour chaque *meta*-groupe M_k ainsi déterminé et chaque individu x_i de ce groupe, la contribution de x_i à la définition de M_k.

De manière plus détaillée, l'approche MCLA est séparable en quatre étapes que sont :

1. La construction d'un *meta*-graphe, dans lequel les sommets correspondent aux différents groupes $C_k^{(r)}$ présents dans les différentes partitions $C^{(r)}$ et les arêtes reflètent une similarité entre groupes. La similarité proposée par les auteurs est l'indice de Jaccard (1.20) qui mesure, dans ce contexte, pour deux groupes donnés $C_k^{(r)}$ et $C_{k'}^{(r)}$ (deux sommets), la proportion de paires d'individus présents simultanément dans ces deux groupes :

$$K(C_k^{(r)}, C_{k'}^{(r')}) = Jaccard(C_k^{(r)}, C_{k'}^{(r')})$$

   En particulier, les *clusterings* $C^{(r)}$ étant supposés stricts, on a l'égalité suivante :

$$K(C_k^{(r)}, C_{k'}^{(r')}) = 0 \; \forall r \in [1..n_r], \; \forall k \neq k'$$

2. Le *clustering* du *meta*-graphe permet quant à lui d'identifier k *meta*-groupes représentatifs des différents groupes des individus issus de toutes les partitions du profil. L'idée étant d'identifier la correspondance entre les groupes dans les différentes partitions. En ce sens, deux groupes en forte correspondance issus de deux partitions différentes devraient appartenir à un même *meta*-groupe. Cette correspondance est directement déduite de la mesure de Jaccard et le *clustering* est réalisé au moyen de l'algorithme METIS. On obtient alors un *meta-clustering* M qui est une partition de l'ensemble $\bigcup_{r \in [1..n_r]} C^{(r)}$.
3. La consolidation des *meta*-groupes permet de redéfinir ces *meta*-groupes proprement comme des *meta*-hyper-arêtes correspondantes aux différents groupes du *meta*-groupe (la consolidation est réalisée par ajout d'hyper-arêtes). Chaque *meta*-groupe M_k est associé à un vecteur de contributions des individus $x_i \in \mathcal{X}$ à la définition de ce *meta*-groupe. Soit $H^{(r)} \in \{0,1\}^{n \times n_k}$ la matrice indiquant pour chaque individu x_i le groupe auquel il appartient :

$$H_{ik}^{(r)} = \begin{cases} 1 & \text{si } x_i \in C_k^{(r)} \\ 0 & \text{sinon} \end{cases}$$

   Cette contribution u_{ik} de l'individu x_i au *meta*-groupe M_k est obtenue par :

$$u_{ik} = \frac{1}{n_r |M_k|} \sum_{r=1}^{n_r} \sum_{C_{k'}^{(r)} \in M_k} Z_{ik'}^{(r)}$$

4. L'affectation des individus afin d'obtenir le *clustering* consensus C final est réalisée selon les valeurs de contributions déterminées à l'étape précédente. Ainsi, si l'on s'autorise à interpréter les valeurs de contributions comme des probabilités *a posteriori*, la règle MAP est alors appliquée. Autrement dit les individus sont effectivement affectés au *meta*-groupe pour lequel sa contribution est la plus importante :

$$x_i \in C_k \Leftrightarrow k = \underset{k' \in [1..n_k]}{arg\ max}\ u_{ik'}$$

Enfin, les auteurs proposent de définir, pour une meilleure interprétabilité des résultats, une confiance pour chaque affectation des individus. Ainsi cette confiance s'exprime comme la valeur de contribution au groupe auquel l'individu est affecté, relativement à toutes les autres valeurs de contribution de cet individu :

$$\alpha_i = \frac{u_{ik}}{\sum_{\substack{\overline{k}=1 \\ \overline{k}\neq k}}^{n_k} u_{i\overline{k}}} \; \forall x_i \in C_k$$

Algorithme 27 CE

ENTRÉES : $\mathcal{X}$, n_k, Γ
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Génération de $\{C^{(r)}\}_{r\in[1..n_r]}$ par n_r *clusterings* différents de $\mathcal{X}$
2 : $C = \Gamma(\{C_1^{(1)}, \ldots, C_{n_k^{(1)}}^{(1)}, \ldots, C_1^{(n_r)}, \ldots, C_{n_k^{(n_r)}}^{(n_r)}\})$

Discussion

L'apport de l'approche de *clustering ensemble* réside essentiellement dans les heuristiques de combinaisons de partitions. MCLA semble correspondre au meilleur compromis entre la qualité du consensus obtenu au sens de l'information mutuelle normalisée et l'efficacité au sens de la complexité algorithmique ($\mathcal{O}(n.n_k^2.n_r^2)$). L'heuristique HGPA est la plus efficace en complexité algorithmique ($\mathcal{O}(n.n_k.n_r)$) mais peine à être efficace dans l'obtention d'une solution consensus. CSPA est l'heuristique la plus complexe ($\mathcal{O}(n^2.n_k.n_r)$) mais est aussi efficace que MCLA et offre une flexibilité dès lors que l'on s'autorise à utiliser un autre algorithme de *clustering* que METIS.

Finalement l'inconvénient majeur que l'on peut formuler est que les *clusterings* de l'ensemble ne sont jamais remis en question pour faciliter l'obtention d'une meilleure solution consensus et les heuristiques proposées n'utilisent pas, même localement les variables descriptives si elles existent.

4.3.2 Consensus de partitions

Parmi les premières approches cherchant à obtenir un *clustering* consensus à partir d'un ensemble de *clusterings* ou partitions de base figurent celles dédiées à la problématique de partition médiane ou partition centrale. Cette problématique fut étudiée très tôt dans la communauté francophone de classification notamment par Simon Régnier [Regnier, 1965] et reprise et développée plus récemment dans les travaux d'Alain Guénoche [Guénoche, 2011].

Objectif

Le problème est posé comme la recherche d'une solution optimale à un problème d'optimisation défini informellement comme la recherche d'un nouveau *clustering* des individus proche, selon une mesure de similarité S particulière, de tous les *clusterings* présents dans l'ensemble. Formellement le *clustering* consensus est défini comme l'optimum du critère objectif :

$$\max_C Q'_{\text{FT}}(C, \Pi) = \max_C \sum_{r=1}^{n_r} S(C, C^{(r)}) \tag{4.1}$$

où $S(C, C^{(r)}) = \frac{n(n-1)}{2} - |\Delta(C, C^{(r)})|$ et $\Delta(C, C^{(r)})$ est la distance des différences symmétriques entre les *clusterings* C et $C^{(r)}$. Soit $H^{(r)}$ la matrice des résultats du r-ième *clustering* de l'ensemble (que l'on supposera être le résultat d'un algorithme $A^{(r)}$) :

$$H_{ij}^{(r)} = \begin{cases} 1 & \text{si } Link^{(r)}(x_i, x_j) \\ 0 & \text{si } \overline{Link}^{(r)}(x_i, x_j) \end{cases} \tag{4.2}$$

Soit H la matrice des hypothèses du *clustering* consensus en construction, la distance des différences symmétriques revient à compter le nombre de paires d'individus $(x_i, x_j) \in \mathcal{X}^2$ pour lesquelles les hypothèses de *clusterings* H_{ij} et $H_{ij}^{(r)}$ sont différentes. Le critère (4.1) est équivalent en maximisation au critère Q_{FT} défini par :

$$Q_{\text{FT}}(H) = H_{ij} \sum_{(x_i,x_j)\in\mathcal{X}^2} \left(\left(\sum_{r=1}^{n_r} H_{ij}^{(r)} \right) - \frac{n_r}{2} \right) \tag{4.3}$$

$$\tag{4.4}$$

Soit $W_{ij} = \left(\left(\sum_{r=1}^{n_r} H_{ij}^{(r)} \right) - \frac{n_r}{2} \right)$, le problème d'optimisation peut alors être posé :

$$\begin{aligned} \max_H Q_{\text{FT}}(H) = & \max_H \sum_{\substack{(x_i,x_j)\in\mathcal{X}^2 \\ i\leq j}} H_{ij} W_{ij} \\ s.c. \quad & H_{ij} \in \{0,1\} \qquad \forall (x_i, x_j)_{i\leq j} \in \mathcal{X}^2 \\ & H_{ij} + H_{jk} - H_{ik} \leq 1 \quad \forall (x_i, x_j, x_k)_{i\neq j\neq k} \in \mathcal{X}^3 \end{aligned} \tag{4.5}$$

Algorithme

Les auteurs proposent de résoudre ce problème par un algorithme adapté (algorithme 28), FUSION-TRANSFERT (FT), composé de deux étapes. L'étape de fusion fait appel à une heuristique et s'inspire du principe de classification ascendante hiérarchique AGNES (cf. section 1.2.2) pour lequel le critère d'arrêt n'est pas l'obtention de la partition à 1 groupe contenant tous les individus, mais l'atteinte d'une partition maximale selon le critère Q_{FT}. Ainsi, partant de la partition atomique correspondant à l'ensemble des singletons d'individus, le principe est de fusionner à chaque étape les deux groupes ou amas tels que l'amélioration du critère soit maximum. Partant de $A_i = \{x_i\}$ et $\mathcal{D}_0 = \{A_i\}_{i\in[1..n]}$. $\mathcal{D}$ est la structure de dendrogramme associée à la classification hiérarchique.

Soit $W(A_k) = \sum_{(x_i,x_j)\in A_k^2} W_{ij}$, et soit un *clustering* de $\mathcal{X}$ en n_k amas, le critère Q_{FT} peut alors être réécrit :

$$Q_{\text{FT}}(A_1, \ldots, A_{n_k}) = \sum_{k=1}^{n_k} W(A_k) \tag{4.6}$$

Soit ζ_i l'ensemble des paires d'amas candidates pour la fusion :

$$\zeta_i = \{(A_k, A_{k'}) \in \mathcal{D}_{i-1}^2 \mid (W(A_k \cup A_{k'}) - (W(A_k) + W(A_{k'}))) \geq 0\}$$

ζ_i est l'ensemble des paires d'amas de $\mathcal{D}_{i-1}$ qui apporte un gain au critère Q_{FT}. La règle permettant d'obtenir le *clustering* correspondant à $\mathcal{D}_i$ et maximisant Q_{FT} est la suivante :

$$(A_k, A_{k'}) = \underset{(A_l,A_{l'})\in\zeta_i}{arg\ max}\, W(A_l \cup A_{l'}) \Rightarrow \mathcal{D}_i = \mathcal{D}_{i-1} \setminus (A_k, A_{k'}) \cup \{A_k \cup A_{k'}\} \tag{4.7}$$

Ce principe de fonctionnement est simple et en général efficace, mais il souffre du problème bien connu des approches de classification hiérarchique qui est la non remise en cause des fusions réalisées.

Pour outre-passer ce défaut et améliorer la qualité de la partition consensus, l'étape de transfert propose de déplacer certains éléments susceptibles d'améliorer Q_{FT}. On calcul pour ce faire un nouveau poids u_{ik} pour chaque individu x_i et chaque groupe C_k déterminé à l'issue du processus de fusion, selon l'équation suivante :

$$u_{ik} = \sum_{x_j \in C_k} W_{ij} \tag{4.8}$$

Ainsi u_{ik} modélise bien la contribution de l'individu x_i au groupe C_k. En particulier, si $x_i \in C_k$, u_{ik} correspond à la contribution de x_i à la valeur du critère Q_{FT}. De la même façon, on définit pour chaque individu $x_i \in C_k$ un gain de transfert Δ de C_k à $C_{k'}$ par la formule :

$$\Delta(x_i, C_k, C_{k'}) = u_{ik'} - u_{ik} \tag{4.9}$$

La procédure de transfert consiste alors à déplacer parmi tous les individus, celui qui maximise le plus son éventuel gain de transfert, dont les différents paramètres optimaux sont définis formellement par :

$$(C_k^*,\ C_{k'}^*,\ x_i^*) = \underset{(k,k') \in [1..n_k]^2,\ x_i \in C_k}{arg\ max} \Delta(x_i, C_k, C_{k'}) \tag{4.10}$$

Ainsi deux cas peuvent se produire :

- le gain maximum de transfert est positif ou nul, auquel cas on transfère effectivement l'individu x_i^* du groupe C_k^* au groupe $C_{k'}^*$:

$$\Delta(x_i^*, C_k^*, C_{k'}^*) \geq 0 \Rightarrow \Big(((C_k^* = C_k^* \setminus \{x_i^*\}) \wedge (C_{k'}^* = C_{k'}^* \cup \{x_i^*\})) \Big) \tag{4.11}$$

- le gain maximum de transfert est négatif, auquel cas on transfère l'individu x_i^* du groupe C_k^* à un nouveau groupe $C_{k''}$:

$$\Delta(x_i^*, C_k^*, C_{k'}^*) < 0 \Rightarrow \Big(((C_k^* = C_k^* \setminus \{x_i^*\}) \wedge (C_{k''} = \{x_i^*\})) \Big) \tag{4.12}$$

Algorithme 28 FT

ENTRÉES : $\mathcal{X}$, $\{C^{(r)}\}_{r \in [1..n_r]}$
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialiser $A_i = \{x_i\}$ et $\mathcal{D}_0 = \{A_i\}_{i \in [1..n]}$
2 : Application AGNES sur $\mathcal{X}$ en utilisant la règle (4.7) pour obtenir C
3 : Déterminer $(C_k^*,\ C_{k'}^*,\ x_i^*)$ selon (4.10)
4 : Transférer x_i^* selon (4.11) ou (4.12)

Discussion

L'algorithme FT est une approche heuristique permettant d'atteindre un *clustering* consensus formulé comme la recherche de la partition médiane de l'ensemble ou profil des *clusterings* de base. L'approche a comme défaut de reposer sur un algorithme hiérarchique qui ne permet pas à lui seul de corriger la construction d'une mauvaise hiérarchie menant à un mauvais *clustering* au dernier niveau du dendrogramme mais atteignant un optimum du critère Q_{FT}. Ce défaut est corrigé par la procédure de transfert, mais l'ensemble des deux procédures mène à une approche complexe ($\mathcal{O}(n_r.n^2) + \mathcal{O}(n^3)$).

À l'instar de CE, FT se place dans un cadre où les *clusterings* de base ne sont jamais remis en question et de plus les variables descriptives des individus, si elles existent, ne sont pas exploitées.

4.4 Approches collaboratives

4.4.1 SAMARAH : système d'apprentissage multi-agents de raffinement automatique de hiérarchies

La méthode SAMARAH [Wemmert et al., 2000] est une approche essentiellement algorithmique qui a pour objectif de trouver un consensus entre plusieurs méthodes de *clustering* à travers un mécanisme contrôlé de collaboration entre ces différentes méthodes. L'objectif affiché est l'amélioration de la robustesse d'une solution de *clustering* en minimisant l'impact du choix d'une méthode de *clustering* particulière ou de ses paramètres.

Algorithme

SAMARAH (algorithme 29) repose sur différentes étapes :

- la génération de *clusterings* initiaux qui consiste à obtenir différents *clusterings* à partir d'un même jeu de données. Les auteurs proposent dans leur contexte d'appliquer différentes méthodes de *clusterings* ou une même méthode de *clustering* avec des paramètrages différents ;
- le raffinement des résultats qui a pour but d'identifier des conflits et de les résoudre. Ces conflits correspondent à des différences observées entre les *clusterings* produits, décidées à partir de l'évaluation d'une similarité entre ces derniers. À l'issue du raffinement, les différentes partitions sont supposées devenir plus similaires entre elles, et chacune peut alors être considérée comme une partition consensus ;
- La combinaison des résultats qui cherche à déterminer une solution unique de *clustering* à partir des différentes partitions raffinées. Cette étape correspond alors pleinement à la problématique de *clustering ensemble*.

L'étape la plus importante est la seconde puisque c'est elle qui fait intervenir le mécanisme de collaboration Γ entre les différentes méthodes de *clusterings*. Les résultats de *clustering* et les distributions des objets au sein des groupes des différents résultats sont comparés *via* les matrices de confusion $M \in \mathbb{N}^{n_k \times n_k}$ pour tout couple de groupes issus de *clusterings* différents. Cette matrice permet d'observer globalement les différences deux à deux entre *clusterings*. Elle est définie par :

$$M_{kk'}^{(r)(r')} = \frac{|C_k^{(r)} \cap C_{k'}^{(r')}|}{|C_k^{(r)}|} \; \forall (r, r') \in R^2, \; \forall (C_k^{(r)}, C_{k'}^{(r')}) \in C^{(r)} \times C^{(r')}$$

Les auteurs proposent d'utiliser cette matrice de confusion pour établir une mesure de similarité entre deux groupes issus de *clusterings* différents. Cette mesure notée K est définie

par :

$$K(C_k^{(r)}, C_{k'}^{(r')}) = \rho_k^{(r)(r')} M_{k'k}^{(r')(r)} \text{ et } \rho_k^{(r)(r')} = \sum_{k'=1}^{n_k^{(r')}} M_{kk'}^{(r)(r')}$$

Le choix d'une telle mesure de similarité permet de quantifier et d'ordonner les correspondances entre les groupes issus de vues différentes. Notamment, étant donnés le k-ième groupe du *clustering* $C^{(r)}$ et un *clustering* $C^{(r')}$, il est possible de déterminer le meilleur correspondant de $C_k^{(r)}$ parmi les groupes de $C^{(r')}$ par :

$$f^*(C_k^{(r)}, C^{(r')}) = \underset{C_{k'}^{(r')} \in C^{(r')}}{arg\ max}\ K(C_k^{(r)}, C_{k'}^{(r')})$$

À partir de cette correspondance est définit le conflit. Si un groupe ne se retrouve pas complètement dans un *clustering*, *i.e.* $K(C_k^{(r)}, f^*(C_k^{(r)}, C^{(r')})) < 1$, alors il y a conflit. Cette règle permet de définir un ensemble des conflits ζ comme l'ensemble des couples ($C_k^{(r)}$, $C^{(r')}$) tel que le groupe $C_k^{(r)}$ ne soit pas en parfaite correspondance avec un des groupes du *clustering* $C^{(r')}$:

$$\zeta = \{(C_k^{(r)}, C^{(r')}) | r \neq r' \wedge K(C_k^{(r)}, f^*(C_k^{(r)}, C^{(r')})) < 1\}$$

Cet ensemble est muni d'une relation d'ordre pour former une liste qui est traité par l'algorithme de résolution des conflits. La première stratégie proposée par les auteurs consiste à ordonner les couples de l'ensemble par la valeur de similarité entre les groupes et leurs meilleurs correspondants. Plus la similarité entre un groupe et son meilleur correspondant dans un autre *clustering* est faible et plus le conflit est grand. La résolution de ces conflits a alors lieu dans un processus itératif où chaque étape revient à apporter des modifications sur les différentes partitions impliquées dans le conflit courant au travers l'application de trois opérateurs que sont :

- la fusion de groupes : les individus de deux groupes d'un même *clustering* sont réunis dans un seul groupe ;
- la scission d'un groupe : un *clustering* est appliqué aux individus d'un groupe donné ;
- le *reclustering* : un groupe donné est retiré, et les individus de ce groupe sont réaffectés aux autres groupes.

Le choix des opérateurs à appliquer est décidé à l'aide d'un paramètre σ supplémentaire dépendant du nombre de groupes impliqués dans le conflit. En d'autres termes, pour un couple conflictuel ($C_k^{(r)}, C^{(r')}$) donné, le paramètre dépend de la distribution des individus de $C_k^{(r)}$ dans $C^{(r')}$. Ainsi, si les valeurs de similarité caractérisant ce couple sont plus grandes que le paramètre σ : $K(C_k^{(r)}, C_{k'}^{(r')}) \geq \sigma\ \forall C_{k'}^{(r')} \in C^{(r')}$, alors $C_{k'}^{(r')}$ est considéré comme un bon contributeur pour la correspondance.

Si il n'y a pas de bons contributeurs pour le conflit ($C_k^{(r)}, C^{(r')}$) alors l'opérateur de *reclustering* est appliqué sur $C_k^{(r)}$. En revanche, soit m le nombre de bons contributeurs pour $C_k^{(r)}$ dans $C^{(r')}$, les auteurs proposent de construire les clusterings $C'^{(r)}$ et $C'^{(r')}$ tels que :

- $C'^{(r)}$ corresponde à $C^{(r)}$ où le groupe $C_k^{(r)}$ est scindé en m ;
- $C'^{(r')}$ corresponde à $C^{(r')}$ où les m bons contributeurs sont fusionnés.

Les auteurs proposent alors deux fonctions de qualité, non retranscrites ici, locale et globale pour décider de l'application effective des opérateurs. La fonction de qualité locale, permet de trouver la paire de *clusterings* optimale ($C^{*(r)}, C^{*(r')}$) parmi les paires ($C^{(r)}, C^{(r')}$), ($C^{(r)}, C'^{(r')}$), ($C'^{(r)}, C^{(r')}$) et ($C'^{(r)}, C'^{(r')}$). La paire optimale obtenue implique une mise à jour des *clusterings* correspondant. Cependant cette mise à jour n'est effective que selon le comportement de la fonction de qualité globale. Ainsi :

- si la résolution locale (entre deux vues) du conflit améliore la qualité globale, alors la mise à jour est réalisée et les conflits sont recalculés ;
- si $(C^{*(r)}, C^{*(r')}) = (C^{(r)}, C^{(r')})$, alors le conflit n'a pas d'intérêt et est retiré de la liste à résoudre ;
- si la résolution locale du conflit déteriore la qualité globale, celui-ci est résolu, sous réserve qu'une amélioration de la qualité globale soit observée au plus après la résolution de la moitié des conflits restants.

Pour finir, et même si chaque partition raffinée est issue d'une procédure collaborative tendant vers un consensus, une combinaison des résultats raffinés est réalisée par une procédure de vote entre les différents algorithmes de *clusterings* locaux, sur le meilleur groupe correspondant à chaque individu. Ceci afin d'obtenir une unique partition consensus, dans la suite des approches de *clustering* d'ensemble.

Algorithme 29 SAMARAH

ENTRÉES : $\mathcal{X}$, σ
SORTIES : $C = \{C_1, ..., C_{n_k}\}$
1 : Générer n_r *clusterings* $\{C^{(r)}\}_{r\in[1..n_r]}$ à partir de $\mathcal{X}$
2 : Raffiner itérativement chaque $C^{(r)}$ en résolvant les conflits
3 : $C = Vote(\{C^{(r)}\}_{r\in[1..n_r]})$

Discussion

L'approche SAMARAH, illustrative des approches de *clustering* collaborative, se distingue des approches de type *clustering ensemble* par la remise en cause des partitions du profil *via* l'étape de raffinement des résultats. L'approche permet de concilier plusieurs partitions en nombre de groupes différents. Elle possède également les différents avantages des approches de *clustering ensemble* que sont la réutilisation des connaissances, la décentralisation des calculs et le respect de la confidentialité des données. Néanmoins, à l'instar des approches précédentes, elle n'utilise pas les descriptions des individus lorsqu'elles sont disponibles. L'approche SAMARAH a également été étendue dans le cadre de la thèse de [Forestier, 2010], par l'ajout de nouvelles stratégies de résolution de conflits et également par la prise en compte de connaissances externes pour guider la recherche d'un clustering consensus par semi-supervision.

4.4.2 MOCLE : *clustering* d'ensemble multi-objectif

L'approche multi-objectif pour le *clustering ensemble* MOCLE proposée par [Faceli et al., 2009] vise à produire non pas un *clustering* consensus mais un ensemble de *clusterings* consensus. À partir d'un ensemble de *clusterings* initiaux, les auteurs proposent d'appliquer un algorithme génétique permettant de maintenir à chaque itération ou génération un tel ensemble de *clusterings*.

Il s'agit d'une approche principalement algorithmique (algorithme 30) qui se décline ainsi en deux étapes que sont :
- la génération de partitions de base réalisée de la même manière que pour l'approche SAMARAH;
- la recherche d'un ensemble de partitions consensus différentes réalisant chacune un compromis particulier de plusieurs critères objectifs.

L'apport principal de cette approche réside dans la seconde étape qui fait appel à deux opérateurs, croisement et sélection, permettant de faire évoluer la population de solutions potentielles (les différentes partitions de base) vers l'objectif visé.

L'opérateur de croisement permet, à partir d'une paire de partitions de la population, d'obtenir une nouvelle partition consensus. Les paires de partitions sont sélectionnées aléatoirement selon le principe de tournoi binaire. Même si l'approche MOCLE vise à offrir un paradigme très généraliste pour la production de plusieurs partitions consensus, il est nécessaire de spécifier effectivement cet opérateur de croisement. Les auteurs proposent d'utiliser l'algorithme MCLA (cf. section 4.3.1). Les nouvelles partitions sont alors ajoutées à la population existante.

Le deuxième opérateur a pour but de limiter la taille de la population, afin d'éviter de maintenir une sous population de faible qualité. Ainsi les auteurs proposent de définir différents critères permettant d'identifier les partitions de bonne qualité. L'opérateur de sélection consiste à déterminer, parmi les partitions de la population, celles qui approximent le mieux le front de Pareto correspondant aux optima de ces différents critères.

Les critères proposés pour évaluer chaque partition $C^{(r)}$ sont (1) l'inertie $Q_{inrt}^{(r)}$ (à minimiser) ainsi que (2) sa connectivité $Q_{con}^{(r)}$ (à minimiser). L'inertie de la partition $C^{(r)}$ est définie comme une somme des inerties intra-groupes par le critère correspondant à celui de KM :

$$Q_{inrt}^{(r)} = \sum_{k=1}^{n_k^{(r)}} \sum_{x_i \in C_k^{(r)}} ||x_i^{(r)} - c_k^{(r)}||_2^2 \tag{4.13}$$

La connectivité est mesurée en observant le nombre de fois ou deux individus voisins se retrouvent dans un même groupe :

$$Q_{con}^{(r)} = \sum_{x_i \in \mathcal{X}} \sum_{j=1}^{n-1} \delta^{(r)}(x_i, \mathcal{N}_j^{(r)}(x_i)) \tag{4.14}$$

où $\mathcal{N}_j^{(r)}(x_i)$ correspond au j-ième plus proche voisin de x_i dans la partition $C^{(r)}$ et $\delta^{(r)}$ est définit par :

$$\delta^{(r)}(x_i, \mathcal{N}_j^{(r)}(x_i)) = \begin{cases} \frac{1}{j} & \text{si } \overline{Link}^{(r)}(x_i, \mathcal{N}_j^{(r)}(x_i)) \\ 0 & \text{si } Link^{(r)}(x_i, \mathcal{N}_j^{(r)}(x_i)) \end{cases}$$

Ainsi, la connectivité est nulle (minimale) lorsque tous les voisins de chaque individu (pour un voisinage de taille arbitrairement grand) sont regroupés avec celui-ci *i.e.* lorsqu'il n'y a qu'un seul groupe.

L'objectif est alors d'identifier parmi les partitions de la génération courante, celles qui optimisent (minimisent) simultanément ces deux critères.

Discussion

On remarque que sans contraintes sur le nombre de groupes présents dans une partition donnée, le critère inertiel $Q_{inrt}^{(r)}$ favorisera la solution dégénérée de partition atomique *i.e.* en singletons, alors que le critère de connectivité $Q_{con}^{(r)}$ favorisera la solution dégénérée d'un seul groupe contenant tous les individus. Néanmoins les auteurs proposent de contraindre la taille des groupes notamment lors de l'application de l'opérateur de croisement, le nombre de groupes

Algorithme 30 MOCLE

ENTRÉES : $\mathcal{X}$, t_f
SORTIES : $\{C^{(r)}\}_{r\in[1..n_r]}$
1 : Générer n_r *clusterings* $\{C^{(r)}\}_{r\in[1..n_r]}$ à partir de $\mathcal{X}.t = 0$
2 : Appliquer le croisement sur $\{C^{(r)}\}_{r\in[1..n_r]}$
3 : Augmenter $\{C^{(r)}\}_{r\in[1..n_r]}$ avec les résultats du croisement
4 : Sélectionner les *clusterings* dominants au sens des critères (4.13) et (4.14)
5 : Si $t < t_f$ aller en **2**

de la partition consensus obtenue devant être compris entre les valeurs des nombres de groupes des partitions parentes sélectionnées lors du tournoi. On peut formuler néanmoins l'hypothèse que les solutions optimales approximant le front de Pareto seront telles que les partitions de nombre de groupes élevé seront favorisées par le terme d'inertie et inversement les partitions en faible nombre de groupe seront favorisées par le terme de connectivité, dans une autre région du front. Pour finir, sous l'hypothèse que deux *clusterings* issus de deux régions différentes du front (approximant des optimums différents des critères) sont des solutions de *clustering* différentes, alors MOCLE permet l'obtention de partitions alternatives entre elles. Ceci permet d'élargir l'analyse exploratoire pour diversifier l'interprétation des résultats. La problématique spécifique de la recherche de partitions alternatives est l'objet de la prochaine section.

4.5 Approches alternatives

Le but des approches de *clustering* alternatif est d'obtenir un ensemble de *clusterings* en adéquation avec la distribution naturelle des individus et différents les uns par rapport aux autres. La première condition est appelée critère de qualité et le second est un critère de dissimilarité. Ainsi le compromis recherché (à maximiser) peut être exprimé simplement sous la forme générale suivante :

$$\text{clustering alternatif} = \sum_{r=1}^{n_r} \text{objectif local}(r) + \text{désaccord}(\Pi) \tag{4.15}$$

La forme générale laisse apparaître un formalisme proche du *clustering* multi-vues (2.1), mais cette fois le désaccord est recherché et donc, à maximiser et non pas à minimiser.

L'objectif est d'apporter à un utilisateur différentes analyses d'un même jeu de donnée lors d'une réelle analyse exploratoire, afin de permettre la découverte de motifs différents mais cohérents, dans les données.

4.5.1 COALA : *clustering* hiérarchique alternatif

L'approche COALA [Bae and Bailey, 2006] considère un premier *clustering* $C^{(1)}$ de $\mathcal{X}$ fixé. Elle vise à répondre au problème posé comme la recherche d'un *clustering* $C^{(2)}$ différent de $C^{(1)}$ par une approche purement algorithmique se fondant sur les méthodes agglomératives hiérarchiques.

Algorithme

L'algorithme utilisé est le clustering par lien moyen ALINK (cf. section 1.2.2). Partant des amas singletons $A_i = \{x_i\}$ et $\mathcal{D}_0 = \{A_i\}_{i\in 1..n}$ avec $\mathcal{D}$ la structure de dendrogramme associée

à la classification hiérarchique. Soit D la mesure de distance entre amas, les deux amas les moins distants sont successivement fusionnés dans un processus itératif jusqu'à atteindre un amas contenant l'ensemble des individus. Les auteurs proposent de biaiser la construction du dendrogramme $\mathcal{D}$ en utilisant les connaissances du *clustering* $C^{(1)}$ avec l'objectif d'obtenir un *clustering* $C^{(2)}$ dissimilaire.

L'approche suit alors plusieurs étapes pour répondre à cet objectif :

1. la **génération de contraintes** consiste à construire des contraintes de type $\mathcal{CL}$, pour toute paire d'individus appartenant au même groupe dans $C^{(1)}$, plus formellement :

$$\mathcal{CL} = \{(x_i, x_j) \in \mathcal{X}^2 \mid Link^{(1)}(x_i, x_j)\}$$

   Autrement dit, les contraintes traduisent l'inverse du résultat de $C^{(1)}$. L'algorithme COALA a pour objectif de satisfaire les contraintes $(x_i, x_j) \in \mathcal{CL}$ *i.e.* ne pas regrouper x_i et x_j déja ensemble dans $C^{(1)}$.

2. la **génération de candidats** à l'agglomération permet d'identifier simultanément deux paires d'amas qui sont susceptibles d'être regroupés à une itération particulière de l'algorithme hiérarchique. Soit ζ_i l'ensemble des paires d'amas candidates pour la fusion :

$$\zeta_i = \{(A_k, A_{k'}) \in \mathcal{D}_i^2\}$$

   et ζ_i^+ l'ensemble des paires d'amas candidates pour la fusion telles que la fusion de ces amas ne violerai aucune contrainte $\mathcal{CL}$:

$$\zeta_i^+ = \{(A_k, A_{k'}) \in \mathcal{D}_i^2 \mid \forall (x_i, x_j) \in A_k \times A_{k'},\ (x_i, x_j) \notin \mathcal{CL}\}$$

   On note :
   d^- : la distance entre les amas (A_i^*, A_j^*) les moins distants :

$$d^- = \underset{(A_i, A_j) \in \zeta_i}{min} D(A_i, A_j)$$

   d^+ : la distance entre les amas (B_i^*, B_j^*) les moins distants satisfaisant les contraintes $\mathcal{CL}$:

$$d^+ = \underset{(B_i, B_j) \in \zeta_i^+}{min} D(B_i, B_j)$$

3. la **détermination du candidat** permet de décider effectivement laquelle des deux paires candidates choisir afin d'atteindre l'objectif. Une première stratégie employable est de systématiquement choisir les paires d'amas distants de d^-. Ceci permet d'atteindre l'objectif de qualité mais ne tient pas du tout compte du *clustering* $C^{(1)}$, ainsi l'objectif de dissimilarité n'est pas atteint. De manière duale, une seconde stratégie consiste à toujours fusionner les paires d'amas distants de d^+ permettant cette fois de réaliser le critère de dissimilarité, mais non le critère de qualité. Ainsi les auteurs proposent d'introduire un nouveau paramètre σ, et modulent la décision en observant le ratio entre les valeurs de distances d^- et d^+ :

$$\frac{d^-}{d^+} < \sigma \quad \rightarrow \quad \mathcal{D}_i = \mathcal{D}_{i-1} \setminus A_i^* \setminus A_j^* \cup (A_i^* \cup A_j^*) \tag{4.16}$$

$$\frac{d^-}{d^+} \geq \sigma \quad \rightarrow \quad \mathcal{D}_i = \mathcal{D}_{i-1} \setminus B_i^* \setminus B_j^* \cup (B_i^* \cup B_j^*) \tag{4.17}$$

Ainsi selon les valeurs de σ le compromis entre les deux objectifs de dissimilarité et de qualité peut être atteint.

Algorithme 31 COALA

ENTRÉES : $\mathcal{X}$, $C^{(1)}$, $n_k^{(2)}$, σ
SORTIES : $C^{(2)}$
1 : Construction de $\mathcal{CL}$ selon §1
2 : $C^{(2)}$ = Appliquer AGNES sur $\mathcal{X}$ selon les règles (4.16) et (4.17)

Discussion

Les auteurs ne proposent pas de moyens automatiques pour estimer la meilleure valeur du paramètre σ. Une proposition pour fournir un ensemble de *clusterings* alternatifs consiste à appliquer récursivement COALA, puis enrichir les contraintes $\mathcal{CL}$ à chacune de ces applications. Cette proposition est limitée car un trop grand nombre de *clusterings* alternatifs entraînera une dégradation inévitable de la qualité.

4.5.2 ADFT : apprentissage de distance alternative

L'approche ADFT (*Alternative Distance Function Transformation*) [Davidson and Qi, 2008] permet de générer deux *clusterings* alternatifs $C^{(1)}$ et $C^{(2)}$ de $\mathcal{X}$. $C^{(1)}$ est obtenu classiquement par application d'un algorithme de *clustering* quelconque A. L'apport principal de l'approche est alors de proposer un algorithme simple et intuitif pour garantir l'obtention du *clustering* $C^{(2)}$ alternatif à $C^{(1)}$.

Algorithme

ADFT (algorithme 32) est composée de cinq étapes :

1. la génération du premier *clustering* $C^{(1)}$;
2. la caractérisation de $C^{(1)}$ par génération d'un ensemble de contraintes $\mathcal{ML}$ et $\mathcal{CL}$ en adéquation avec $C^{(1)}$, et apprentissage d'une nouvelle fonction de distance $d^{(1)}$ à partir de l'ensemble des individus impliqués dans ces contraintes ;
3. le calcul d'une fonction de distance $d^{(2)}$ alternative à $d^{(1)}$;
4. la transformation de X (matrice représentant les données) en X' en adéquation avec $d^{(2)}$;
5. le *clustering* de $\mathcal{X}$ représenté par X' pour obtenir $C^{(2)}$.

L'étape d'apprentissage de distance caractérisant $C^{(1)}$ est l'application des travaux de recherches de [Xing et al., 2002b] et ne fait pas l'objet d'adaptation particulière dans ADFT. Il n'est pas non plus précisé la manière dont sont générées les contraintes utilisées.

En revanche, en supposant $d^{(1)}(x_i, x_j)$ connue $\forall (x_i, x_j) \in \mathcal{X}^2$, les auteurs proposent un moyen optimal d'obtenir une distance alternative. Soit $D^{(1)}$ la matrice représentant la fonction de distance telle que $d^{(1)}(x_i, x_j) = \sqrt{(x_i - x_j)D^{(1)}(x_i - x_j)^\top}$ où les x_i sont des vecteurs lignes, la décomposition en valeurs singulières de $D^{(1)}$ offre une intuition particulière sur $D^{(1)}$:

$$D^{(1)} = U\Sigma V$$

L'intuition derrière la décomposition SVD est que la transformation réalisée par $D^{(1)}$ peut être décomposée en une succession de trois transformations V, Σ et U interprétables géométriquement :

V décrit *via* ses vecteurs lignes une nouvelle base orthonormée ;

Σ est une matrice diagonale dont les valeurs Σ_{jj} dilatent ($\Sigma_{jj} > 1$), ou compressent ($\Sigma_{jj} < 1$) la j-ième dimension de la nouvelle base V ;

U effectue une rotation des axes *via* ses vecteurs colonnes.

Une distance entre les individus correspond alors à la création d'une nouvelle base orthogonale V dans laquelle l'unité de la dimension est pondérée par les valeurs respectives de la diagonale de Σ et dans laquelle les données sont déplacées par rotation selon U. Partant de cette interprétation de la distance $d^{(1)}$ apprise à partir de $C^{(1)}$, les auteurs proposent de déterminer $d^{(2)}$ en modifiant les altérations des dimensions de la base orthogonale associée à $D^{(1)}$ dans la décomposition SVD. En particulier les dimensions dilatés doivent être compressées, et réciproquement. Les auteurs proposent alors d'utiliser l'inverse de la matrice Σ, ainsi la nouvelle mesure de distance $d^{(2)}$ est définie à partir de sa matrice par :

$$D^{(2)} = U\Sigma^{-1}V \tag{4.18}$$

La transformation de X en une nouvelle représentation alternative X' est obtenue en posant :

$$X' = {D^{(2)}}^{\top} X \tag{4.19}$$

Pour finir, $C^{(2)}$ est obtenu en effectuant un *clustering* de X'.

Algorithme 32 ADFT

ENTRÉES : $\mathcal{X}$, $n_k^{(1)}$, $n_k^{(2)}$, A
SORTIES : $C^{(1)}, C^{(2)}$
1 : $C^{(1)} =$ appliquer A sur $\mathcal{X}$ représenté par X
2 : Calcul de $D^{(1)}$
3 : Calcul de $D^{(2)}$ selon (4.18)
4 : Calcul de la nouvelle représentation X' par (4.19)
5 : $C^{(2)} =$ appliquer A sur $\mathcal{X}$ représenté par X'

Discussion

L'apport principal de cette approche est de considérer l'obtention de solutions alternatives en extrayant des contraintes $\mathcal{ML}$ et $\mathcal{CL}$ à partir d'un premier *clustering* optimal pour les données. Néanmoins, l'approche ADFT est conçue pour trouver uniquement deux *clusterings* et ne semble pas être extensible dans le même esprit au cas où l'on souhaite un nombre plus élevé d'alternatives, sauf peut-être en sélectionnant à partir du premier *clustering*, différents ensembles de contraintes menant à différentes matrices de distances.

4.5.3 CAMI : estimation d'un mélange de modèles alternatifs

L'approche CAMI développée par [Dang and Bailey, 2010], est une approche générative permettant d'obtenir deux *clusterings* alternatifs à partir d'un unique jeu de donnée. L'approche est fondée sur l'hypothèse d'un modèle de mélange gaussien censé avoir généré l'échantillon $\mathcal{X}$ et l'objectif est de trouver deux ensembles de paramètres $\Theta^{(1)}$ et $\Theta^{(2)}$ du mélange tels que :

- $\Theta^{(1)}$ et $\Theta^{(2)}$ sont de bons paramètres au sens où ils permettent de maximiser la log-vraisemblance des données ;
- $\Theta^{(1)}$ et $\Theta^{(2)}$ induisent des clusterings différents sous l'hypothèse *MAP*.

Objectif

L'objectif est alors de simultanément :
- maximiser la log-vraisemblance des données paramétrée par $\Theta^{(1)}$: $\mathcal{L}(\mathcal{X}, Z; \Theta^{(1)})$
- maximiser la log-vraisemblance des données paramétrée par $\Theta^{(2)}$: $\mathcal{L}(\mathcal{X}, Z; \Theta^{(2)})$
- minimiser l'information mutuelle entre $C^{(1)}$ et $C^{(2)}$ conditionnellement aux paramètres $\Theta = (\Theta^{(1)}, \Theta^{(2)})$: $MI(C^{(1)}, C^{(2)}|\Theta)$

Le critère global à optimiser s'exprime alors sous la forme :

$$Q_{\text{CAMI}} = \mathcal{L}(\mathcal{X}, Z; \Theta^{(1)}) + \mathcal{L}(\mathcal{X}, Z; \Theta^{(2)}) - \eta MI(C^{(1)}, C^{(2)})$$

où

$$\mathcal{L}(\mathcal{X}, Z; \Theta^{(r)}) = \sum_{x_i \in \mathcal{X}} \sum_{k=1}^{n_k^{(r)}} z_{ik}^{(r)} \log(\alpha_k^{(r)} f_k^{(r)}(x_i^{(r)}; \theta_k^{(r)})) \tag{4.20}$$

$$MI(C^{(1)}, C^{(2)}) = \sum_{k_1=1}^{n_k{}^{(1)}} \sum_{k_2=1}^{n_k{}^{(2)}} MI(C_{k_1}^{(1)}, C_{k_2}^{(2)}|\Theta) \tag{4.21}$$

Algorithme

L'algorithme permettant d'obtenir les meilleurs paramètres $\Theta^{(1)}$ et $\Theta^{(2)}$ suit le principe de EM (1.4.2), et alterne une étape de calcul de l'espérance de la log-vraisemblance des données complétées connaissant une estimation courante des paramètres, puis une étape de maximisation de cette espérance selon les paramètres.

Soient $z_{ik}^{(r)} = f(Z_i^{(r)} = k|x_i; \Theta^{*(r)})$ et $\tilde{z}_{kl}^{(r)} = f(C_k^{(r)}|C_l^{(\overline{r})}; \Theta^{*(r)})$. L'étape du calcul de l'espérance des variables latentes $Z_i^{(r)}$ est décomposée en un terme correspondant à la probabilité *a posteriori* issue de la part des log-vraisemblances locales :

$$z_{ik}^{(r)} = \frac{\alpha_k^{(r)} \mathcal{N}(x_i - \mu_k^{(r)}, \Sigma_k^{(r)})}{\sum_{k'=1}^{n_k} \alpha_{k'}^{(r)} \mathcal{N}(x_i - \mu_{k'}^{(r)}, \Sigma_{k'}^{(r)})} \tag{4.22}$$

avec $r \in \{1, 2\}$, et un terme correspondant à la part d'information mutuelle :

$$\tilde{z}_{kl}^{(r)} = \frac{\alpha_k^{(r)} \alpha_l^{(\overline{r})} \mathcal{N}(\mu_l^{(\overline{r})} - \mu_k^{(r)}, \Sigma_l^{(\overline{r})} + \Sigma_k^{(r)})}{\sum_{k'=1}^{n_k} \alpha_{k'}^{(r)} \alpha_l^{(\overline{r})} \mathcal{N}(\mu_l^{(\overline{r})} - \mu_{k'}^{(r)}, \Sigma_l^{(\overline{r})} + \Sigma_{k'}^{(r)})} \tag{4.23}$$

où $(r, \overline{r}) \in \{1, 2\}^2$ et $\overline{r} \neq r$.

La valeur de probabilité *a posteriori* $z_{ik}^{(r)}$ est d'autant plus forte que l'individu x_i est proche de la moyenne $\mu_k^{(r)}$ de la k-ième gaussienne relativement aux moyennes des autres gaussiennes du clustering $C^{(r)}$ et selon la matrice de variance $\Sigma^{(r)}$. De la même manière, la valeur de probabilité *a posteriori* $\tilde{z}_{kl}^{(r)}$ est d'autant plus forte que la moyenne $\mu_k^{(r)}$ de la k-ième gaussienne du clustering $C^{(r)}$ est proche de la moyenne $\mu_l^{(\overline{r})}$ de la l-ième gaussienne du clustering $C^{(\overline{r})}$ relativement aux autres gaussiennes de ce clustering.

L'étape de maximisation de l'algorithme EM consiste à maximiser en Θ l'espérance sur Z de la log vraisemblance $\mathcal{L}(\mathcal{X}, Z; \Theta^{(r)})$ (4.5.3). Les conditions d'optimalité du premier ordre donnent

les mises à jours optimales des paramètres $\alpha_k^{(r)}$ et $\mu_k^{(r)}$ $\forall r \in \{1,2\}, C_k^{(r)} \in C^{(r)}$:

$$\alpha_k^{(r)} = \frac{1}{n - \eta n_k{}^{(\overline{r})}} \left(\sum_{x_i \in \mathcal{X}} z_{ik}^{(r)} - \eta \sum_{l=1}^{n_k{}^{(\overline{r})}} \tilde{z}_{kl}^{(r)} \right) \tag{4.24}$$

$$\mu_k^{(r)} = \frac{\sum_{x_i \in \mathcal{X}} z_{ik}^{(r)} \Sigma_k^{(r)^{-1}} x_i - \eta \sum_{l=1}^{n_k{}^{(\overline{r})}} \tilde{z}_{kl}^{(r)} (\Sigma_k^{(r)} + \Sigma_l^{(\overline{r})})^{-1} \mu_l^{\overline{r}}}{\sum_{x_i \in \mathcal{X}} z_{ik}^{(r)} \Sigma_k^{(r)^{-1}} - \eta \sum_{l=1}^{n_k{}^{(\overline{r})}} \tilde{z}_{kl}^{(r)} (\Sigma_k^{(r)} + \Sigma_l^{(\overline{r})})^{-1}} \tag{4.25}$$

La mise à jour de la matrice de variances/covariances est obtenue de sorte à maximiser une borne inférieure du critère Q_{CAMI} :

$$\Sigma_k^{(r)} = \frac{\sum_{x_i \in \mathcal{X}} z_{ik}^{(r)} (x_i - \mu_k^{(r)})(x_i - \mu_k^{(r)})^\top}{\sum_{x_i \in \mathcal{X}} z_{ik}^{(r)} - \frac{\eta}{2} \sum_{l=1}^{n_{k\overline{r}}} \tilde{z}_{kl}^{(r)}} \tag{4.26}$$

Algorithme 33 CAMI

ENTRÉES : $\mathcal{X}$, $n_k^{(1)}$, $n_k^{(2)}$
SORTIES : $C^{(1)}, C^{(2)}$
1 : Initialisation aléatoire des Θ_r, $\forall r \in \{1,2\}$
2 : Étape E : Mise à jour des $z_{ik}^{(r)}$ en utilisant (4.22)
3 : Étape E : Mise à jour des $\tilde{z}_{kl}^{(r)}$ en utilisant (4.23)
4 : Étape M : Mise à jour des $\Theta_k^{(r)}$ en utilisant (4.24), (4.25) et (4.26)
5 : Si Q_{CAMI} change alors aller en **2**
6 : $C_k^{(r)} = \{x_i \in \mathcal{X} | z_{ik}^{(r)} = \max_{k' \in [1..n_k]} z_{ik'}^{(r)}\}$, $\forall k \in [1..n_k]$

Discussion

L'approche CAMI propose de résoudre la problématique de clustering alternatif sous l'hypothèse d'un modèle de mélange. Cette approche est limitée par le fait qu'elle propose de fournir uniquement un ensemble de deux *clusterings* alternatifs. Néanmoins, elle a l'avantage de permettre l'obtention de *clusterings* de nombre de groupes différents tout en reposant sur une formalisation solide et caractérisant l'ensemble des solutions comme des approximations d'estimateurs de maximum de vraisemblance pénalisée par l'objectif de dissimilarité entre les *clusterings*. Finalement, la contrepartie de la rigueur et de la solidité du formalisme est payée par le fait qu'il n'est pas possible de choisir différents algorithmes pour produire les différents *clusterings* malgré la possibilité de choisir des familles de lois du mélange différentes pour chaque *clustering*.

4.6 Contributions

4.6.1 Motivation

L'approche COBOC proposée s'inspire des méthodes d'ensemble de *clusterings* et de consensus de *clusterings* présentées précédemment : CE et SAMARAH. Pour répondre à la problématique du *clustering* multi-vues en exploitant les représentations disponibles des individus (ce qui

n'est pas réalisé dans les approches précédentes présentées dans ce chapitre), CoBoC repose sur l'approche AdaUzaBoC présentée au chapitre précédent (cf. section 3.7.3). L'idée est d'appliquer sur chaque vue le *meta*-algorithme AdaUzaBoC, ce qui permet d'utiliser n'importe quel objectif de clustering sous-jacent. La réponse au problème du clustering multi-vues, la réalisation de l'hypothèse du consensus, est réalisée au moyen de la génération incrémentale d'un ensemble de contraintes que devront respecter au mieux l'ensemble des algorithmes de *clusterings* locaux. Ce mécanisme de génération incrémentale de contraintes est tiré d'un principe ayant fait ses preuves en apprentissage semi-supervisé : le co-apprentissage.

co-apprentissage

L'algorithme de co-apprentissage [Blum and Mitchell, 1998] vise à construire deux classifieurs à partir d'un jeu de donnée $\mathcal{X}$ décrit selon deux vues, et pour lequel on dispose d'une faible quantité d'individus étiquettés. On pose alors $\mathcal{X} = \mathcal{L} \cup \mathcal{U}$ avec $|\mathcal{L}| << |\mathcal{U}|$ où $\mathcal{L}$ est l'ensemble des individus pour lesquels on dispose de l'information de classe et $\mathcal{U}$ est l'ensemble des individus non étiquettés. L'idée de l'algorithme est alors de construire à partir de $\mathcal{L}$ un classifieur dans chaque vue. Soit $\mathcal{U}' \subset \mathcal{U}$ avec $|\mathcal{U}'| = u$ fixé, chaque classifieur est utilisé pour étiquetter les exemples de $\mathcal{U}$ tout en leur associant une confiance. Les m^+ exemples positifs et m^- exemples négatifs associés à une confiance maximale sont alors sélectionnés parmi les u exemples classifiés. Ces exemples sont injectés parmi les individus étiquettés $\mathcal{L}$, et $m^+ + m^-$ individus $x_i \in \mathcal{U}$ sont retirés aléatoirement et réinjectés dans $\mathcal{U}'$.

construction incrémentale de contraintes

Le mécanisme de construction incrémentale des contraintes s'appuie directement sur ce principe de co-apprentissage. Dans notre contexte, les exemples sont les paires d'individus, pour lesquels les éléments devront être classés ensembles ou non. La terminologie des contraintes *must-link* ($\mathcal{ML}$) et *cannot-link* ($\mathcal{CL}$) peut alors être employée pour décrire les exemples positifs et négatifs respectivement. L'approche CoBoC va alors générer à chaque étape un ensemble de contraintes parmi les plus «évidentes», *i.e.* associées à une plus grande confiance, construisant ainsi l'équivalent de l'ensemble $\mathcal{L}$ du co-apprentissage. Les nouvelles contraintes sélectionnées à chaque étape sont choisies parmi $\mathcal{U} = \mathcal{X}^2 \setminus \mathcal{L}$. Un ensemble final $\mathcal{L}$ de paires $\mathcal{ML}$ ou $\mathcal{CL}$ jugé satisfaisant sert alors de guide aux algorithmes de *clusterings* locaux qui cherchent dans chaque vue une partition de $\mathcal{X}$ dans un contexte alors semi-supervisé, les contraintes constituant les exemples de $\mathcal{L}$ devant être respectées. Les différentes hypothèses de départ émisent mènent à deux variantes de ce mécanisme de co-apprentissage pour le *clustering* dans un cadre de multiplicité :

CoBoC : les partitions locales proches peuvent être obtenues selon :

- CoBoC *consensus*, la génération d'un ensemble $\mathcal{L}$ de paires d'individus, unique et commun à toutes les vues, permettant aux différents algorithmes de *clusterings* d'obtenir des résultats proches en respectant les mêmes contraintes ;
- CoBoC *complémentaire*, la génération d'une collection $\{\mathcal{L}^{(r)}\}_{r\in[1..n_r]}$ d'ensembles de paires d'individus, différents pour toutes les vues, permettant aux algorithmes de *clusterings* d'obtenir des résultats proches. Cette recherche de consensus est atteinte en s'assurant que si deux individus sont regroupés (respectivement séparés) par tous les algorithmes de *clustering* locaux sauf un, alors on doit parmettre à celui-ci de parvenir également à regrouper (respectivement séparer) ces mêmes individus.

AlterBoC : L'obtention de partitions locales différentes peut être obtenue selon :

- AlterBoC *global*, la génération d'une collection $\{\mathcal{L}\}_{r\in[1..n_r]}$ d'ensembles de paires d'individus, différents pour toutes les vues, permettant aux algorithmes de *clusterings* d'obte-

nir des résultats différents en s'assurant que chaque algorithme ne puisse respecter des contraintes que les autres algorithmes parviennent à satisfaire ;

ALTERBOC *complémentaire*, un cas particulier du mécanisme précédent en ne considérant que les contraintes que les premiers algorithmes satisfont par eux même localement. À titre d'exemple, si on a dans l'esprit : $Link^{(\overline{r})}(x_i, x_j) \wedge Link^{(r)}(x_i, x_j)\ \forall \overline{r} \neq r$, alors il est cohérent de considérer ultérieurement $(x_i, x_j) \in \mathcal{CL}^{(r)}$ afin de contraindre $\{A^{(r)}\}$ à réaliser un *clustering* différent de ceux obtenus par les $A^{(\overline{r})}$.

Dans la suite sont déclinées les deux variantes et leurs heuristiques correspondantes, en reprennant une notation plus proche de celle du chapitre 3. Les deux approches se basent sur ADAUZABOC pour faire en sorte qu'un algorithme de *clustering* quelconque satisfasse localement un ensemble de contraintes données.

Objectif

Soient $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ $\forall r \in [1..n_r]$ la recherche d'une représentation optimale facilitant le respect des contraintes par $A^{(r)}$ est caractérisée pour rappel par $X^{(r)*} = X^{(r)}P^{(r)*}$ où $P^* = \{P^{(r)*}\}_{r\in[1..n_r]}$ est la solution optimale du problème suivant :

$$\begin{array}{rrcll} \max\limits_{P} \sum\limits_{r=1}^{n_r} Q^{(r)}_{\text{COH}}(P^{(r)}) & = & \sum\limits_{r=1}^{n_r} trace({P^{(r)}}^\top {X^{(r)}}^\top X^{(r)} P^{(r)}) & & \\ s.t. & & {P^{(r)}}^\top P^{(r)} & = & Id_s \quad \forall r \in [1..n_r] \\ & & d^2_{P^{(r)}}(x_i, x_j) & \leq & \xi^{(r)}_{ij} \quad \forall (x_i, x_j) \in \mathcal{ML}^{(r)} \\ & & d^2_{P^{(r)}}(x_i, x_j) & \geq & \xi^{(r)}_{ij} \quad \forall (x_i, x_j) \in \mathcal{CL}^{(r)} \\ & & \xi^{(r)}_{ij} & \geq & 0 \quad \forall r \in [1..n_r],\ \forall (x_i, x_j) \in \mathcal{ML}^{(r)} \cup \mathcal{CL}^{(r)} \end{array} \tag{4.27}$$

La différenciation entre les deux approches COBOC et ALTERBOC se fait *via* la génération des contraintes. COBOC et ALTERBOC sont des instanciations de la plateforme () utilisant différentes heuristiques qui sont autant de propositions pour le développement d'approches génériques de types multi-vues ou alternatives. Dans ce cadre, les contributions présentées par la suite sont essentiellement algorithmiques et prennent la forme de stratégies dont on espère *a priori* qu'elles amélioreront la qualité des *clusterings* produits.

L'idée est de partir des ensembles $\mathcal{ML}^{(r)} = \mathcal{CL}^{(r)} = \emptyset$ et d'alterner deux étapes qui sont :
- la recherche des *clusterings* optimaux locaux selon ADAUZABOC, pour $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ fixés ;
- l'augmentation de $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ selon les *clusterings* locaux obtenus.

La recherche d'un *clustering* local optimal étant indépendante de la recherche des *clusterings* dans les autres vues, cette étape est réalisée indépendamment dans chaque vue et correspond exactement à l'algorithme ADAUZABOC détaillé en section 3.7.3.

La seconde étape consiste à augmenter $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$. Cette augmentation est réalisée en sélectionnant à partir de l'ensemble des paires d'individus non présentes dans les contraintes, un ensemble de m paires candidates pour chaque *clustering* local. Les paires candidates sont associées à une confiance indiquant leur prédisposition à être une contrainte $\mathcal{ML}$ ou $\mathcal{CL}$.

Soit $H^{+(r)}$ et $H^{-(r)}$ les matrices des hypothèses de *clustering* dans la vue r définies par :

$$H^{+(r)}_{ij} = \begin{cases} 1 & \text{si } i \neq j \text{ et } Link^{(r)}(x_i, x_j) \\ 0 & \text{sinon} \end{cases}$$

$$H^{-(r)}_{ij} = \begin{cases} -1 & \text{si } i \neq j \text{ et } \overline{Link}^{(r)}(x_i, x_j) \\ 0 & \text{sinon} \end{cases}$$

La matrice complète des hypothèses de *clustering* est alors donnée par :

$$H^{(r)} = H^{+(r)} + Id_n + H^{-(r)} \tag{4.28}$$

En particulier les paires d'individus $(x_i, x_i) \in \mathcal{X}$ sont toujours classées ensembles par $A^{(r)}$, ainsi $H^{(r)}_{ii} = 1$. Les valeurs positives (respectivement négatives) de la matrice des hypothèses de *clustering* $H^{(r)}$ sont alors les paires d'individus correspondant aux exemples étiquettés positivement (respectivement négativement), dans la terminologie du co-apprentissage.

Soit $D^{(r)}$ la matrice des distances entre individus dans le dernier sous-espace optimal de la vue r définie par :

$$D^{(r)}_{ij} = d^2_{P^{(r)*}}(x_i, x_j) \tag{4.29}$$

On pose $D^{(r)}_{ij_{max}}$ et $D^{(r)}_{ij_{min}}$ tels que :

$$D^{(r)}_{ij_{max}} = \begin{cases} \underset{(x_k,x_l)\in\mathcal{X}^2}{max} \left(H^{+(r)}_{kl} D^{(r)}_{kl} \right) & \text{si } i \neq j \text{ et } Link^{(r)}(x_i, x_j) \\ \underset{(x_k,x_l)\in\mathcal{X}^2}{max} \left(H^{-(r)}_{kl} D^{(r)}_{kl} \right) & \text{si } i \neq j \text{ et } \overline{Link}^{(r)}(x_i, x_j) \end{cases}$$

$$D^{(r)}_{ij_{min}} = \begin{cases} \underset{(x_k,x_l)\in\mathcal{X}^2}{min} \left(H^{+(r)}_{kl} D^{(r)}_{kl} \right) & \text{si } i \neq j \text{ et } Link^{(r)}(x_i, x_j) \\ \underset{(x_k,x_l)\in\mathcal{X}^2}{min} \left(H^{-(r)}_{kl} D^{(r)}_{kl} \right) & \text{si } i \neq j \text{ et } \overline{Link}^{(r)}(x_i, x_j) \end{cases}$$

On pose $\alpha^{(r)}_{ij}$ la confiance associée à la paire (x_i, x_j) dans la vue r qui s'exprime par :

$$\alpha^{(r)}_{ij} = \frac{H^{(r)}_{ij}(D^{(r)}_{ij_{max}} - D^{(r)}_{ij})}{D^{(r)}_{ij_{max}} - D^{(r)}_{ij_{min}}} \tag{4.30}$$

Cette confiance est à la base des différentes déclinaisons de CoBoC et AlterBoC. Les hypothèses considérées pour les approches proposées sont alors les suivantes :

- plus une confiance $\alpha^{(r)}_{ij} > 0$ est élevée, plus on a la certitude d'avoir $Link^{(r)}(x_i, x_j)$;
- plus une confiance $\alpha^{(r)}_{ij} < 0$ est faible, plus on a la certitude d'avoir $\overline{Link}^{(r)}(x_i, x_j)$.

En raisonnant en terme de distance, et non en terme de confiance, ces hypothèses reflètent les résultats obtenus à l'issue des travaux sur AdaUzaBoC au chapitre précédent (cf. section 3.7.3).

4.6.2 CoBoC : *boosting* collectif et collaboratif pour la recherche de consensus

L'approche générique de recherche de consensus entre plusieurs vues d'un même jeu de données, ou entre plusieurs algorithmes de *clusterings* appliqués à un jeu de donnée mono-vue se décline en deux heuristiques :

CoBoC consensus, pour laquelle chaque vue participe à la construction d'un même ensemble de contraintes que tous les algorithmes de *clustering* devront satisfaire au mieux ;

CoBoC complémentaire, pour laquelle chaque vue $\overline{r} \neq r$ participe à la construction d'un même ensemble de contraintes pour r que l'algorithme $A^{(r)}$ ne parvient pas par lui même à satisfaire *a priori*.

CoBoC consensus

On se place dans le cadre où chaque vue participe à la construction du même ensemble de contraintes. Ainsi pour simplifier on notera $\mathcal{ML} = \mathcal{ML}^{(r)}$ et $\mathcal{CL} = \mathcal{CL}^{(r)}$ $\forall r \in [1..n_r]$. L'idée est de partir des ensembles $\mathcal{ML} = \mathcal{CL} = \emptyset$ et d'alterner deux étapes qui sont :
- la recherche des *clusterings* optimaux locaux selon AdaUzaBoC, pour $\mathcal{ML}$ et $\mathcal{CL}$ fixés ;
- l'augmentation de $\mathcal{ML}$ et $\mathcal{CL}$ selon les *clusterings* locaux obtenus et la stratégie de recherche de consensus, notée Γ, choisie.

La première étape est le cœur du chapitre précédent et ne sera pas détaillée davantage, elle consiste simplement à résoudre le problème (4.27).

Concernant la seconde étape, partant du calcul de la confiance $\alpha_{ij}^{(r)}$ (4.30), on calcul une confiance globale pour chaque paire d'individus comme une moyenne des confiances locales :

$$\alpha_{ij} = \frac{1}{n_r} \sum_{r=1}^{n_r} \alpha_{ij}^{(r)} \tag{4.31}$$

Une valeur positive et élevée de α_{ij} indique que x_i et x_j ont majoritairement été classés ensemble par les algorithmes $A^{(r)}$ et que ceux-ci sont dans chaque vue plus proches entre eux que des autres individus. Dans ce cas on est davantage certain que x_i et x_j devraient appartenir à un même groupe. Cette confiance permet de définir $\zeta_{\mathcal{ML}}$ et $\zeta_{\mathcal{CL}}$ comme l'ensemble des paires d'individus candidates :

$$\zeta_{\mathcal{ML}} = \{(x_i, x_j) \in \mathcal{X}^2 \setminus (\mathcal{ML} \cup \mathcal{CL}) \mid \alpha_{ij} > 0\} \tag{4.32}$$
$$\zeta_{\mathcal{CL}} = \{(x_i, x_j) \in \mathcal{X}^2 \setminus (\mathcal{ML} \cup \mathcal{CL}) \mid \alpha_{ij} < 0\} \tag{4.33}$$

Ces ensembles sont munis de la relation d'ordre $\prec$ définie par :

$$(x_i, x_j) \prec (x_{i'}, x_{j'}) \Leftrightarrow |\alpha_{ij}| > |\alpha_{i'j'}|$$

qui permet de former une liste ordonnée par la confiance des éléments de $\zeta_{\mathcal{ML}}$ et $\zeta_{\mathcal{CL}}$.

La génération des nouvelles contraintes $\Gamma(\zeta)$ peut alors être obtenue selon trois opérateurs (ou stratégies) que sont :
- la sélection aléatoire Γ_{Random} qui consiste à tirer m^+ et m^- paires d'individus aléatoirement parmi $\zeta_{\mathcal{ML}}$ et $\zeta_{\mathcal{CL}}$ respectivement ;
- la sélection confiante Γ_{Max} qui consiste à sélectionner les m^+ et m^- premières paires d'individus des listes ordonnées associées à $\zeta_{\mathcal{ML}}$ et $\zeta_{\mathcal{CL}}$ respectivement ;
- le sélection incertaine Γ_{Min} qui consiste à sélectionner les m^+ et m^- dernières paires d'individus des listes ordonnées associées à $\zeta_{\mathcal{ML}}$ et $\zeta_{\mathcal{CL}}$ respectivement.

Les règles d'augmentations sont alors définies par :

$$\mathcal{ML} = \mathcal{ML} \cup \Gamma(\zeta_{\mathcal{ML}}) \tag{4.34}$$
$$\mathcal{CL} = \mathcal{CL} \cup \Gamma(\zeta_{\mathcal{CL}}) \tag{4.35}$$

CoBoC complémentaire

L'idée est de partir des ensembles $\mathcal{ML}^{(r)} = \mathcal{CL}^{(r)} = \emptyset$ et d'alterner deux étapes qui sont :
- la recherche des *clusterings* optimaux locaux selon AdaUzaBoC, pour $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ fixés ;

Algorithme 34 CoBoC consensus

Entrées : $\mathcal{X}$, $\{X^{(r)}\}_{r\in[1..n_r]}$, n_k, $\{A^{(r)}\}_{r\in[1..n_r]}$, Γ, m^+, m^-, t_f
Sorties : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialiser $\mathcal{CL} = \mathcal{ML} = \emptyset$
2 : Initialiser $t = 0$
3 : Appliquer AdaUzaBoC sur $X^{(r)}$ avec $A^{(r)}$, $\mathcal{CL}$ et $\mathcal{ML}$
4 : Déterminer $H^{(r)}$ selon (4.28), $\forall r \in [1..n_r]$
5 : Calculer α_{ij} selon (4.31), $\forall (x_i, x_j) \in \mathcal{X}^2$
6 : Augmenter $\mathcal{ML}$ et $\mathcal{CL}$ par (4.34) et (4.35)
7 : Si $t < t_f$ alors $t = t + 1$ et aller en **3**
8 : $C = Vote(\{H^{(r)}, X^{*(r)}\}_{r\in[1..n_r]})$

– l'augmentation de $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ selon les *clusterings* locaux obtenus et la stratégie de recherche de consensus Γ choisie.

Partant du calcul de la confiance $\alpha_{ij}^{(r)}$ (4.30), on calcul une confiance $\tilde{\alpha}_{ij}^{(r)}$ comme moyenne sur les vues $\overline{r}$ des confiances locales associées aux paires d'individus :

$$\tilde{\alpha}_{ij}^{(r)} = \frac{1}{n_r - 1} \sum_{\substack{\overline{r}=1 \\ \overline{r}\neq r}}^{n_r} \alpha_{ij}^{(\overline{r})} \tag{4.36}$$

Une valeur positive et élevée de $\tilde{\alpha}_{ij}^{(r)}$ indique que x_i et x_j sont majoritairement classés ensemble par les algorithmes $A^{(\overline{r})}$ $\forall \overline{r} \in [1..n_r] \wedge \overline{r} \neq r$ et que ces individus sont pour chaque vue autre que r, plus proches entre eux qu'aux autres individus. Dans ce cas on est davantage convaincu que x_i et x_j devraient appartenir à un même groupe dans les autres vues. Dans ce contexte complémentaire, l'idée est que si deux individus appartiennent à un même groupe dans les vues $\overline{r} \neq r$, et si ces individus sont séparés par $A^{(r)}$, alors il faut suggérer à $A^{(r)}$ de les regrouper.

La confiance $\tilde{\alpha}$ permet alors de définir pour chaque vue r, les ensembles de paires d'individus candidates $\zeta_{\mathcal{ML}}^{(r)}$ et $\zeta_{\mathcal{CL}}^{(r)}$:

$$\zeta_{\mathcal{ML}}^{(r)} = \{(x_i, x_j) \in \mathcal{X}^2 \setminus (\mathcal{ML}^{(r)} \cup \mathcal{CL}^{(r)}) \mid \tilde{\alpha}_{ij}^{(r)} > 0 \wedge \overline{Link}^{(r)}(x_i, x_j)\} \tag{4.37}$$

$$\zeta_{\mathcal{CL}}^{(r)} = \{(x_i, x_j) \in \mathcal{X}^2 \setminus (\mathcal{ML}^{(r)} \cup \mathcal{CL}^{(r)}) \mid \tilde{\alpha}_{ij}^{(r)} < 0 \wedge Link^{(r)}(x_i, x_j)\} \tag{4.38}$$

Ces ensembles sont munis de la relation d'ordre $\prec^{(r)}$ définie par :

$$(x_i, x_j) \prec^{(r)} (x_{i'}, x_{j'}) \Leftrightarrow |\tilde{\alpha}_{ij}^{(r)}| > |\tilde{\alpha}_{i'j'}^{(r)}|$$

qui permet de former une liste ordonnée par la confiance des éléments de $\zeta_{\mathcal{ML}}^{(r)}$ et $\zeta_{\mathcal{CL}}^{(r)}$.

La génération des nouvelles contraintes $\Gamma(\zeta)$ peut alors être obtenue selon les trois opérateurs Γ_{Random}, Γ_{Max} et Γ_{Min} définis comme précédemment.
Les règles d'augmentations sont alors définies par :

$$\mathcal{ML}^{(r)} = \mathcal{ML}^{(r)} \cup \Gamma(\zeta_{\mathcal{ML}}^{(r)}) \tag{4.39}$$

$$\mathcal{CL}^{(r)} = \mathcal{CL}^{(r)} \cup \Gamma(\zeta_{\mathcal{CL}}^{(r)}) \tag{4.40}$$

Algorithme 35 CoBoC complémentaire

Entrées : $\mathcal{X}$, $\{X^{(r)}\}_{r\in[1..n_r]}$, $n_k^{(r)}$, $\{A^{(r)}\}_{r\in[1..n_r]}$, Γ, m^+, m^-, t_f
Sorties : $C = \{C_1, ..., C_{n_k}\}$
1 : Initialiser $\mathcal{CL}^{(r)} = \mathcal{ML}^{(r)} = \emptyset$, $\forall r \in [1..n_r]$
2 : Initialiser $t = 0$
3 : Appliquer AdaUzaBoC sur $X^{(r)}$ avec $A^{(r)}$, $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$
4 : Déterminer $H^{(r)}$ selon (4.28), $\forall r \in [1..n_r]$
5 : Calculer $\tilde{\alpha}_{ij}$ selon (4.36), $\forall (x_i, x_j) \in \mathcal{X}^2$
6 : Augmenter $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ par (4.39) et (4.40)
7 : Si $t < t_f$ alors $t = t + 1$ et aller en **3**
8 : $C = Vote(\{H^{(r)}, X^{*(r)}\}_{r\in[1..n_r]})$

Construction de la partition unique.

Dans l'esprit des méthodes de *clustering* multi-vues auxquelles se confronte CoBoC, un unique *clustering* des individus est attendu. Dans ce contexte, une fusion finale est réalisée sous la forme d'un vote à la majorité entre les différents *clusterings* de chaque vue. Ces *clusterings* sont alors considérés comme des hypothèses qui peuvent être combinées de différentes façons. À partir de l'ensemble $\{H^{(r)}\}_{r\in[1..n_r]}$ des hypothèses de *clustering* sur les paires d'individus et l'ensemble $\{X^{*(r)}_{r\in[1..n_r]}\}$ des représentations optimales locales de $\mathcal{X}$ obtenues par AdaUzaBoC, un *clustering* C final peut être obtenu par :

1. La construction d'une matrice de similarité K_1 à partir des hypothèses de *clustering* :

$$K_1 = \sum_{r=1}^{n_r} \tilde{H}^{(r)} \tag{4.41}$$

   où $\tilde{H}^{(r)} = \frac{1}{2}(H^{(r)} + 1)$, ainsi $H_{ij}^{(r)} \in \{0, 1\}$. K_1 est ensuite utilisé comme matrice de similarité, dans un algorithme de *clustering* classique mono-vue adapté (*e.g.* Agnes, KKM, KFKM, SC, *etc.*).

2. Selon le même principe de vote, mais en utilisant davantage les représentations finales optimales des individus en recalculant les confiances α_{ij} pour chaque paire (x_i, x_j). Soit α_{min} quantité négative correspondante à la plus faible des confiances sur les paires d'individus :

$$\alpha_{min} = \underset{(x_i,x_j)\in\mathcal{X}^2}{min} \alpha_{ij}$$

   et α_{max} tel que :

$$\alpha_{max} = \underset{(x_i,x_j)\in\mathcal{X}^2}{max} (\alpha_{ij} - \alpha_{min})$$

   un noyau normalisé peut alors être construit à partir de α par :

$$K_{2ij} = \frac{\alpha_{ij} - \alpha_{min}}{\alpha_{max}} \tag{4.42}$$

$\alpha_{min} < 0$ étant la plus petite valeur de confiance, le numérateur permet de translater les confiances vers des valeurs positives. Le dénominateur permet alors de ramener la valeur maximale de confiance translatée à 1. K_{2ij} peut donc être vue comme une mesure de similarité normalisée entre 0 et 1.

Discussion

Les deux approches CoBoC consensus et CoBoC complémentaire proposées reposent sur la même procédure pour atteindre dans chaque vue r une représentation optimale et un *clustering* optimal dans cette représentation, respectant au mieux les ensembles de contraintes données $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$. Chaque vue, en respectant ses contraintes, doit aller vers une solution de *clustering* proche de celle des autres vues, par construction. Le comportement de ces approches heuristiques sera présenté plus en détail dans la section d'évaluation. Le même genre de mécanisme peut être proposé pour la recherche de plusieurs partitions alternatives d'un ensemble d'individus $\mathcal{X}$, ce qui est l'objet de la prochaine section.

4.6.3 AlterBoC : *boosting* collectif et collaboratif pour la recherche d'alternatives

AlterBoC est une approche heuristique de découverte de *clusterings* alternatifs dont le mécanisme est calqué sur celui de CoBoC. Les différentes heuristiques proposées pour l'obtention d'alternatives sont inspirées des travaux de [Davidson and Qi, 2008] pour ADFT (cf. section 4.5.2). Rappellons qu'AdaUzaBoC peut fournir pour chaque alternative, le *clustering* $C^{(r)}$ fondé sur la fonction de distance $P^{(r)}{P^{(r)}}^\top$ apprise. Cette distance étant apprise par l'intermédiaire des contraintes, il est alors envisageable de contrôler, par la construction de contraintes appropriées, la recherche de sous-espaces différents, induisant, par hypothèse, des *clusterings* différents. Le simple fait que les *clusterings* obtenus localement soient optimaux, relativement aux distances apprises, suggère un mécanisme d'obtention de *clusterings* alternatifs, au sens de la problématique de l'*alternative clustering*.

L'approche générique de recherche de *clusterings* alternatifs d'un même jeu de données se décline également en deux heuristiques :

AlterBoC global, pour laquelle chaque vue ou alternative $\overline{r} \neq r$ participe à la construction d'un même ensemble de contraintes pour r quels que soient les résultats de $A^{(r)}$ *a priori* sur ces contraintes ;

AlterBoC complémentaire, pour laquelle chaque alternative $\overline{r} \neq r$ participe à la construction d'un même ensemble de contraintes pour r que l'algorithme $A^{(r)}$ ne parvient pas à satisfaire *a priori*.

AlterBoC global

À partir d'une représentation matricielle X de l'ensemble d'individu $\mathcal{X}$, l'idée est de construire des ensembles $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ permettant à un algorithme $A^{(r)}$ d'obtenir un des n_r *clusterings* alternatifs. Soient $\mathcal{ML}^{(r)} = \mathcal{CL}^{(r)} = \emptyset$, l'approche consiste à alterner deux étapes qui sont :

- la recherche des *clusterings* optimaux locaux selon AdaUzaBoC, pour $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ fixés ;
- l'augmentation de $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ selon les *clusterings* locaux obtenus et la stratégie de recherche d'alternatives Γ choisie.

La recherche d'un *clustering* local optimal est toujours réalisée grâce à l'algorithme AdaUzaBoC détaillé en section 3.7.3.

Soit la confiance $\tilde{\alpha}_{ij}^{(r)}$ (4.36). Une valeur positive et élevée de $\tilde{\alpha}_{ij}^{(r)}$ indique que x_i et x_j ont majoritairement été classés ensemble par les algorithmes $A^{(\overline{r})}$ ($\overline{r} \neq r$) et que ceux ci sont pour chaque vue autre que r, plus proches entre eux qu'aux autres individus. L'idée dans le cadre de la recherche d'un *clustering* $C^{(r)}$ alternatif à $\{C^{(\overline{r})}\}$ est de s'assurer que $A^{(r)}$ ne regroupe pas x_i et x_j. Ainsi, (x_i, x_j) doit correspondre à une contrainte $\mathcal{CL}$.

La confiance (4.36) permet de définir pour chaque vue r, les ensembles de paires d'individus candidates $\zeta^{(r)}_{\mathcal{ML}}$ et $\zeta^{(r)}_{\mathcal{CL}}$:

$$\zeta^{(r)}_{\mathcal{ML}} = \{(x_i, x_j) \in \mathcal{X}^2 \setminus (\mathcal{ML}^{(r)} \cup \mathcal{CL}^{(r)}) \mid \tilde{\alpha}^{(r)}_{ij} < 0\} \quad (4.43)$$

$$\zeta^{(r)}_{\mathcal{CL}} = \{(x_i, x_j) \in \mathcal{X}^2 \setminus (\mathcal{ML}^{(r)} \cup \mathcal{CL}^{(r)}) \mid \tilde{\alpha}^{(r)}_{ij} > 0\} \quad (4.44)$$

Ces ensembles sont munis de la relation d'ordre $\prec^{(r)}$ défini par :

$$(x_i, x_j) \prec^{(r)} (x_{i'}, x_{j'}) \Leftrightarrow |\tilde{\alpha}^{(r)}_{ij}| > |\tilde{\alpha}^{(r)}_{i'j'}|$$

qui permet de former une liste ordonnée par la confiance des éléments de $\zeta^{(r)}_{\mathcal{ML}}$ et $\zeta^{(r)}_{\mathcal{CL}}$.

La génération des nouvelles contraintes $\Gamma(\zeta)$ peut alors être obtenue selon trois opérateurs (ou stratégies) que sont :

- la sélection aléatoire Γ_{Random} qui consiste à tirer m^+ et m^- paires d'individus aléatoirement parmi $\zeta^{(r)}_{\mathcal{ML}}$ et $\zeta^{(r)}_{\mathcal{CL}}$ respectivement ;
- la sélection confiante Γ_{Max} qui consiste à sélectionner les m^+ et m^- premières paires d'individus des listes ordonnées associées à $\zeta^{(r)}_{\mathcal{ML}}$ et $\zeta^{(r)}_{\mathcal{CL}}$ respectivement ;
- le sélection incertaine Γ_{Min} qui consiste à sélectionner les m^+ et m^- dernières paires d'individus des listes ordonnées associées à $\zeta^{(r)}_{\mathcal{ML}}$ et $\zeta^{(r)}_{\mathcal{CL}}$ respectivement.

et les règles d'augmentations sont définies par :

$$\mathcal{ML}^{(r)} = \mathcal{ML}^{(r)} \cup \Gamma(\zeta^{(r)}_{\mathcal{ML}}) \quad (4.45)$$

$$\mathcal{CL}^{(r)} = \mathcal{CL}^{(r)} \cup \Gamma(\zeta^{(r)}_{\mathcal{CL}}) \quad (4.46)$$

Algorithme 36 ALTERBOC global

ENTRÉES : $\mathcal{X}$, $\{A^{(r)}\}_{r\in[1..n_r]}$, Γ, m^+, m^-, t_f
SORTIES : $\Pi = \{C^{(1)}, ..., C^{(n_r)}\}$
1 : Initialiser $\mathcal{CL}^{(r)} = \mathcal{ML}^{(r)} = \emptyset$, $\forall r \in [1..n_r]$
2 : Initialiser $t = 0$
3 : Appliquer ADAUZABOC sur $\mathcal{X}$ avec $A^{(r)}$, $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ $\forall r \in [1..n_r]$
4 : Déterminer $H^{(r)}$ selon (4.28), $\forall r \in [1..n_r]$
5 : Calculer $\tilde{\alpha}_{ij}$ selon (4.36), $\forall (x_i, x_j) \in \mathcal{X}^2$
6 : Augmenter $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ par (4.45) et (4.46)
7 : Si $t < t_f$ alors $t = t + 1$ et aller en **3**
8 : $C^{(r)}$ = Application de ADAUZABOC sur $\mathcal{X}$ avec $A^{(r)}$, $\forall r \in [1..n_r]$

ALTERBOC complémentaire

L'heuristique complémentaire est essentiellement la même que la précédente, si ce n'est dans la construction explicite des ensembles de paires d'individus candidates $\zeta^{(r)}_{\mathcal{ML}}$ et $\zeta^{(r)}_{\mathcal{CL}}$:

$$\zeta^{(r)}_{\mathcal{ML}} = \{(x_i, x_j) \in \mathcal{X}^2 \setminus (\mathcal{ML}^{(r)} \cup \mathcal{CL}^{(r)}) \mid \tilde{\alpha}^{(r)}_{ij} < 0 \wedge \overline{Link}^{(r)}(x_i, x_j)\}$$

$$\zeta^{(r)}_{\mathcal{CL}} = \{(x_i, x_j) \in \mathcal{X}^2 \setminus (\mathcal{ML}^{(r)} \cup \mathcal{CL}^{(r)}) \mid \tilde{\alpha}^{(r)}_{ij} > 0 \wedge Link^{(r)}(x_i, x_j)\}$$

munis de la même relation d'ordre $\prec^{(r)}$ permettant de former les listes ordonnées par la confiance des éléments de $\zeta^{(r)}_{\mathcal{ML}}$ et $\zeta^{(r)}_{\mathcal{CL}}$. Intuitivement, un bon exemple de paire candidate pour être une

contrainte $(x_i, x_j) \in \mathcal{ML}^{(r)}$ est un couple séparé dans les alternatives $\overline{r}$ et aussi dans r. Ainsi une façon de forcer les algorithmes de *clustering* à se comporter différemment est d'insister pour que $A^{(r)}$ regroupe x_i et x_j.

La génération des nouvelles contraintes est également réalisée au choix par Γ_{Random}, Γ_{Max} ou Γ_{Min}. Les règles d'augmentation sont également inchangées :

$$\mathcal{ML}^{(r)} = \mathcal{ML}^{(r)} \cup \Gamma(\zeta^{(r)}_{\mathcal{ML}}) \quad (4.47)$$

$$\mathcal{CL}^{(r)} = \mathcal{CL}^{(r)} \cup \Gamma(\zeta^{(r)}_{\mathcal{CL}}) \quad (4.48)$$

Algorithme 37 ALTERBOC complémentaire

ENTRÉES : $\mathcal{X}$, $\{A^{(r)}\}_{r\in[1..n_r]}$, Γ, m^+, m^-, t_f
SORTIES : $\Pi = \{C^{(1)}, ..., C^{(n_r)}\}$
1 : Initialiser $\mathcal{CL}^{(r)} = \mathcal{ML}^{(r)} = \emptyset$, $\forall r \in [1..n_r]$
2 : Initialiser $t = 0$
3 : Appliquer ADAUZABOC sur $\mathcal{X}$ avec $A^{(r)}$, $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ $\forall r \in [1..n_r]$
4 : Déterminer $H^{(r)}$ selon (4.28), $\forall r \in [1..n_r]$
5 : Calculer $\tilde{\alpha}_{ij}$ selon (4.36), $\forall (x_i, x_j) \in \mathcal{X}^2$
6 : Augmenter $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$ par (4.47) et (4.48)
7 : Si $t < t_f$ alors $t = t + 1$ et aller en **3**
8 : $C^{(r)}$ = Application de ADAUZABOC sur $\mathcal{X}$ avec $A^{(r)}$, $\forall r \in [1..n_r]$

Discussion

Les approches ALTERBOC global et ALTERBOC complémentaire proposent d'atteindre un ensemble de représentations optimales associées chacunes à un *clustering* optimal, respectant au mieux les ensembles de contraintes données $\mathcal{ML}^{(r)}$ et $\mathcal{CL}^{(r)}$. Les contraintes sont construites de sorte à rechercher une divergence entre les alternatives. Les sous-espaces de représentation obtenus doivent alors être distincts et les *clusterings* associés doivent être des optima différents.

Tout comme les approches de *clustering* alternatifs présentées précédemment, l'intuition de l'efficacité de la recherche d'alternatives se compromet, à nombre d'alternatives augmentant. En effet, il est plus facile d'envisager l'obtention de partitions différentes dans le cas de deux alternatives que pour un nombre plus élevé. Par exemple, dès trois alternatives, soient $C^{(1)}$, $C^{(2)}$ et $C^{(3)}$ trois partitions d'un même jeu de données obtenues par $A^{(1)}$, $A^{(2)}$ et $A^{(3)}$. On ne peut dans ce contexte garantir une réelle différence entre les alternatives car la décision finale associée à chaque paire d'individus est binaire (regroupée ou séparée). Parmi les trois décideurs $A^{(1)}$, $A^{(2)}$, $A^{(3)}$, si deux d'entre eux permettent d'obtenir des partitions différentes, alors le troisième aura nécessairement une partie commune avec au moins l'un d'entre eux, voire même les deux. La tâche est alors de contrôler dans quelle mesure le troisième algorithme aura des parties communes, mais réduites, avec les deux autres.

4.7 Évaluation

Les approches COBOC et ALTERBOC ont été testées expérimentalement en suivant différentes procédures d'évaluation internes et externes. Les jeux de données qui ont servi de base de test sont tirés des chapitres précédents.

L'approche COBOC a été testée dans deux contextes applicatifs différents :

– le contexte multi-vues (cf. chapitre 2) où l'on cherche une partition consensus de l'ensemble $\mathcal{X}$ où chaque individu est décrit simultanément par plusieurs représentations. Ce cadre applicatif est celui des approches de *clustering* multi-vues ;
– le contexte de la *combinaison de modèles*, où l'on applique plusieurs algorithmes de *clustering* différents sur un jeu de donnée mono-vue. Ce cadre applicatif est typique des développement des approches de *clustering* d'ensemble, de *clustering* collaboratif ou de *clustering* alternatif.

L'application au contexte multi-vues est observée sur le jeu de données *mfeat* (cf. section 2.5.1), et l'application au contexte de la combinaison de modèles pour la recherche de consensus ou d'alternatives est observée sur les jeux de données *Iris*, *parkinson* et *Wine* (cf. section 3.8.1).

4.7.1 Protocole expérimental

Dans un premier temps, la recherche d'une solution consensus par CoBoC et de solutions alternatives par AlterBoC sont caractérisées en termes d'évaluation interne, en observant l'évolution de la moyenne des informations mutuelles entre les différents *clusterings* locaux (avant l'étape de vote final pour CoBoC) :

$$AvgNMI(\Pi) = \frac{1}{n_r}\sum_{r=1}^{n_r} NMI(C, C^{(r)}) \; ; \; \Pi = \{C^{(1)}, \ldots, C^{(n_r)}\}$$

Dans un second temps, la performance des différentes approches est mesurée par une évaluation externe (% *F-mesure*, *AvgEnt* et *NMI*). Cette évaluation est réalisée selon plusieurs objectifs :
– observer l'apport des approches collaboratives sur chaque algorithme de *clustering* $A^{(r)}$ (avant la fusion finale pour CoBoC) selon la stratégie de collaboration Γ employée et au regard des résultats obtenus par chacun de ces algorithmes sans procédure de collaboration ;
– observer l'apport des solutions obtenues par CoBoC et de la fusion finale par calcul de K_1 et K_2, et comparée à une solution de *clustering* multi-vues : CoFKM ;
– observer l'apport des solutions locales proches obtenues par (CoBoC) ou alternatives obtenues par (AlterBoC) comme prémisse à l'application de CoFKM ou CoKFKM. Cette observation a pour but d'observer l'apport de la diversité parmi les différents *clusterings* sur les résultats des approches multi-vues : CoFKM et CoKFKM.

Les résultats obtenus correspondent à une moyenne de 20 exécutions pour *Iris*, 10 exécutions pour *wine* et *parkinson* et 5 exécutions pour *mfeat*. L'augmentation du nombre de contraintes est paramétré de la façon suivante :
– le nombre maximum d'augmentations de contraintes est fixé à 10 ;
– à chaque itération de CoBoC ou AlterBoC, *i.e.* à chaque augmentation du nombre de contraintes, $m = p_{\%} \times (\frac{n(n-1)}{2n_k})$ contraintes sont générées, où $p_{\%}$ est un pourcentage prédéfini. le terme m correspond à un pourcentage de nombre de contraintes $\mathcal{ML}$ pouvant être générées, sous hypothèse de groupes de tailles homogènes. Dans les expériences, $p_{\%} = 1$, ainsi le nombre total de contraintes générées est de $\frac{1}{10n_k}\times$ le nombre de paires d'individus différentes.

Lorsqu'ils sont utilisés, les algorithmes de *clustering* sont paramétrés de manière classique. Si le nombre de groupes est nécessaire, celui-ci correspond au nombre de classes du jeu de données correspondant. Les paramètres de flou éventuels nécessaires sont tous fixés à $\beta = 1.25$.

Les approches AdaUzaBoC encapsulant les algorithmes précédents sont paramétrées par le choix heuristique de la dimensionnalité du sous-espace à calculer à chaque étape : s, correspondant au nombre de valeurs propres positives de la matrice à diagonaliser. L'initialisation

de l'algorithme boîte noire employé est invariante pour une recherche de sous-espace optimal donnée, mais différente entre les vues ou alternatives.

4.7.2 Évaluation interne

FIGURE 4.2 — Légende pour l'évaluation interne de COBOC et ALTERBOC.

Évaluation interne de COBOC

L'évaluation interne de COBOC consiste essentiellement à observer l'impact de la recherche heuristique de collaboration entre les algorithmes locaux dans les contextes de la combinaison de modèles et du *clustering* multi-vues. L'objectif, malgré un faible contrôle sur le comportement des différents algorithmes, est d'obtenir une augmentation de la valeur d'information mutuelle normalisée moyenne entre les résultats de ces algorithmes.

Évaluation interne de COBOC dans le cadre de la combinaison de modèle. Les heuristiques consensus et complémentaire ont été observées sur une exécution dans le cadre de la combinaison de modèles non supervisés (cf. figure 4.3 et 4.4).

Selon l'heuristique consensus (figure 4.3), la stratégie maximum (Γ_{Max}) n'apporte pas de résultats significatifs, dans la mesure où les paires d'individus de $\zeta_{\mathcal{ML}}$ (respectivement $\zeta_{\mathcal{CL}}$) sélectionnées comme étant les plus confiantes sont déjà regroupées (respectivement séparées) par tous les algorithmes locaux. Néanmoins il existe certaines paires d'individus pour lesquelles ces observations ne sont pas vraies. Il en résulte une modification mineure de la mesure de similarité (NMI) entre les résultats des algorithmes locaux qui n'est favorable que dans les cas présentés de recherche de consensus entre quatre algorithmes pour *wine*, et six algorithmes pour *Iris*. Néanmoins, cette observation est limitée à une exécution, pour une configuration particulière de l'algorithme COBOC, et un choix particulier des algorithmes locaux. Le résultat positif qui en ressort est qu'il est possible d'atteindre une solution offrant un meilleur consensus entre les algorithmes locaux. La stratégie minimum (Γ_{Min}) n'est efficace dans la recherche de consensus que pour le jeu de donnée *Parkinson*, pour lequel les algorithmes locaux utilisés se comporte vraiment différemment, et les résultats de base obtenus sont très dissimilaires. Elle est donc globalement peu concluente dans ce contexte. La stratégie *random* a un comportement plus instable. En général, la tendance est plutôt négative, à nombre d'échange de contraintes augmentant. Néanmoins, on observe la possibilité d'atteindre un meilleur consensus que la stratégie maximum, ce qui est un résultat très positif. Cependant l'identification de tels cas particuliers n'a pas été l'objet de cette étude.

Les observations issues des expériences sur l'heuristique complémentaire (figure 4.4) corroborent les observations précédentes au sujet de l'inefficacité de la stratégie minimum (malgré une observation positive à faible nombre d'échanges de contraintes pour *wine* avec six algorithmes locaux) et l'atteignabilité de très bonnes solutions de consensus par la stratégie *random*. En revanche, dans ce contexte, la stratégie maximum est plus instable, et tend davantage à

s'éloigner des solutions de *clusterings* de base. Cette observation n'est pas souvent positive, si ce n'est pour le jeu de donnée *Parkinson* pour lequel la stratégie maximum n'avait aucun impact sur l'heuristique CoBoC consensus.

Pour dresser le bilan des différentes observations de CoBoC pour la combinaison de modèles, la stratégie *random* permet d'obtenir souvent le meilleur consensus, mais les causes de cette observation n'ont pu être déterminées. La stratégie maximum permet parfois d'obtenir un meilleur consensus mais celui-ci est limité. Enfin la stratégie minimum est peu pertinente dans ce contexte.

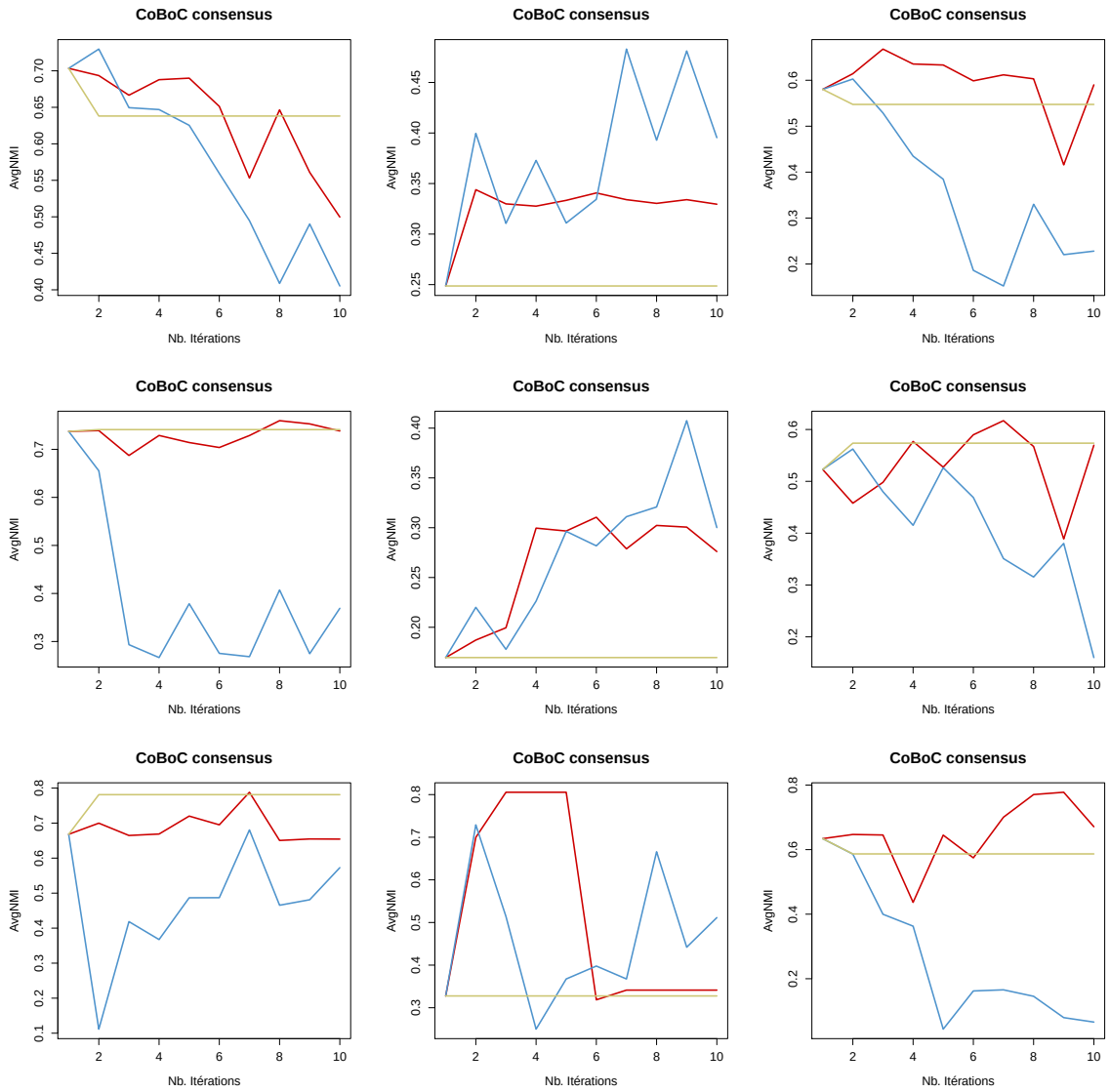

FIGURE 4.3 — Évolution de l'$AvgNMI$ pour la combinaison de modèle et l'heuristique consensus. Dans l'ordre, les données *iris*, *parkinson* et *wine*. Les trois lignes correspondent (1) à l'application de deux algorithmes : KM et CLINK, (2) à l'application de quatre algorithmes : KM, SC, SLINK et CLINK, (3) à l'application de six algorithmes : KM, FKM, SC, SLINK, ALINK et CLINK.

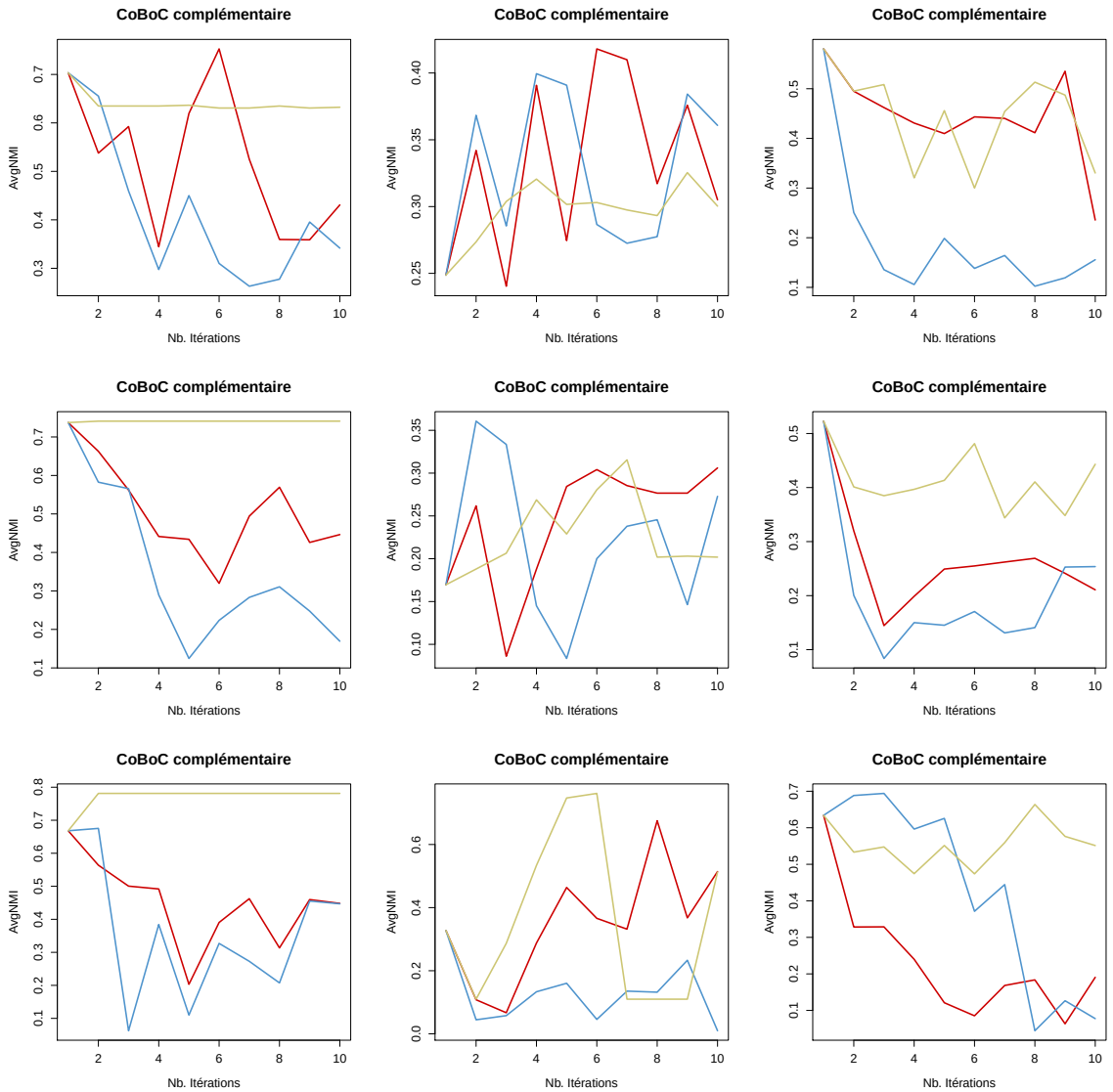

FIGURE 4.4 — Évolution de l'$AvgNMI$ pour la combinaison de modèle et l'heuristique complémentaire. Dans l'ordre, les données *iris*, *parkinson* et *wine*. Les trois lignes correspondent (1) à l'application de deux algorithmes : KM et CLINK, (2) à l'application de quatre algorithmes : KM, SC, SLINK et CLINK, (3) à l'application de six algorithmes : KM, FKM, SC, SLINK, ALINK et CLINK.

Évaluation interne de COBOC dans le cadre multi-vues. Les heuristiques consensus et complémentaire ont été observées sur une exécution dans le cadre de la recherche de consensus dans un contexte multi-vues sur le jeu de donnée *mfeat* (cf. figure 4.5 et 4.6).

Les résultats observés pour l'heuristique COBOC consensus (figure 4.5) sont semblables aux observations du contexte de la combinaison de modèles. De manière flagrante, les *clusterings* obtenus par la stratégie maximum peinent à s'éloigner des *clusterings* de base, pour la simple raison que les paires d'individus sélectionnées sont déja regroupées de la même manière par les algorithmes de *clustering* dans toutes les vues. La stratégie minimum permet d'atteindre brièvement une solution consensus pour un faible nombre d'échanges de contraintes, mais tend

davantage à produire des *clusterings* dissimilaires. Finalement, l'heuristique la plus pertinente sur l'exemple présenté est bien la stratégie *random*. Néanmoins l'étude réalisée ne permet pas d'identifier *pourquoi* c'est le cas.

Les observations issues des expériences sur l'heuristique complémentaire (figure 4.6) sont ici sensiblement différentes, si ce n'est pour l'inefficacité de la stratégie minimum. La stratégie *random* ne permet pas d'atteindre un consensus. En revanche, la stratégie maximum, elle, réussit à l'atteindre.

On ne peut dégager la meilleure des approches à considérer dans le contexte multi-vues, puisque la stratégie aléatoire pour l'heuristique CoBoC consensus atteint les mêmes performances en terme d'information mutuelle normalisée que la stratégie maximum pour l'heuristique CoBoC complémentaire. De plus, aucune similitude analytique ne peut être mise en évidence entre ces deux approches. Globalement, concernant la stratégie minimum, on constate que si celle-ci est intuitive, puisqu'elle permet d'aider globalement la décision sur les paires d'individus pour lesquels les différents algorithmes locaux peinent à décider, elle n'est néanmoins presque jamais efficace.

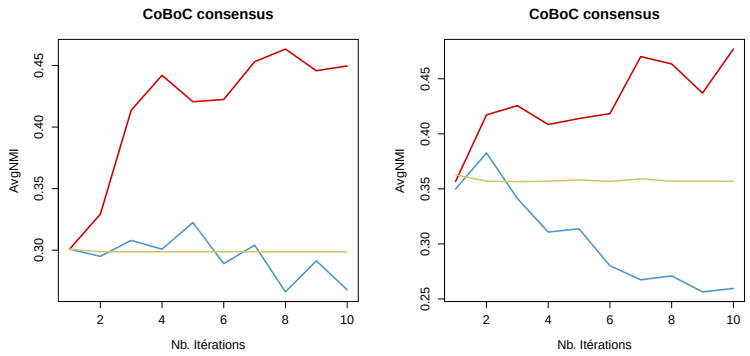

FIGURE 4.5 — Évolution de l'$AvgNMI$ pour le *clustering* multi-vues et l'heuristique consensus. Dans l'ordre, les données *mfeat* avec le même algorithme pour toutes les six vues, et *mfeat* avec des algorithmes différents pour chaque vue.

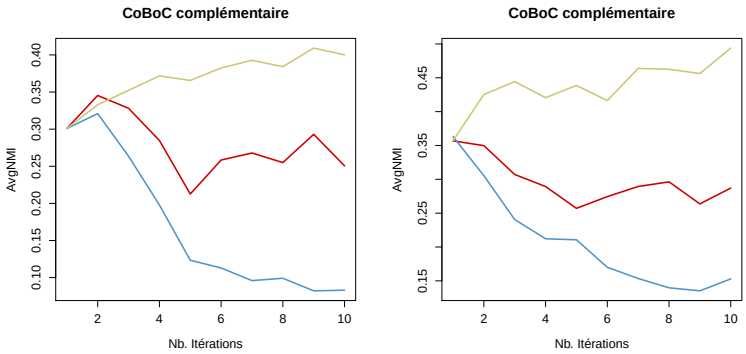

FIGURE 4.6 — Évolution de l'$AvgNMI$ pour le *clustering* multi-vues et l'heuristique complémentaire. Dans l'ordre, les données *mfeat* avec le même algorithme pour toutes les six vues, et *mfeat* avec des algorithmes différents pour chaque vue.

Évaluation interne de ALTERBOC

L'évaluation interne d'ALTERBOC vise, contrairement à COBOC, à observer une diminution de la valeur d'information mutuelle normalisée moyenne entre les résultats des algorithmes locaux.

Évaluation interne de ALTERBOC dans le cadre de la multiplicité des modèles. Les heuristiques consensus et complémentaire ont été observées sur une exécution dans le cadre de la combinaison de modèles (cf. figure 4.7 et 4.8).

L'heuristique ALTERBOC global vise à encourager les algorithmes locaux à rechercher des solutions de *clusterings* différentes. Selon cet objectif, les trois stratégies (minimum, maximum et *random*) parviennent à atteindre de bonnes solutions. Cependant, la stratégie *random* permet d'obtenir la meilleure tendance. Les performances des stratégies minimum et maximum sont interverties selon les jeux de données. Enfin, dans le cas général, les *clusterings* alternatifs sont obtenus plutôt pour un faible nombre d'échanges de contraintes. Un trop grand nombre de contraintes échangées tend à reproduire une forme de consensus faible.

L'heuristique ALTERBOC complémentaire permet également d'atteindre des solutions alternatives et les stratégies associées ont un comportement semblable à celui de la précédente heuristique. On remarque également le danger de réaliser un nombre trop élevé d'échanges de contraintes, notamment dans le cas du jeu de données *Parkinson* avec six algorithmes.

On remarque globalement que l'on peut atteindre différentes formes d'alternatives avec toutes les stratégies. En revanche, les expériences montrent qu'il est recommandé dans ce contexte de limiter le nombre d'échanges de contraintes entre les vues, sous peine de finir par atteindre une solution consensus de faible qualité.

Évaluation interne de ALTERBOC dans le cadre multi-vues. Les heuristiques consensus et complémentaire ont été observées sur une exécution dans le cadre de la recherche de *clusterings* alternatifs dans un contexte de multiplicité des vues sur le jeu de donnée *mfeat* (cf. figure 4.9 et 4.10).

Les heuristiques ALTERBOC global et ALTERBOC complémentaire satisfont toutes les deux l'objectif, quelles que soient les stratégies employées. L'obtention de solutions de *clusterings* réellement différentes est cependant plus nette pour l'heuristique complémentaire. Pour l'heuristique consensus, on constate qu'encore une fois la stratégie *random* est la meilleure pour atteindre l'objectif, alors que la stratégie minimum atteint un meilleur ensemble de *clusterings* que la stratégie maximum.

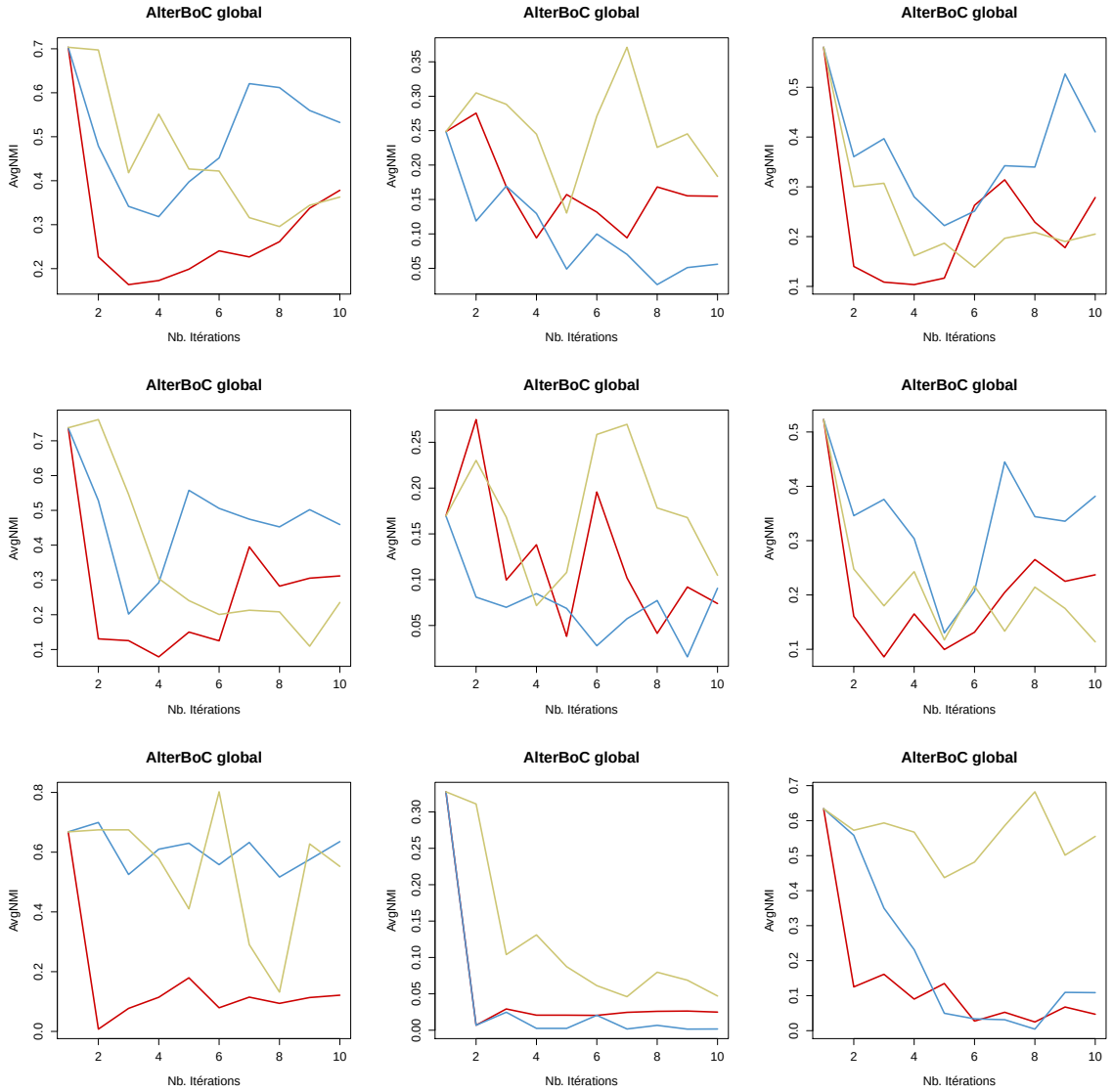

FIGURE 4.7 — Évolution de l'$AvgNMI$ pour la combinaison de modèle et l'heuristique global. Dans l'ordre, les données *iris*, *parkinson* et *wine*. Les trois lignes correspondent (1) à l'application de deux algorithmes : KM et CLINK, (2) à l'application de quatre algorithmes : KM, SC, SLINK et CLINK, (3) à l'application de six algorithmes : KM, FKM, SC, SLINK, ALINK et CLINK.

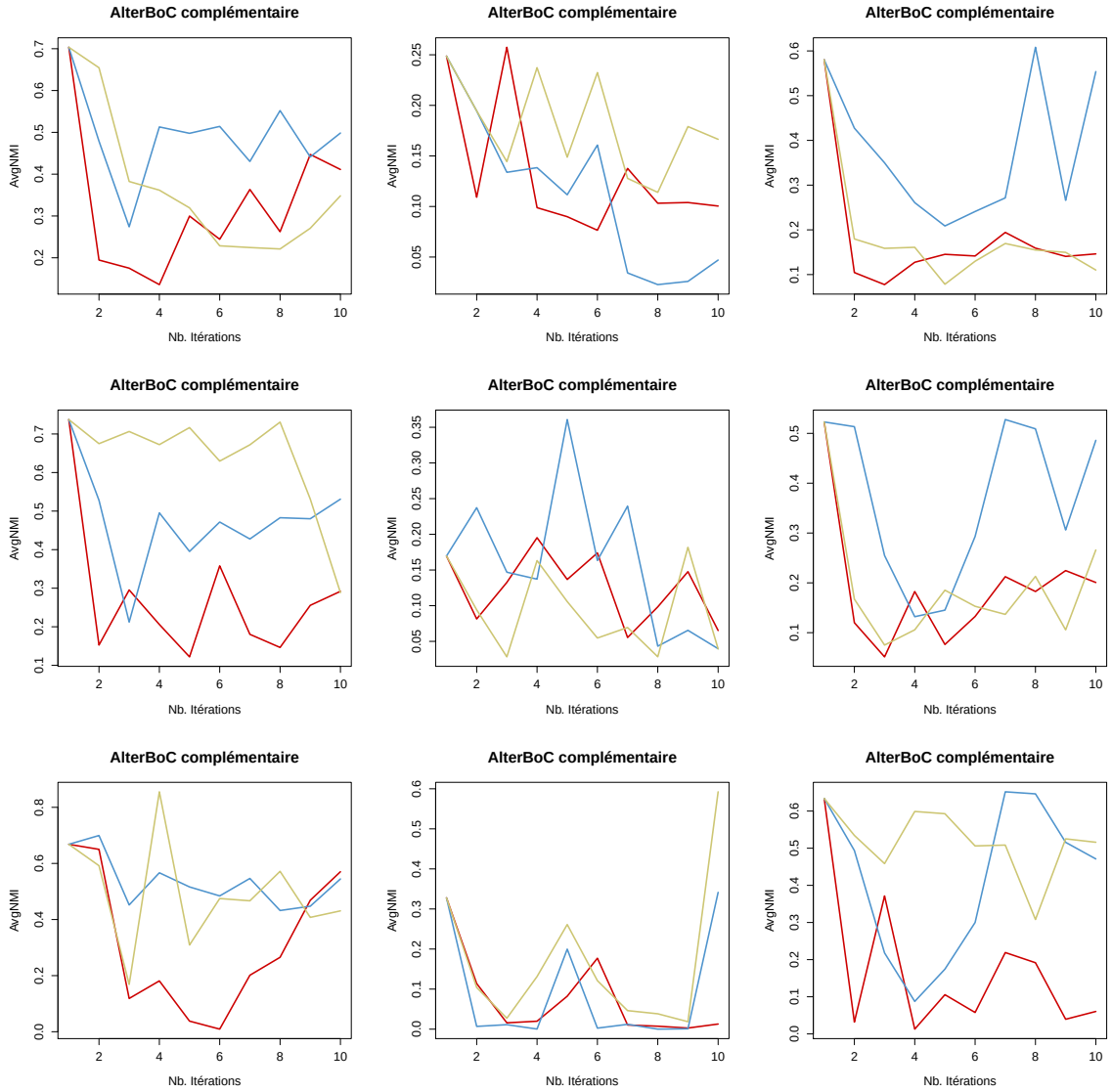

FIGURE 4.8 — Évolution de l'$AvgNMI$ pour la combinaison de modèle et l'heuristique complémentaire. Dans l'ordre, les données *iris*, *parkinson* et *wine*. Les trois lignes correspondent (1) à l'application de deux algorithmes : KM et CLINK, (2) à l'application de quatre algorithmes : KM, SC, SLINK et CLINK, (3) à l'application de six algorithmes : KM, FKM, SC, SLINK, ALINK et CLINK.

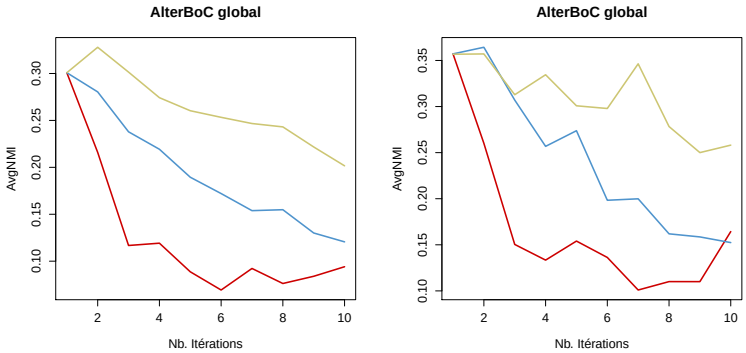

FIGURE 4.9 — Évolution de l'$AvgNMI$ pour le *clustering* multi-vues et l'heuristique global. Dans l'ordre, les données *mfeat* avec le même algorithme pour toutes les six vues, et *mfeat* avec des algorithmes différents pour chaque vue.

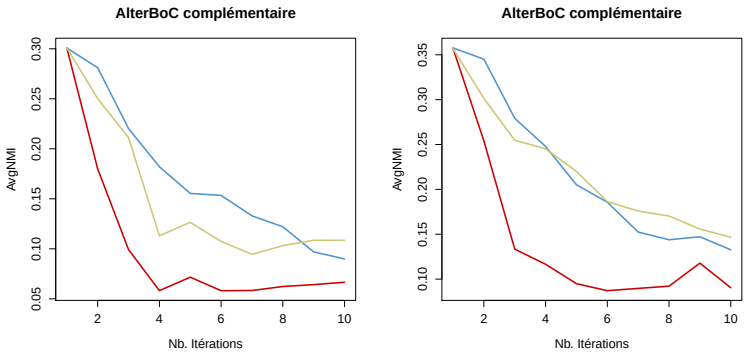

FIGURE 4.10 — Évolution de l'$AvgNMI$ pour le *clustering* multi-vues et l'heuristique complémentaire. Dans l'ordre, les données *mfeat* avec le même algorithme pour toutes les six vues, et *mfeat* avec des algorithmes différents pour chaque vue.

4.7.3 Évaluation externe

Évaluation externe de CoBoC pour la combinaison de modèles

L'apport des deux heuristiques CoBoC consensus et CoBoC complémentaire ainsi que des stratégies associées (Γ_{Random}, Γ_{Min} et Γ_{Max}) est d'abord observé sur les données *Iris*, *Wine* et *parkinson*. Le contexte est celui de la combinaison de modèles, où un ensemble d'algorithmes de *clustering* est appliqué à un jeu de donnée classique mono-vue.

Apport de la collaboration à chaque algorithme pour la combinaison de modèles. Le tableau 4.1 servant de référence dans ce paragraphe montre les résultats obtenus sur les jeux de données *Iris*, *Parkinson* et *Wine* avec utilisation de six algorithmes différents. L'objectif ici est d'observer les performances des différentes approches de recherche de consensus par CoBoC relativement à ces résultats.

	% *F-mesure*			*AvgEnt*			*NMI*		
Iris : Algorithmes de *clustering* locaux									
KM vue 0	73.39	±	3.21	0.29	±	0.03	0.65	±	0.02
FKM vue 1	74.52	±	0	0.24	±	0	0.66	±	0
SC vue 2	73.35	±	0	0.25	±	0	0.63	±	0
SLink vue 3	68.64	±	0	0.31	±	0	0.59	±	0
ALink vue 4	72.06	±	0	0.27	±	0	0.65	±	0
CLink vue 5	72.54	±	0	0.26	±	0	0.65	±	0
Wine : Algorithmes de *clustering* locaux									
KM vue 0	92.96	±	0.69	0.13	±	0.01	0.87	±	0.01
FKM vue 1	93.19	±	0	0.13	±	0	0.88	±	0
SC vue 2	93.57	±	0	0.1	±	0	0.9	±	0
SLink vue 3	59.07	±	0	0.51	±	0	0.37	±	0
ALink vue 4	68.8	±	0	0.42	±	0	0.59	±	0
CLink vue 5	71.94	±	0	0.26	±	0	0.61	±	0
Parkinson : Algorithmes de *clustering* locaux									
KM vue 0	62.51	±	2.48	0.25	±	0	0.12	±	0.02
FKM vue 1	62.49	±	0	0.25	±	0	0.12	±	0
SC vue 2	61.3	±	0	0.29	±	0	0.2	±	0
SLink vue 3	76.14	±	0	0.49	±	0	0.01	±	0
ALink vue 4	75.22	±	0	0.25	±	0	0.02	±	0
CLink vue 5	70.8	±	0	0.25	±	0	0.05	±	0

Tableau 4.1 — Évaluation externe de CoBoC consensus sur *Iris* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC.

Les tableaux 4.2 à 4.7 montrent les résultats obtenus par chaque algorithme de *clustering* localement, avec collaboration par CoBoC.

On constate tout d'abord que dans la grande majorité des cas, l'heuristique CoBoC consensus associée à la stratégie Γ_{Max} ne réalise aucun apport. Ceci est dû au fait que les couples sélectionnés comme des contraintes $\mathcal{ML}$ (respectivement $\mathcal{CL}$), de confiance maximale sont le plus souvent les couples déjà regroupés ensemble (respectivement séparés) dans toutes les vues. Cette observation par critère externe conforte les observations réalisées par l'évaluation interne. Ce résultat n'est néanmoins pas toujours le cas, dans la mesure où quelques algorithmes de *clustering* peuvent se comporter de façons différentes sur le regroupement de ces couples et être ainsi corrigés pour se rapprocher des autres algorithmes. Dans ce contexte et avec la stratégie Γ_{Max}, l'heuristique CoBoC complémentaire se comporte alors de façon semblable.

La stratégie Γ_{Min}, quelquesoit l'heuristique CoBoC consensus ou CoBoC complémentaire, tend à rapprocher les performances des différents algorithmes employés. Cependant l'apport ne semble intéressant que pour les jeux de données difficiles pour les algorithmes classiques (*parkinson*). Autrement la performance est systématiquement dégradée. Cette observation est intéressante puisqu'elle corrobore l'observation que, sur le jeu de donnée *Parkinson*, la stratégie minimum permettait d'atteindre un consensus par CoBoC (cf. section 4.7.2). Ceci donne une indication sur la pertinence de rechercher un tel consensus pour améliorer la performance des algorithmes de *clusterings* que l'on souhaite combiner.

La stratégie aléatoire Γ_{Random} peut aider à améliorer certains algorithmes, notamment sur *Iris* (Tab. 4.1) ou sur *Wine* (Tab. 4.4). Dans tous les cas, aucune tendance générale vers une amélioration ne peut être dégagée à partir des heuristiques et stratégies proposées. Pris isoléments, les algorithmes proposés, avec collaboration n'améliore pas en terme de mesure de performance externe, les algorithmes classiques.

Iris	*% F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	77.69	±	9.91	0.25	±	0.08	0.69	±	0.12
CoBoC vue 1	70.25	±	17.24	0.41	±	0.28	0.58	±	0.25
CoBoC vue 2	81.98	±	6.2	0.22	±	0.05	0.74	±	0.07
CoBoC vue 3	70.85	±	4.28	0.3	±	0.09	0.62	±	0.08
CoBoC vue 4	69.97	±	5.82	0.33	±	0.13	0.64	±	0.09
CoBoC vue 5	71.01	±	8.21	0.3	±	0.13	0.64	±	0.11
Stratégie Γ_{Min}									
CoBoC vue 0	54.27	±	13.5	0.64	±	0.25	0.37	±	0.2
CoBoC vue 1	55.86	±	12.68	0.59	±	0.23	0.4	±	0.18
CoBoC vue 2	57.51	±	16.22	0.54	±	0.25	0.4	±	0.24
CoBoC vue 3	62.66	±	11.31	0.52	±	0.3	0.46	±	0.25
CoBoC vue 4	64.95	±	10.77	0.42	±	0.22	0.52	±	0.18
CoBoC vue 5	59.05	±	13.48	0.59	±	0.33	0.42	±	0.24
Stratégie Γ_{Max}									
CoBoC vue 0	73.42	±	3.22	0.29	±	0.03	0.65	±	0.02
CoBoC vue 1	74.52	±	0	0.24	±	0	0.66	±	0
CoBoC vue 2	71.31	±	0	0.29	±	0	0.61	±	0
CoBoC vue 3	68.29	±	0	0.32	±	0	0.58	±	0
CoBoC vue 4	59.13	±	0	0.39	±	0	0.45	±	0
CoBoC vue 5	73.07	±	0	0.32	±	0	0.69	±	0

Tableau 4.2 — Évaluation externe de CoBoC consensus sur *Iris* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC.

Iris	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	74.39	±	2.37	0.27	±	0.05	0.65	±	0.02
CoBoC vue 1	72.09	±	8.28	0.3	±	0.12	0.61	±	0.13
CoBoC vue 2	68.95	±	7.69	0.34	±	0.14	0.56	±	0.12
CoBoC vue 3	68.47	±	5.39	0.34	±	0.1	0.57	±	0.11
CoBoC vue 4	67.46	±	7.55	0.4	±	0.21	0.58	±	0.14
CoBoC vue 5	66.37	±	9.78	0.39	±	0.18	0.56	±	0.14
Stratégie Γ_{Min}									
CoBoC vue 0	67.59	±	6.12	0.34	±	0.12	0.56	±	0.1
CoBoC vue 1	61.79	±	13.94	0.47	±	0.26	0.47	±	0.2
CoBoC vue 2	48.95	±	11.06	0.75	±	0.23	0.25	±	0.18
CoBoC vue 3	68.53	±	5.35	0.36	±	0.15	0.58	±	0.11
CoBoC vue 4	62.97	±	9.26	0.42	±	0.15	0.49	±	0.17
CoBoC vue 5	53.23	±	11.02	0.7	±	0.25	0.32	±	0.2
Stratégie Γ_{Max}									
CoBoC vue 0	72.59	±	2.64	0.27	±	0.03	0.64	±	0.02
CoBoC vue 1	74.52	±	0	0.24	±	0	0.66	±	0
CoBoC vue 2	71.31	±	0	0.29	±	0	0.61	±	0
CoBoC vue 3	68.29	±	0	0.32	±	0	0.58	±	0
CoBoC vue 4	60.35	±	1	0.39	±	0.01	0.44	±	0
CoBoC vue 5	74	±	0.76	0.33	±	0.01	0.66	±	0.02

Tableau 4.3 — Évaluation externe de CoBoC complémentaire sur *Iris* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC.

Wine	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	77.67	±	5.63	0.28	±	0.06	0.68	±	0.08
CoBoC vue 1	78.36	±	5.18	0.26	±	0.02	0.68	±	0.06
CoBoC vue 2	77.93	±	5.7	0.28	±	0.04	0.67	±	0.06
CoBoC vue 3	58.56	±	4.99	0.51	±	0.21	0.4	±	0.11
CoBoC vue 4	69.47	±	12.74	0.39	±	0.22	0.57	±	0.22
CoBoC vue 5	71.27	±	7.24	0.36	±	0.18	0.62	±	0.09
Stratégie Γ_{Min}									
CoBoC vue 0	62.65	±	4.56	0.52	±	0.14	0.47	±	0.06
CoBoC vue 1	60.96	±	3.48	0.47	±	0.06	0.44	±	0.04
CoBoC vue 2	68.44	±	6.81	0.41	±	0.1	0.53	±	0.08
CoBoC vue 3	54.39	±	4.97	0.77	±	0.28	0.28	±	0.1
CoBoC vue 4	56.03	±	6.36	0.76	±	0.27	0.27	±	0.19
CoBoC vue 5	55.13	±	7.34	0.59	±	0.19	0.33	±	0.12
Stratégie Γ_{Max}									
CoBoC vue 0	92.96	±	0.69	0.13	±	0.01	0.87	±	0.01
CoBoC vue 1	93.19	±	0	0.13	±	0	0.88	±	0
CoBoC vue 2	93.57	±	0	0.1	±	0	0.9	±	0
CoBoC vue 3	61.34	±	0	0.4	±	0	0.47	±	0
CoBoC vue 4	59.42	±	0	0.56	±	0	0.37	±	0
CoBoC vue 5	67.7	±	0	0.42	±	0	0.58	±	0

Tableau 4.4 — Évaluation externe de CoBoC consensus sur *Wine* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC.

Wine	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	75.12	±	4.82	0.31	±	0.04	0.64	±	0.06
CoBoC vue 1	77.24	±	3.76	0.29	±	0.03	0.66	±	0.05
CoBoC vue 2	78.78	±	5.05	0.31	±	0.05	0.68	±	0.06
CoBoC vue 3	59.39	±	4.5	0.48	±	0.19	0.42	±	0.11
CoBoC vue 4	68.25	±	8.8	0.42	±	0.21	0.54	±	0.19
CoBoC vue 5	73.06	±	8.45	0.37	±	0.18	0.63	±	0.12
Stratégie Γ_{Min}									
CoBoC vue 0	67.19	±	8.4	0.41	±	0.12	0.51	±	0.12
CoBoC vue 1	63.14	±	8.36	0.47	±	0.08	0.44	±	0.12
CoBoC vue 2	77.9	±	5.85	0.32	±	0.05	0.65	±	0.07
CoBoC vue 3	55.03	±	6.44	0.83	±	0.25	0.24	±	0.18
CoBoC vue 4	59.9	±	5.28	0.53	±	0.28	0.4	±	0.17
CoBoC vue 5	59.12	±	7.23	0.58	±	0.23	0.4	±	0.14
Stratégie Γ_{Max}									
CoBoC vue 0	92.96	±	0.69	0.13	±	0.01	0.87	±	0.01
CoBoC vue 1	93.19	±	0	0.13	±	0	0.88	±	0
CoBoC vue 2	93.57	±	0	0.1	±	0	0.9	±	0
CoBoC vue 3	61.34	±	0	0.4	±	0	0.47	±	0
CoBoC vue 4	59.42	±	0	0.56	±	0	0.37	±	0
CoBoC vue 5	67.7	±	0	0.42	±	0	0.58	±	0

Tableau 4.5 — Évaluation externe de CoBoC complémentaire sur *Wine* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC.

parkinson	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	70.76	±	0.29	0.25	±	0	0.05	±	0
CoBoC vue 1	70.97	±	0.21	0.25	±	0	0.05	±	0
CoBoC vue 2	60.28	±	0.9	0.32	±	0.01	0.15	±	0.01
CoBoC vue 3	75.77	±	0.45	0.39	±	0.12	0.01	±	0
CoBoC vue 4	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC vue 5	74.69	±	1.22	0.25	±	0	0.02	±	0.01
Stratégie Γ_{Min}									
CoBoC vue 0	71.16	±	0.42	0.25	±	0	0.04	±	0.01
CoBoC vue 1	65.77	±	6.09	0.26	±	0.02	0.09	±	0.06
CoBoC vue 2	65.75	±	5.55	0.32	±	0.07	0.08	±	0.06
CoBoC vue 3	75.82	±	1.88	0.38	±	0.12	0.04	±	0.06
CoBoC vue 4	73.37	±	4.89	0.3	±	0.1	0.04	±	0.06
CoBoC vue 5	72.61	±	3.33	0.25	±	0	0.04	±	0.02
Stratégie Γ_{Max}									
CoBoC vue 0	62.51	±	2.48	0.25	±	0	0.12	±	0.02
CoBoC vue 1	62.49	±	0	0.25	±	0	0.12	±	0
CoBoC vue 2	61.3	±	0	0.29	±	0	0.2	±	0
CoBoC vue 3	76.14	±	0	0.49	±	0	0.01	±	0
CoBoC vue 4	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC vue 5	70.8	±	0	0.25	±	0	0.05	±	0

Tableau 4.6 — Évaluation externe de CoBoC consensus sur *parkinson* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC.

parkinson	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	70.97	±	0.28	0.25	±	0	0.05	±	0
CoBoC vue 1	70.97	±	0.21	0.25	±	0	0.05	±	0
CoBoC vue 2	60.02	±	0.71	0.32	±	0.02	0.15	±	0.02
CoBoC vue 3	75.96	±	0.37	0.44	±	0.1	0.01	±	0
CoBoC vue 4	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC vue 5	74.16	±	1.83	0.25	±	0	0.03	±	0.01
Stratégie Γ_{Min}									
CoBoC vue 0	64.05	±	5.73	0.27	±	0.02	0.11	±	0.06
CoBoC vue 1	62.36	±	5.54	0.27	±	0.02	0.14	±	0.06
CoBoC vue 2	61.7	±	3.09	0.31	±	0.02	0.15	±	0.05
CoBoC vue 3	73.61	±	1.67	0.3	±	0.1	0.03	±	0.01
CoBoC vue 4	74.08	±	2.71	0.27	±	0.07	0.03	±	0.02
CoBoC vue 5	70.34	±	4.68	0.25	±	0	0.06	±	0.04
Stratégie Γ_{Max}									
CoBoC vue 0	62.51	±	2.48	0.25	±	0	0.12	±	0.02
CoBoC vue 1	62.49	±	0	0.25	±	0	0.12	±	0
CoBoC vue 2	61.3	±	0	0.29	±	0	0.2	±	0
CoBoC vue 3	76.14	±	0	0.49	±	0	0.01	±	0
CoBoC vue 4	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC vue 5	70.8	±	0	0.25	±	0	0.05	±	0

Tableau 4.7 — Évaluation externe de CoBoC complémentaire sur *parkinson* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC.

Apport de la fusion finale par le noyau K_1 et K_2. Le tableaux 4.8 servant de référence dans ce paragraphe montre les résultats obtenus sur les jeux de données *Iris*, *Parkinson* et *Wine* avec application pour chacun de l'algorithme CoFKM (2.4.2) dans ses déclinaisons *a priori* et *a posteriori*. L'objectif ici est d'observer l'impact des différentes approches de recherche de consensus par CoBoC sur différentes solutions de fusion adaptées à la combinaison de modèles pour la recherche de consensus, et relativement aux résultats des approches multi-vues.

	% *F-mesure*			*AvgEnt*			*NMI*		
Iris : Approche multi-vues CoFKM									
CoFKM post	70.53	±	6.28	0.34	±	0.14	0.62	±	0.11
CoFKM	74.52	±	0	0.24	±	0	0.66	±	0
CoFKM concat	74.52	±	0	0.24	±	0	0.66	±	0
Wine : Approche multi-vues CoFKM									
CoFKM post	81.28	±	11	0.24	±	0.09	0.73	±	0.13
CoFKM	93.19	±	0	0.13	±	0	0.88	±	0
CoFKM concat	93.19	±	0	0.13	±	0	0.88	±	0
Parkinson : Approche multi-vues CoFKM									
CoFKM post	65.12	±	4.38	0.25	±	0	0.1	±	0.03
CoFKM	62.06	±	0.34	0.25	±	0	0.12	±	0
CoFKM concat	62.06	±	0.34	0.25	±	0	0.12	±	0

Tableau 4.8 — Évaluation externe de CoFKM dans le contexte de la combinaison de modèles.

La fusion finale permet de construire une solution unique consensus entre les différentes solutions locales obtenues. Dans ce paragraphe sont étudiés les noyaux K_1 (4.41) et K_2 (4.42)

considérés comme des mesures de similarité sur les paires d'individus. Deux individus x_i et x_j sont alors similaires si ils sont souvent regroupés ensemble par les différents algorithmes de *clustering* $\{A^{(r)}\}$.

À partir de ces mesures de similarité, des algorithmes spécifiques sont utilisés pour construire le *clustering* final. Les algorithmes implémentés sont SLINK, ALINK, CLINK, KKM et KFKM. L'adjonction de la fusion finale avec COBOC place l'approche dans un contexte multi-vues. Les différents algorithmes employés pour la fusion sont alors comparés à l'approche multi-vues COFKM appliquée sur les donnnées classiques. Les différentes vues des données sont identiques ici car les jeux de données employés sont mono-vue. Il sont alors recopiés autant de fois que d'algorithmes ont été utilisés dans l'approche COBOC.

Globalement, pour le noyau K_1, la stratégie Γ_{Max} se comporte bien quelque soit l'heuristique. En revanche les autres stratégies et heuristiques ne parviennent pas à dépasser l'approche multi-vue de référence (Tab. 4.8). Une amélioration flagrante est néanmoins obtenue pour l'heuristique COBOC consensus et la stratégie Γ_{Random} (Tab. 4.9). Dans ce dernier cas la performance obtenue dépasse également celles des approches classiques (Tab. 4.1). La stratégie Γ_{Min} n'est pas efficace.

Concernant le noyau K_2, les différentes stratégies sont plus ou moins efficaces selon les jeux de données et les critères d'évaluations. La stratégie Γ_{Random} est plus efficace sur *Iris* (Tab. 4.9 ou Tab. 4.10 par le *clustering* par lien moyen) ou bien encore sur *parkinson* pour l'heuristique COBOC consensus (meilleure *F-mesure* ou meilleure *NMI*, Tab. 4.13). La stratégie Γ_{Min} est encore une fois rarement efficace, mais parvient à avoir de bonnes performances sur *parkinson* pour l'heuristique COBOC complémentaire. Le résultat le plus intéressant est l'obtention du meilleur score sur *Wine* pour la stratégie Γ_{Max}, meilleur que l'approche COFKM (Tab. 4.8), ou que l'application des algorithmes classiques (Tab. 4.1).

Iris	*% F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	74.33	±	7.18	0.29	±	0.12	0.67	±	0.1
CoBoC ALink	75.84	±	6.77	0.25	±	0.06	0.69	±	0.08
CoBoC CLink	75.37	±	6.64	0.27	±	0.05	0.68	±	0.08
CoBoC KKM	77.38	±	7.19	0.25	±	0.05	0.7	±	0.08
CoBoC KFKM	78.22	±	8.24	0.27	±	0.11	0.71	±	0.09
CoBoC SC	74.05	±	11.13	0.42	±	0.25	0.65	±	0.18
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	60.9	±	11.66	0.52	±	0.26	0.44	±	0.22
CoBoC ALink	63.08	±	12.09	0.5	±	0.29	0.48	±	0.21
CoBoC CLink	60.43	±	12.51	0.54	±	0.3	0.44	±	0.22
CoBoC KKM	58.11	±	13.69	0.56	±	0.3	0.42	±	0.22
CoBoC KFKM	60.22	±	12.59	0.51	±	0.25	0.46	±	0.18
CoBoC SC	64.46	±	10.55	0.53	±	0.24	0.55	±	0.16
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	71.17	±	2.13	0.26	±	0.01	0.62	±	0.03
CoBoC ALink	72.67	±	0.64	0.29	±	0.09	0.65	±	0.02
CoBoC CLink	73.44	±	1.13	0.26	±	0.03	0.66	±	0.01
CoBoC KKM	72.87	±	2.21	0.31	±	0.13	0.64	±	0.01
CoBoC KFKM	73.09	±	0.52	0.26	±	0.01	0.64	±	0.01
CoBoC SC	70.68	±	4.99	0.3	±	0.05	0.6	±	0.06
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	73.31	±	6.22	0.32	±	0.13	0.66	±	0.07
CoBoC ALink	75.74	±	5.77	0.29	±	0.13	0.69	±	0.06
CoBoC CLink	74.32	±	5.67	0.27	±	0.06	0.67	±	0.07
CoBoC KKM	73.22	±	9.59	0.39	±	0.23	0.66	±	0.12
CoBoC KFKM	78.67	±	6.92	0.27	±	0.11	0.71	±	0.08
CoBoC SC	72.92	±	10.77	0.45	±	0.27	0.65	±	0.17
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	67.82	±	8.01	0.38	±	0.23	0.56	±	0.15
CoBoC ALink	70.03	±	7.53	0.36	±	0.2	0.61	±	0.13
CoBoC CLink	70.63	±	6.12	0.29	±	0.1	0.62	±	0.08
CoBoC KKM	64.9	±	9.92	0.49	±	0.23	0.55	±	0.15
CoBoC KFKM	68.08	±	8.81	0.33	±	0.15	0.57	±	0.12
CoBoC SC	66.66	±	8.4	0.49	±	0.24	0.58	±	0.14
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	70.64	±	1.55	0.3	±	0.01	0.61	±	0.02
CoBoC ALink	71.66	±	0.76	0.31	±	0.08	0.63	±	0.03
CoBoC CLink	72.32	±	0.91	0.28	±	0.01	0.62	±	0.01
CoBoC KKM	72.05	±	3.78	0.31	±	0.13	0.63	±	0.06
CoBoC KFKM	73.22	±	0.68	0.26	±	0.01	0.64	±	0.01
CoBoC SC	71.59	±	2.1	0.33	±	0.13	0.62	±	0.01

Tableau 4.9 — Évaluation externe de CoBoC consensus sur *Iris* selon différentes fusions finales pour les noyaux K_1 et K_2.

Iris	*% F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	71.97	±	5.28	0.3	±	0.07	0.62	±	0.09
CoBoC ALink	73.48	±	2.3	0.36	±	0.16	0.66	±	0.03
CoBoC CLink	73.94	±	2.56	0.29	±	0.05	0.65	±	0.03
CoBoC KKM	72.35	±	4.73	0.31	±	0.1	0.63	±	0.06
CoBoC KFKM	74.09	±	1.6	0.3	±	0.04	0.64	±	0.02
CoBoC SC	69.82	±	6.31	0.47	±	0.22	0.62	±	0.07
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	65.87	±	5.7	0.36	±	0.1	0.53	±	0.1
CoBoC ALink	71.47	±	2.29	0.31	±	0.09	0.62	±	0.03
CoBoC CLink	69.1	±	7.19	0.32	±	0.13	0.58	±	0.12
CoBoC KKM	66.81	±	7.23	0.34	±	0.11	0.55	±	0.11
CoBoC KFKM	70.67	±	2.43	0.27	±	0.03	0.6	±	0.03
CoBoC SC	70.09	±	4.13	0.33	±	0.15	0.61	±	0.05
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	65.63	±	7.99	0.36	±	0.11	0.51	±	0.14
CoBoC ALink	73.56	±	0.8	0.28	±	0.09	0.65	±	0.02
CoBoC CLink	71.79	±	1.64	0.27	±	0.01	0.63	±	0.01
CoBoC KKM	73.2	±	1.76	0.28	±	0.09	0.64	±	0.02
CoBoC KFKM	73.7	±	0.67	0.26	±	0.02	0.64	±	0.01
CoBoC SC	72.86	±	1.63	0.32	±	0.13	0.63	±	0.01
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	73.17	±	2.4	0.28	±	0.03	0.64	±	0.03
CoBoC ALink	74.71	±	2.05	0.28	±	0.04	0.67	±	0.02
CoBoC CLink	73.64	±	2.65	0.27	±	0.05	0.65	±	0.03
CoBoC KKM	71.87	±	6.34	0.36	±	0.17	0.62	±	0.1
CoBoC KFKM	73.81	±	2.3	0.31	±	0.09	0.65	±	0.02
CoBoC SC	68.98	±	5.9	0.49	±	0.22	0.61	±	0.07
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	70.7	±	2.04	0.29	±	0.02	0.62	±	0.03
CoBoC ALink	71.28	±	1.79	0.31	±	0.08	0.63	±	0.03
CoBoC CLink	71.93	±	1.95	0.28	±	0.03	0.63	±	0.02
CoBoC KKM	69.29	±	4.87	0.34	±	0.14	0.6	±	0.06
CoBoC KFKM	71.13	±	2.2	0.27	±	0.03	0.61	±	0.02
CoBoC SC	69.89	±	3.01	0.4	±	0.19	0.62	±	0.02
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	72.32	±	0.91	0.28	±	0.01	0.62	±	0.01
CoBoC ALink	72.4	±	0.66	0.27	±	0.01	0.64	±	0.01
CoBoC CLink	71.9	±	0.68	0.29	±	0.01	0.62	±	0.01
CoBoC KKM	70.25	±	7.75	0.36	±	0.22	0.59	±	0.13
CoBoC KFKM	74.38	±	0.41	0.24	±	0.01	0.66	±	0.01
CoBoC SC	72.22	±	1.58	0.3	±	0.09	0.62	±	0.01

Tableau 4.10 — Évaluation externe de CoBoC complémentaire sur *Iris* selon différentes fusions finales pour les noyaux K_1 et K_2.

Wine	*% F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	71.56	±	7.02	0.29	±	0.09	0.64	±	0.08
CoBoC ALink	71.48	±	5.08	0.31	±	0.08	0.63	±	0.07
CoBoC CLink	73.24	±	4.87	0.28	±	0.08	0.66	±	0.05
CoBoC KKM	74.28	±	9.86	0.32	±	0.09	0.63	±	0.13
CoBoC KFKM	79	±	4.72	0.27	±	0.04	0.7	±	0.05
CoBoC SC	81.35	±	8.57	0.23	±	0.05	0.73	±	0.1
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	52.63	±	5.83	0.79	±	0.21	0.32	±	0.13
CoBoC ALink	66.79	±	9.7	0.47	±	0.2	0.5	±	0.18
CoBoC CLink	60.74	±	11.4	0.55	±	0.29	0.42	±	0.22
CoBoC KKM	67.79	±	10.12	0.37	±	0.11	0.54	±	0.15
CoBoC KFKM	73.64	±	8.98	0.37	±	0.14	0.61	±	0.11
CoBoC SC	75.9	±	10.47	0.31	±	0.08	0.65	±	0.11
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	86.68	±	0	0.17	±	0	0.8	±	0
CoBoC ALink	90.59	±	3.19	0.15	±	0.02	0.85	±	0.04
CoBoC CLink	90.59	±	3.19	0.15	±	0.02	0.85	±	0.04
CoBoC KKM	91.08	±	1.66	0.15	±	0.01	0.85	±	0.02
CoBoC KFKM	90.46	±	1.02	0.16	±	0.01	0.84	±	0.01
CoBoC SC	70.57	±	0.96	0.28	±	0.03	0.67	±	0.03
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	73.81	±	8.61	0.28	±	0.1	0.64	±	0.12
CoBoC ALink	73.54	±	4.79	0.29	±	0.08	0.67	±	0.05
CoBoC CLink	79.71	±	5.94	0.24	±	0.16	0.74	±	0.06
CoBoC KKM	75.53	±	7.86	0.34	±	0.15	0.66	±	0.09
CoBoC KFKM	80.1	±	4.73	0.26	±	0.03	0.71	±	0.05
CoBoC SC	77	±	15.61	0.32	±	0.24	0.66	±	0.24
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	64.84	±	5.11	0.47	±	0.14	0.52	±	0.1
CoBoC ALink	63.54	±	9.89	0.6	±	0.29	0.48	±	0.17
CoBoC CLink	65.99	±	10.05	0.49	±	0.23	0.53	±	0.19
CoBoC KKM	75.72	±	11.59	0.29	±	0.07	0.65	±	0.13
CoBoC KFKM	85.74	±	4.6	0.23	±	0.06	0.76	±	0.06
CoBoC SC	86.78	±	9.4	0.19	±	0.08	0.8	±	0.1
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	70.88	±	0.88	0.37	±	0.02	0.6	±	0.02
CoBoC ALink	86.66	±	9.97	0.16	±	0.04	0.81	±	0.1
CoBoC CLink	86.68	±	0	0.17	±	0	0.8	±	0
CoBoC KKM	83.95	±	11.93	0.21	±	0.11	0.77	±	0.13
CoBoC KFKM	91.31	±	0.66	0.14	±	0.01	0.85	±	0.01
CoBoC SC	94.6	±	0	0.08	±	0	0.91	±	0

Tableau 4.11 — Évaluation externe de CoBoC consensus sur *Wine* selon différentes fusions finales pour les noyaux K_1 et K_2.

Wine	*% F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	72.19	±	5.55	0.29	±	0.09	0.64	±	0.08
CoBoC ALink	70.82	±	3.37	0.33	±	0.07	0.64	±	0.04
CoBoC CLink	70.29	±	8.79	0.36	±	0.18	0.61	±	0.14
CoBoC KKM	78.59	±	7.4	0.29	±	0.2	0.69	±	0.11
CoBoC KFKM	81.05	±	3.02	0.24	±	0.03	0.73	±	0.04
CoBoC SC	83.65	±	7.26	0.22	±	0.05	0.76	±	0.07
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	61.11	±	5.62	0.58	±	0.21	0.44	±	0.11
CoBoC ALink	66.89	±	9.36	0.53	±	0.27	0.5	±	0.19
CoBoC CLink	65.77	±	9.03	0.53	±	0.28	0.49	±	0.19
CoBoC KKM	74.1	±	8	0.32	±	0.1	0.62	±	0.11
CoBoC KFKM	79.29	±	7.47	0.29	±	0.1	0.68	±	0.1
CoBoC SC	79.6	±	7.57	0.26	±	0.06	0.7	±	0.08
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	86.68	±	0	0.17	±	0	0.8	±	0
CoBoC ALink	90.59	±	3.19	0.15	±	0.02	0.85	±	0.04
CoBoC CLink	90.59	±	3.19	0.15	±	0.02	0.85	±	0.04
CoBoC KKM	91.28	±	1.5	0.15	±	0.01	0.85	±	0.02
CoBoC KFKM	90.46	±	1.02	0.16	±	0.01	0.84	±	0.01
CoBoC SC	70.57	±	0.96	0.28	±	0.03	0.67	±	0.03
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	69.95	±	4.86	0.31	±	0.09	0.6	±	0.08
CoBoC ALink	69.57	±	3.87	0.38	±	0.12	0.61	±	0.06
CoBoC CLink	74.55	±	6.23	0.26	±	0.08	0.69	±	0.07
CoBoC KKM	80.73	±	5.5	0.24	±	0.07	0.73	±	0.06
CoBoC KFKM	81.84	±	2.59	0.24	±	0.03	0.74	±	0.03
CoBoC SC	77.42	±	17.17	0.32	±	0.26	0.67	±	0.24
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	62.67	±	5.52	0.49	±	0.19	0.45	±	0.12
CoBoC ALink	70.65	±	12.39	0.45	±	0.29	0.57	±	0.2
CoBoC CLink	70.56	±	8.01	0.39	±	0.18	0.6	±	0.1
CoBoC KKM	79.54	±	7.78	0.28	±	0.09	0.69	±	0.1
CoBoC KFKM	83.14	±	5.51	0.25	±	0.07	0.73	±	0.08
CoBoC SC	86.02	±	5.1	0.19	±	0.05	0.79	±	0.05
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	70.88	±	0.88	0.37	±	0.02	0.6	±	0.02
CoBoC ALink	86.66	±	9.97	0.16	±	0.04	0.81	±	0.1
CoBoC CLink	86.68	±	0	0.17	±	0	0.8	±	0
CoBoC KKM	88.93	±	8.27	0.17	±	0.08	0.82	±	0.1
CoBoC KFKM	91.31	±	0.66	0.14	±	0.01	0.85	±	0.01
CoBoC SC	94.6	±	0	0.08	±	0	0.91	±	0

Tableau 4.12 — Évaluation externe de CoBoC complémentaire sur *Wine* selon différentes fusions finales pour les noyaux K_1 et K_2.

parkinson	% *F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC ALink	74.64	±	1.34	0.25	±	0	0.02	±	0.01
CoBoC CLink	74.64	±	1.34	0.25	±	0	0.02	±	0.01
CoBoC KKM	60.28	±	0.9	0.32	±	0.01	0.15	±	0.01
CoBoC KFKM	60.28	±	0.9	0.32	±	0.01	0.15	±	0.01
CoBoC SC	60.28	±	0.9	0.32	±	0.01	0.15	±	0.01
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	74.06	±	2.68	0.31	±	0.1	0.04	±	0.05
CoBoC ALink	70.82	±	4.51	0.25	±	0.01	0.05	±	0.05
CoBoC CLink	71.29	±	4.65	0.25	±	0	0.05	±	0.05
CoBoC KKM	66.32	±	6.84	0.31	±	0.07	0.1	±	0.07
CoBoC KFKM	64.94	±	5.14	0.29	±	0.05	0.09	±	0.06
CoBoC SC	66.96	±	5.24	0.32	±	0.06	0.06	±	0.06
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	76.05	±	0.28	0.47	±	0.07	0.01	±	0
CoBoC ALink	66.82	±	1.19	0.25	±	0	0.08	±	0.01
CoBoC CLink	67.3	±	2.64	0.25	±	0	0.08	±	0.02
CoBoC KKM	61.49	±	0.19	0.27	±	0.02	0.16	±	0.04
CoBoC KFKM	61.46	±	0.19	0.27	±	0.02	0.17	±	0.04
CoBoC SC	62.88	±	4.73	0.31	±	0.06	0.18	±	0.06
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	75.31	±	0.28	0.27	±	0.07	0.02	±	0
CoBoC ALink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC CLink	71.63	±	1.2	0.25	±	0	0.05	±	0.01
CoBoC KKM	61.38	±	3.26	0.31	±	0.03	0.14	±	0.03
CoBoC KFKM	60.28	±	0.9	0.32	±	0.01	0.15	±	0.01
CoBoC SC	60.28	±	0.9	0.32	±	0.01	0.15	±	0.01
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	74.2	±	1.54	0.27	±	0.07	0.02	±	0.01
CoBoC ALink	71.91	±	4.68	0.27	±	0.07	0.04	±	0.04
CoBoC CLink	71.51	±	2.24	0.25	±	0	0.04	±	0.02
CoBoC KKM	63.14	±	4.79	0.27	±	0.02	0.11	±	0.06
CoBoC KFKM	60.77	±	4.3	0.32	±	0.06	0.09	±	0.07
CoBoC SC	65.78	±	5.46	0.34	±	0.08	0.07	±	0.05
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC ALink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC CLink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC KKM	61.49	±	0.19	0.27	±	0.02	0.16	±	0.04
CoBoC KFKM	61.3	±	0	0.29	±	0	0.2	±	0
CoBoC SC	61.3	±	0	0.29	±	0	0.2	±	0

Tableau 4.13 — Évaluation externe de CoBoC consensus sur *parkinson* selon différentes fusions finales pour les noyaux K_1 et K_2.

parkinson	% *F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	74.38	±	1.69	0.25	±	0	0.02	±	0.01
CoBoC ALink	74.24	±	1.67	0.25	±	0	0.03	±	0.01
CoBoC CLink	74.24	±	1.67	0.25	±	0	0.03	±	0.01
CoBoC KKM	62.15	±	4.38	0.3	±	0.03	0.13	±	0.04
CoBoC KFKM	61.1	±	3.31	0.31	±	0.03	0.14	±	0.04
CoBoC SC	60.02	±	0.71	0.32	±	0.02	0.15	±	0.02
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	71.38	±	5.39	0.28	±	0.07	0.07	±	0.09
CoBoC ALink	67.94	±	5.88	0.26	±	0.02	0.09	±	0.07
CoBoC CLink	70.88	±	2.48	0.25	±	0	0.05	±	0.02
CoBoC KKM	62.58	±	5.26	0.27	±	0.02	0.15	±	0.07
CoBoC KFKM	60.76	±	3.71	0.28	±	0.02	0.17	±	0.05
CoBoC SC	61.68	±	3.16	0.31	±	0.02	0.14	±	0.05
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	76.05	±	0.28	0.47	±	0.07	0.01	±	0
CoBoC ALink	66.82	±	1.19	0.25	±	0	0.08	±	0.01
CoBoC CLink	67.3	±	2.64	0.25	±	0	0.08	±	0.02
CoBoC KKM	61.54	±	0.36	0.27	±	0.02	0.17	±	0.04
CoBoC KFKM	61.38	±	0.15	0.28	±	0.02	0.19	±	0.03
CoBoC SC	62.88	±	4.73	0.31	±	0.06	0.18	±	0.06
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	75.41	±	0.37	0.3	±	0.1	0.02	±	0
CoBoC ALink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC CLink	71.27	±	0.13	0.25	±	0	0.05	±	0
CoBoC KKM	61.08	±	3.32	0.31	±	0.03	0.13	±	0.03
CoBoC KFKM	60.02	±	0.71	0.32	±	0.02	0.15	±	0.02
CoBoC SC	60.02	±	0.71	0.32	±	0.02	0.15	±	0.02
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	72.77	±	3.37	0.28	±	0.08	0.04	±	0.04
CoBoC ALink	71.58	±	4.43	0.25	±	0.01	0.04	±	0.03
CoBoC CLink	72	±	4.76	0.25	±	0	0.04	±	0.04
CoBoC KKM	61.87	±	3.69	0.28	±	0.03	0.16	±	0.06
CoBoC KFKM	60.42	±	1.73	0.29	±	0.03	0.18	±	0.06
CoBoC SC	65.58	±	8.48	0.36	±	0.09	0.09	±	0.08
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC ALink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC CLink	75.22	±	0	0.25	±	0	0.02	±	0
CoBoC KKM	61.49	±	0.19	0.27	±	0.02	0.16	±	0.04
CoBoC KFKM	61.3	±	0	0.29	±	0	0.2	±	0
CoBoC SC	61.3	±	0	0.29	±	0	0.2	±	0

Tableau 4.14 — Évaluation externe de CoBoC complémentaire sur *parkinson* selon différentes fusions finales pour les noyaux K_1 et K_2.

Étude de la fusion finale par approche multi-vues. Les approches heuristiques de CoBoC ont également été étudiées en prémisse à l'application d'une approche multi-vues : ici, CoFKM ou CoKFKM. L'idée est de se servir des dernières représentations optimales du jeu de donnée, apprises par l'application de CoBoC, et de construire des données multi-vues pour CoFKM et CoKFKM. Soit $\{X^{*(r)}\}_{r\in[1..n_r]}$ l'ensemble des représentations optimales obtenues par $\{A^{(r)}\}_{r\in[1..n_r]}$:

- CoBoC consensus CoFKM et CoBoC complémentaire CoFKM sont appliqués sur le jeu de donnée multi-vues $\mathcal{X}$ représenté par $\{X^{*(r)}\}_{r\in[1..n_r]}$;
- CoBoC consensus CoKFKM et CoBoC complémentaire CoKFKM sont appliqués sur le jeu de donnée multi-vues $\mathcal{X}$ représenté par $\{K^{(r)}\}_{r\in[1..n_r]}$ où $K^{(r)}$ est défini par :

$$K^{(r)} = \frac{1}{Z} X^{*(r)} X^{*(r)\top}$$

avec $Z = \underset{(x_i,x_j)\in\mathcal{X}^2}{max} \langle x_i, x_j \rangle$

Chaque $K^{(r)}$ est alors une matrice des produits scalaires normalisés entre individus.

L'approche CoKFKM donne de meilleurs résultats que CoFKM après application de CoBoC. Les performances ne parviennent sur *Wine* qu'à égaler celles de CoFKM appliqué sur les données classiques (Tab. 4.8). Concernant les jeux *Iris* et *Parkinson*, la stratégie Γ_{Random} permet d'atteindre des solutions de meilleure qualité (Tab. 4.15) (pour la *F-mesure* concernant *Parkinson*, Tab. (4.17))

Iris	*% F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC consensus CoFKM	72.57	±	4.2	0.25	±	0.04	0.63	±	0.05
CoBoC consensus CoKFKM	76.8	±	6.73	0.26	±	0.06	0.69	±	0.07
CoBoC complement CoFKM	73.06	±	1.51	0.28	±	0.05	0.63	±	0.02
CoBoC complement CoKFKM	74.35	±	0.57	0.26	±	0.03	0.66	±	0.01
Stratégie Γ_{Min}									
CoBoC consensus CoFKM	66.03	±	8	0.39	±	0.14	0.52	±	0.12
CoBoC consensus CoKFKM	67.25	±	8.87	0.35	±	0.15	0.55	±	0.13
CoBoC complement CoFKM	70.62	±	3.39	0.24	±	0.05	0.61	±	0.05
CoBoC complement CoKFKM	71.51	±	2.46	0.25	±	0.05	0.62	±	0.03
Stratégie Γ_{Max}									
CoBoC consensus CoFKM	67.64	±	0.84	0.27	±	0.01	0.58	±	0.01
CoBoC consensus CoKFKM	74.03	±	2.15	0.25	±	0.03	0.66	±	0.02
CoBoC complement CoFKM	67.41	±	0.45	0.27	±	0.01	0.57	±	0
CoBoC complement CoKFKM	73.04	±	3.52	0.26	±	0.05	0.65	±	0.03

TABLEAU 4.15 — Évaluation externe de CoBoC sur *Iris* selon différentes fusions finales multi-vues.

Wine	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC consensus CoFKM	83.55	±	5.18	0.26	±	0.05	0.74	±	0.07
CoBoC consensus CoKFKM	82.91	±	3.8	0.25	±	0.04	0.74	±	0.04
CoBoC complement CoFKM	84.72	±	4.62	0.24	±	0.05	0.75	±	0.06
CoBoC complement CoKFKM	84.92	±	2.16	0.21	±	0.02	0.78	±	0.02
Stratégie Γ_{Min}									
CoBoC consensus CoFKM	76.63	±	4.21	0.34	±	0.04	0.63	±	0.05
CoBoC consensus CoKFKM	85.84	±	4.88	0.24	±	0.09	0.76	±	0.08
CoBoC complement CoFKM	86.8	±	2.83	0.23	±	0.04	0.78	±	0.04
CoBoC complement CoKFKM	85.85	±	2.78	0.22	±	0.04	0.76	±	0.04
Stratégie Γ_{Max}									
CoBoC consensus CoFKM	75.16	±	16.48	0.37	±	0.25	0.61	±	0.23
CoBoC consensus CoKFKM	93.19	±	0	0.13	±	0	0.88	±	0
CoBoC complement CoFKM	79.84	±	9.11	0.28	±	0.09	0.67	±	0.13
CoBoC complement CoKFKM	93.19	±	0	0.13	±	0	0.88	±	0

TABLEAU 4.16 — Évaluation externe de CoBoC sur *Wine* selon différentes fusions finales multi-vues.

parkinson	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC consensus CoFKM	56.96	±	1.34	0.37	±	0.06	0.03	±	0.03
CoBoC consensus CoKFKM	68.85	±	1.51	0.25	±	0	0.07	±	0.01
CoBoC complement CoFKM	57.09	±	1.47	0.33	±	0.03	0.04	±	0.06
CoBoC complement CoKFKM	69.03	±	0.87	0.25	±	0	0.06	±	0.01
Stratégie Γ_{Min}									
CoBoC consensus CoFKM	60.19	±	3.74	0.34	±	0.03	0.11	±	0.07
CoBoC consensus CoKFKM	63.43	±	4.09	0.25	±	0.01	0.12	±	0.04
CoBoC complement CoFKM	60.88	±	2.02	0.34	±	0.02	0.11	±	0.04
CoBoC complement CoKFKM	59.5	±	0.8	0.25	±	0.01	0.23	±	0.03
Stratégie Γ_{Max}									
CoBoC consensus CoFKM	56.14	±	0.8	0.36	±	0.01	0.01	±	0.01
CoBoC consensus CoKFKM	61.85	±	0.51	0.25	±	0	0.12	±	0
CoBoC complement CoFKM	55.59	±	0.28	0.37	±	0.04	0	±	0
CoBoC complement CoKFKM	61.85	±	0.51	0.25	±	0	0.12	±	0

TABLEAU 4.17 — Évaluation externe de CoBoC sur *parkinson* selon différentes fusions finales multi-vues.

Évaluation externe de CoBoC pour le *clustering* multi-vues

Les heuristiques CoBoC consensus et CoBoC complémentaire et les stratégies associées (Γ_{Random}, Γ_{Min} et Γ_{Max}) ont également été observées sur les données *mfeat*. Le contexte est celui du *clustering* multi-vues, où l'on cherche un *clustering* particulier réalisant un consensus en exploitant les descriptions de données multi-vues, décrites par plusieurs groupes de variables.

Apport de la collaboration à chaque algorithme. Le tableau 4.18 montre les résultats obtenus sur le jeu de donnée *mfeat* pour lesquels on applique les algorithmes localement sans collaboration. L'objectif est d'observer les performances des différentes approches de recherche de consensus par CoBoC relativement à ces résultats.

	% *F-mesure*			*AvgEnt*			*NMI*		
mfeat : Algorithmes locaux différents									
KM vue 0	59.37	±	4.61	0.73	±	0.1	0.68	±	0.03
FKM vue 1	33.29	±	1.07	1.76	±	0.08	0.4	±	0.02
SC vue 2	61.93	±	0.85	0.69	±	0.04	0.7	±	0
SLink vue 3	50.46	±	0	1.05	±	0	0.66	±	0
ALink vue 4	39.96	±	0	1.24	±	0	0.54	±	0
CLink vue 5	26.06	±	0	1.87	±	0	0.36	±	0
mfeat : Algorithmes FKM locaux									
FKM vue 0	63.31	±	3.38	0.67	±	0.03	0.7	±	0.02
FKM vue 1	33.89	±	0.4	1.72	±	0.03	0.41	±	0.01
FKM vue 2	21.94	±	0.09	2.53	±	0.02	0.14	±	0
FKM vue 3	56.65	±	2.83	0.8	±	0.06	0.68	±	0.01
FKM vue 4	72.59	±	5.53	0.48	±	0.06	0.77	±	0.04
FKM vue 5	39.53	±	0.19	1.32	±	0.01	0.48	±	0

Tableau 4.18 — Évaluation externe de CoBoC consensus sur *mfeat* selon les résultats locaux.

Dans un premier temps est observé avant fusion l'impact dans chaque vue (ou pour chaque algorithme) du processus de collaboration de CoBoC (Tab. 4.19 à Tab. 4.22) par rapport aux algorithmes appliqués sur chaque vue sans collaboration (Tab. 4.18). Comme dans le contexte de la combinaison de modèle, on n'observe pas de tendance générale d'amélioration de tous les algorithmes de *clustering* locaux. Cependant, on peut observer un rétrécissement de l'écart de performance entre les différents algorithmes. En particulier, la qualité des algorithmes les plus performants est souvent réduite au profit de l'amélioration des algorithmes les moins performants. Par exemple, l'algorithme de *clustering* spectral SC de la vue 2 de qualité maximale dans (Tab. 4.19) voit sa qualité réduite après application de CoBoC avec la stratégie Γ_{Random} (selon la *F-mesure*, de 61.93 à 59.59) là où l'algorithme CLink de la vue 5 voit sa performance augmenter (selon la *F-mesure*, de 26.06 à 42.52). Le même genre d'observation peut être fait sur les autres tableaux de résultats (Tab. 4.20 à Tab. 4.22). En particulier, dans les deux derniers tableaux, les algorithmes employés localement sont les mêmes, la différence entre les performances de ceux-ci sont donc directement dérivées des différentes représentations de $\mathcal{X}$. Les observations décrites précédemment traduisent ici la recherche de collaboration entre les vues des données pour atteindre un consensus, ce qui est l'objectif du *clustering* multi-vues.

mfeat	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	61.63	±	4.38	0.71	±	0.11	0.7	±	0.03
CoBoC vue 1	43.04	±	1.5	1.37	±	0.13	0.5	±	0.02
CoBoC vue 2	59.59	±	2.65	0.69	±	0.02	0.67	±	0.02
CoBoC vue 3	46.01	±	3.28	1.22	±	0.05	0.62	±	0.04
CoBoC vue 4	52.48	±	4.09	0.87	±	0.12	0.66	±	0.04
CoBoC vue 5	42.52	±	1.55	1.23	±	0.08	0.53	±	0.02
Stratégie Γ_{Min}									
CoBoC vue 0	52.48	±	3.26	0.92	±	0.07	0.6	±	0.03
CoBoC vue 1	27.25	±	1.24	2.12	±	0.09	0.27	±	0.02
CoBoC vue 2	40.63	±	3.58	1.27	±	0.11	0.47	±	0.03
CoBoC vue 3	34.82	±	4.92	1.61	±	0.2	0.45	±	0.07
CoBoC vue 4	39.21	±	1.68	1.31	±	0.1	0.53	±	0.02
CoBoC vue 5	32.62	±	3.84	1.58	±	0.1	0.4	±	0.04
Stratégie Γ_{Max}									
CoBoC vue 0	59.37	±	4.61	0.73	±	0.1	0.68	±	0.03
CoBoC vue 1	33.42	±	0.85	1.73	±	0.02	0.4	±	0.01
CoBoC vue 2	62.24	±	0.05	0.67	±	0	0.7	±	0
CoBoC vue 3	54.07	±	0	0.97	±	0	0.67	±	0
CoBoC vue 4	39.96	±	0	1.24	±	0	0.54	±	0
CoBoC vue 5	26.06	±	0	1.87	±	0	0.36	±	0

Tableau 4.19 — Évaluation externe de CoBoC consensus sur *mfeat* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC entre plusieurs algorithmes FKM.

mfeat	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	52.16	±	5.83	0.94	±	0.17	0.61	±	0.05
CoBoC vue 1	29.28	±	1.72	2.01	±	0.13	0.33	±	0.03
CoBoC vue 2	49.13	±	4.21	1	±	0.12	0.58	±	0.04
CoBoC vue 3	37.48	±	5.2	1.46	±	0.2	0.51	±	0.08
CoBoC vue 4	40.6	±	4.54	1.25	±	0.23	0.53	±	0.05
CoBoC vue 5	37.24	±	2.84	1.35	±	0.11	0.47	±	0.03
Stratégie Γ_{Min}									
CoBoC vue 0	45.15	±	2.91	1.13	±	0.12	0.53	±	0.03
CoBoC vue 1	21.66	±	2.47	2.41	±	0.07	0.16	±	0.05
CoBoC vue 2	39.44	±	1.49	1.32	±	0.09	0.48	±	0.02
CoBoC vue 3	30.81	±	5.44	1.8	±	0.25	0.39	±	0.08
CoBoC vue 4	38.25	±	6.79	1.32	±	0.24	0.51	±	0.06
CoBoC vue 5	27.53	±	3.09	1.73	±	0.09	0.36	±	0.03
Stratégie Γ_{Max}									
CoBoC vue 0	61.32	±	2.35	0.69	±	0.09	0.7	±	0.02
CoBoC vue 1	42.74	±	1.18	1.37	±	0.05	0.51	±	0.01
CoBoC vue 2	62.95	±	1.36	0.66	±	0.06	0.72	±	0.01
CoBoC vue 3	48.67	±	3.61	1.07	±	0.08	0.6	±	0.04
CoBoC vue 4	45.13	±	3.45	0.97	±	0.12	0.61	±	0.03
CoBoC vue 5	35.66	±	4.16	1.49	±	0.16	0.45	±	0.03

Tableau 4.20 — Évaluation externe de CoBoC complémentaire sur *mfeat* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC entre plusieurs algorithmes FKM.

mfeat	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	67.8	±	5.51	0.57	±	0.07	0.74	±	0.03
CoBoC vue 1	42.96	±	0.82	1.36	±	0.08	0.5	±	0.01
CoBoC vue 2	53.03	±	5.26	0.98	±	0.2	0.61	±	0.05
CoBoC vue 3	51.56	±	4.12	0.89	±	0.11	0.64	±	0.03
CoBoC vue 4	64.2	±	3.74	0.64	±	0.04	0.7	±	0.02
CoBoC vue 5	50.3	±	2.57	1.01	±	0.08	0.57	±	0.02
Stratégie Γ_{Min}									
CoBoC vue 0	57.21	±	4.67	0.79	±	0.09	0.63	±	0.04
CoBoC vue 1	29.24	±	3.27	2.04	±	0.15	0.31	±	0.06
CoBoC vue 2	26.55	±	2.04	2.21	±	0.16	0.26	±	0.05
CoBoC vue 3	49.49	±	6.08	1.02	±	0.21	0.59	±	0.05
CoBoC vue 4	39.99	±	2.97	1.4	±	0.1	0.46	±	0.04
CoBoC vue 5	38.64	±	4.79	1.38	±	0.19	0.46	±	0.06
Stratégie Γ_{Max}									
CoBoC vue 0	63.31	±	3.38	0.67	±	0.03	0.7	±	0.02
CoBoC vue 1	33.89	±	0.38	1.73	±	0.03	0.41	±	0.01
CoBoC vue 2	21.84	±	0.1	2.55	±	0.04	0.14	±	0.01
CoBoC vue 3	56.65	±	2.83	0.8	±	0.06	0.68	±	0.01
CoBoC vue 4	72.46	±	5.55	0.48	±	0.06	0.77	±	0.04
CoBoC vue 5	39.53	±	0.19	1.32	±	0.01	0.48	±	0

Tableau 4.21 — Évaluation externe de CoBoC consensus sur *mfeat* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC entre plusieurs algorithmes différents.

mfeat	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC vue 0	52.5	±	3.22	0.98	±	0.11	0.6	±	0.03
CoBoC vue 1	28.4	±	1.92	2	±	0.18	0.31	±	0.04
CoBoC vue 2	27.68	±	1.78	2.13	±	0.18	0.3	±	0.04
CoBoC vue 3	47.78	±	5.06	1	±	0.16	0.58	±	0.05
CoBoC vue 4	40.08	±	6.27	1.38	±	0.25	0.46	±	0.06
CoBoC vue 5	46.21	±	4.58	1.15	±	0.15	0.53	±	0.04
Stratégie Γ_{Min}									
CoBoC vue 0	44.1	±	4.04	1.19	±	0.14	0.5	±	0.04
CoBoC vue 1	24.71	±	0.56	2.28	±	0.11	0.23	±	0.02
CoBoC vue 2	25.34	±	2.58	2.25	±	0.17	0.24	±	0.05
CoBoC vue 3	35.81	±	6.86	1.39	±	0.25	0.45	±	0.08
CoBoC vue 4	38.19	±	0.96	1.45	±	0.09	0.45	±	0.01
CoBoC vue 5	36.47	±	2.51	1.47	±	0.1	0.42	±	0.03
Stratégie Γ_{Max}									
CoBoC vue 0	64.01	±	0.41	0.63	±	0.03	0.71	±	0.01
CoBoC vue 1	43.35	±	0.84	1.4	±	0.02	0.51	±	0.01
CoBoC vue 2	30.52	±	3.24	1.84	±	0.17	0.34	±	0.06
CoBoC vue 3	56.29	±	3.6	0.76	±	0.06	0.68	±	0.03
CoBoC vue 4	73.89	±	6.39	0.45	±	0.06	0.78	±	0.04
CoBoC vue 5	48	±	1.8	1.07	±	0.05	0.55	±	0.02

Tableau 4.22 — Évaluation externe de CoBoC complémentaire sur *mfeat* selon les résultats locaux. Chaque *clustering* local est un consensus issu du processus de collaboration de CoBoC entre plusieurs algorithmes différents.

Apport de la fusion finale par le noyau K_1 et K_2. Le tableau 4.23 rappelle les résultats obtenus sur *mfeat* par l'approche CoFKM dans ses trois variantes (cf. section 2.4.2). Ces résultats permettent d'observer l'apport éventuel de CoBoC pour la recherche d'un *clustering* multi-vues.

	% *F-mesure*	*AvgEnt*	*NMI*
Approche multi-vues CoFKM			
CoFKM post	47.49 ± 5.3	0.94 ± 0.13	0.61 ± 0.04
CoFKM	92.86 ± 0.18	0.16 ± 0	0.93 ± 0
CoFKM concat	90.37 ± 3.7	0.19 ± 0.04	0.92 ± 0.02

Tableau 4.23 — Évaluation externe de CoFKM sur *mfeat*.

Les tableaux 4.24 à 4.27 permettent de mesurer l'apport de la fusion finale par les noyaux K_1 et K_2 permettant d'obtenir une solution au problème du *clustering* multi-vues posé par le jeu de donnée *mfeat*. Dans tous les cas, il n'est pas possible d'atteindre, selon le paramétrage des heuristiques et des stratégies, les performances obtenues par CoFKM, même lorsque pour CoBoC, dans chaque vue est appliqué un FKM. En revanche, l'objectif d'atteindre une solution consensus de meilleure qualité que les différents algorithmes de base employés est réalisé. On peut l'observer en croisant par exemple les tableaux 4.24 ou 4.27 et le tableau 4.18. Les stratégies Γ_{Random} et Γ_{Max} permettent une nette amélioration. En revanche, la stratégie Γ_{Min} ne trouve pas de solution consensus satisfaisante.

mfeat	% *F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	48.2	±	4.9	1.04	±	0.18	0.63	±	0.04
CoBoC ALink	63.21	±	3.04	0.66	±	0.08	0.75	±	0.03
CoBoC CLink	46.73	±	7.24	1	±	0.22	0.64	±	0.05
CoBoC KKM	67.21	±	5.8	0.56	±	0.12	0.76	±	0.03
CoBoC KFKM	76.82	±	4.55	0.42	±	0.06	0.81	±	0.03
CoBoC SC	74.85	±	3.15	0.42	±	0.03	0.81	±	0.01
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	35.51	±	5.02	1.54	±	0.15	0.45	±	0.05
CoBoC ALink	50.8	±	2.19	0.94	±	0.17	0.62	±	0.03
CoBoC CLink	27.77	±	3.09	1.66	±	0.12	0.43	±	0.05
CoBoC KKM	56.81	±	3.3	0.82	±	0.18	0.65	±	0.04
CoBoC KFKM	67.68	±	5.74	0.62	±	0.11	0.72	±	0.05
CoBoC SC	66.28	±	2.61	0.61	±	0.06	0.72	±	0.03
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	43.32	±	4.05	1.14	±	0.15	0.59	±	0.03
CoBoC ALink	62.23	±	4.51	0.62	±	0.04	0.73	±	0.03
CoBoC CLink	33.58	±	2.89	1.39	±	0.12	0.54	±	0.03
CoBoC KKM	68.79	±	3.22	0.52	±	0.07	0.77	±	0.03
CoBoC KFKM	77.62	±	2.39	0.38	±	0.05	0.82	±	0.02
CoBoC SC	76.23	±	3.18	0.38	±	0.04	0.82	±	0.02
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	28.47	±	11.56	1.83	±	0.62	0.44	±	0.13
CoBoC ALink	31.2	±	14.48	1.45	±	0.43	0.48	±	0.13
CoBoC CLink	32.7	±	14.85	1.41	±	0.39	0.51	±	0.12
CoBoC KKM	71.87	±	3.6	0.48	±	0.07	0.8	±	0.02
CoBoC KFKM	55.62	±	14.47	0.97	±	0.38	0.64	±	0.13
CoBoC SC	79.22	±	3.05	0.34	±	0.02	0.84	±	0.02
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	29.56	±	9.82	1.5	±	0.31	0.45	±	0.07
CoBoC ALink	30.51	±	12.73	1.52	±	0.31	0.46	±	0.11
CoBoC CLink	30.77	±	12.31	1.46	±	0.3	0.48	±	0.08
CoBoC KKM	64.33	±	6.1	0.6	±	0.14	0.73	±	0.05
CoBoC KFKM	25.36	±	3.07	2.06	±	0.16	0.27	±	0.04
CoBoC SC	75.12	±	2.09	0.45	±	0.08	0.81	±	0.02
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	45.39	±	16.49	1.04	±	0.46	0.61	±	0.13
CoBoC ALink	45.26	±	17.89	1.09	±	0.48	0.6	±	0.16
CoBoC CLink	46.1	±	17.84	0.99	±	0.5	0.61	±	0.14
CoBoC KKM	74.5	±	6.43	0.43	±	0.14	0.83	±	0.03
CoBoC KFKM	32.21	±	5.75	1.77	±	0.28	0.38	±	0.09
CoBoC SC	77	±	2.53	0.39	±	0.07	0.84	±	0.01

Tableau 4.24 — Évaluation externe de CoBoC consensus avec plusieurs algorithmes différents sur *mfeat* selon différentes fusions finales pour les noyaux K_1 et K_2.

mfeat	% *F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	39.38	±	1.91	1.35	±	0.14	0.52	±	0.04
CoBoC ALink	54.66	±	5.55	0.81	±	0.12	0.66	±	0.05
CoBoC CLink	32.89	±	6.1	1.44	±	0.26	0.5	±	0.06
CoBoC KKM	61.03	±	7.15	0.64	±	0.11	0.7	±	0.05
CoBoC KFKM	66.83	±	6.22	0.6	±	0.09	0.72	±	0.05
CoBoC SC	68.42	±	2.7	0.56	±	0.07	0.75	±	0.02
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	28.8	±	3.98	1.82	±	0.17	0.36	±	0.06
CoBoC ALink	44.95	±	6.02	1.1	±	0.1	0.56	±	0.06
CoBoC CLink	25	±	4.03	1.79	±	0.17	0.39	±	0.05
CoBoC KKM	51.43	±	6.88	0.94	±	0.16	0.6	±	0.06
CoBoC KFKM	52.35	±	9.79	1.02	±	0.31	0.6	±	0.09
CoBoC SC	58.43	±	7.71	0.84	±	0.2	0.65	±	0.07
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	53.03	±	4.97	0.89	±	0.17	0.65	±	0.03
CoBoC ALink	63.46	±	4.61	0.62	±	0.12	0.75	±	0.03
CoBoC CLink	46.01	±	8.05	1.07	±	0.28	0.63	±	0.05
CoBoC KKM	69.87	±	2.32	0.48	±	0.06	0.78	±	0.01
CoBoC KFKM	77.37	±	3.86	0.38	±	0.04	0.82	±	0.02
CoBoC SC	74.42	±	3.06	0.4	±	0.04	0.81	±	0.02
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	23.95	±	1.75	1.83	±	0.36	0.4	±	0.05
CoBoC ALink	23.47	±	0.44	1.68	±	0.03	0.4	±	0.01
CoBoC CLink	24.71	±	0.25	1.6	±	0.01	0.45	±	0.01
CoBoC KKM	63.68	±	3.99	0.61	±	0.03	0.74	±	0.02
CoBoC KFKM	27.16	±	5.09	2	±	0.2	0.3	±	0.09
CoBoC SC	77.75	±	5.36	0.41	±	0.08	0.83	±	0.03
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	26.24	±	5.27	1.8	±	0.41	0.41	±	0.06
CoBoC ALink	30.04	±	11.81	1.5	±	0.26	0.47	±	0.1
CoBoC CLink	29.36	±	9.49	1.49	±	0.23	0.47	±	0.07
CoBoC KKM	60.95	±	7.1	0.69	±	0.19	0.7	±	0.06
CoBoC KFKM	22.28	±	1.93	2.18	±	0.07	0.21	±	0.03
CoBoC SC	70.85	±	4.14	0.49	±	0.07	0.77	±	0.03
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	38.11	±	16.93	1.47	±	0.64	0.54	±	0.17
CoBoC ALink	40.59	±	19.76	1.24	±	0.5	0.56	±	0.17
CoBoC CLink	37.86	±	15.56	1.23	±	0.45	0.55	±	0.12
CoBoC KKM	71.86	±	4.75	0.44	±	0.1	0.8	±	0.03
CoBoC KFKM	36.15	±	6.56	1.52	±	0.28	0.44	±	0.09
CoBoC SC	79.09	±	2.76	0.34	±	0.08	0.84	±	0.02

Tableau 4.25 — Évaluation externe de CoBoC complémentaire avec plusieurs algorithmes différents sur *mfeat* selon différentes fusions finales pour les noyaux K_1 et K_2.

mfeat	% *F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	52.69	±	4.6	0.96	±	0.2	0.65	±	0.05
CoBoC ALink	64.36	±	3.37	0.65	±	0.04	0.74	±	0.03
CoBoC CLink	35.02	±	5.67	1.41	±	0.16	0.55	±	0.07
CoBoC KKM	66.76	±	7.95	0.61	±	0.15	0.75	±	0.05
CoBoC KFKM	74.24	±	5.95	0.46	±	0.07	0.79	±	0.03
CoBoC SC	77.81	±	4.23	0.38	±	0.04	0.82	±	0.02
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	36.04	±	3.7	1.5	±	0.12	0.45	±	0.04
CoBoC ALink	52.73	±	5.03	0.93	±	0.16	0.61	±	0.05
CoBoC CLink	29.08	±	5.58	1.62	±	0.18	0.43	±	0.05
CoBoC KKM	57.37	±	6.26	0.78	±	0.11	0.65	±	0.06
CoBoC KFKM	64.66	±	7	0.68	±	0.12	0.69	±	0.06
CoBoC SC	63.9	±	3.62	0.65	±	0.07	0.7	±	0.03
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	42.15	±	4.49	1.16	±	0.15	0.59	±	0.04
CoBoC ALink	69.89	±	7.9	0.56	±	0.16	0.77	±	0.05
CoBoC CLink	40.97	±	5.96	1.27	±	0.18	0.59	±	0.05
CoBoC KKM	72.51	±	5.37	0.49	±	0.09	0.79	±	0.03
CoBoC KFKM	79.56	±	2.42	0.39	±	0.05	0.83	±	0.01
CoBoC SC	80.27	±	3.3	0.35	±	0.04	0.84	±	0.02
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	29.57	±	13.84	2	±	0.76	0.41	±	0.16
CoBoC ALink	32.72	±	16.9	1.46	±	0.5	0.48	±	0.15
CoBoC CLink	33.52	±	17.03	1.4	±	0.42	0.51	±	0.13
CoBoC KKM	68.45	±	5.12	0.54	±	0.06	0.78	±	0.02
CoBoC KFKM	43.48	±	15.17	1.33	±	0.43	0.5	±	0.17
CoBoC SC	81.81	±	2.57	0.31	±	0.02	0.86	±	0.01
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	42.22	±	6.71	1.24	±	0.23	0.56	±	0.06
CoBoC ALink	58.03	±	5.41	0.76	±	0.16	0.69	±	0.05
CoBoC CLink	55.94	±	6.19	0.71	±	0.13	0.66	±	0.06
CoBoC KKM	65.03	±	7.86	0.63	±	0.16	0.73	±	0.07
CoBoC KFKM	28.07	±	4.04	2	±	0.25	0.33	±	0.08
CoBoC SC	72.59	±	3.49	0.48	±	0.09	0.79	±	0.03
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	44.3	±	15.27	1.08	±	0.43	0.61	±	0.12
CoBoC ALink	49.56	±	20.64	1	±	0.53	0.63	±	0.17
CoBoC CLink	47.92	±	19.89	0.99	±	0.51	0.62	±	0.15
CoBoC KKM	68.08	±	5.17	0.53	±	0.05	0.78	±	0.04
CoBoC KFKM	30.9	±	3.23	1.98	±	0.15	0.38	±	0.07
CoBoC SC	76.6	±	1.08	0.35	±	0.02	0.84	±	0.01

Tableau 4.26 — Évaluation externe de CoBoC consensus avec plusieurs algorithmes FKM sur *mfeat* selon différentes fusions finales pour les noyaux K_1 et K_2.

mfeat	% *F-mesure*			*AvgEnt*			*NMI*		
Similarité K_1 - Stratégie Γ_{Random}									
CoBoC SLink	32.3	±	1.81	1.64	±	0.21	0.43	±	0.04
CoBoC ALink	52.82	±	2.25	1	±	0.08	0.62	±	0.03
CoBoC CLink	30.89	±	4.76	1.6	±	0.16	0.45	±	0.04
CoBoC KKM	56	±	4.34	0.82	±	0.13	0.64	±	0.04
CoBoC KFKM	66.59	±	4.44	0.65	±	0.08	0.71	±	0.04
CoBoC SC	68.62	±	5.46	0.57	±	0.07	0.74	±	0.04
Similarité K_1 - Stratégie Γ_{Min}									
CoBoC SLink	28.2	±	1.93	1.82	±	0.08	0.36	±	0.02
CoBoC ALink	40.35	±	5.66	1.33	±	0.19	0.5	±	0.06
CoBoC CLink	21.66	±	0.89	2.03	±	0.1	0.31	±	0.02
CoBoC KKM	47.89	±	3.79	1.09	±	0.12	0.55	±	0.03
CoBoC KFKM	50.02	±	5.22	1.03	±	0.15	0.56	±	0.04
CoBoC SC	53.39	±	4.79	0.96	±	0.12	0.61	±	0.04
Similarité K_1 - Stratégie Γ_{Max}									
CoBoC SLink	49.97	±	7.04	1.08	±	0.2	0.65	±	0.06
CoBoC ALink	70.6	±	7.69	0.48	±	0.11	0.78	±	0.05
CoBoC CLink	36.78	±	8.74	1.29	±	0.29	0.58	±	0.08
CoBoC KKM	73.82	±	6.41	0.48	±	0.12	0.8	±	0.03
CoBoC KFKM	80.49	±	4.37	0.36	±	0.05	0.83	±	0.03
CoBoC SC	76.16	±	4.38	0.43	±	0.08	0.82	±	0.03
Similarité K_2 - Stratégie Γ_{Random}									
CoBoC SLink	29.72	±	9.4	1.48	±	0.29	0.47	±	0.06
CoBoC ALink	30.22	±	13.2	1.5	±	0.42	0.45	±	0.11
CoBoC CLink	29.68	±	10.07	1.47	±	0.27	0.48	±	0.07
CoBoC KKM	63.55	±	1.69	0.7	±	0.06	0.72	±	0.01
CoBoC KFKM	36.3	±	2.82	1.68	±	0.16	0.43	±	0.03
CoBoC SC	76.05	±	3.98	0.44	±	0.11	0.82	±	0.02
Similarité K_2 - Stratégie Γ_{Min}									
CoBoC SLink	33.24	±	8.51	1.64	±	0.46	0.47	±	0.09
CoBoC ALink	42.82	±	15.91	1.14	±	0.45	0.56	±	0.14
CoBoC CLink	41.77	±	14.23	1.18	±	0.37	0.55	±	0.11
CoBoC KKM	61.78	±	4.1	0.73	±	0.06	0.69	±	0.02
CoBoC KFKM	26.07	±	4.33	2.09	±	0.19	0.29	±	0.07
CoBoC SC	65.5	±	4.09	0.64	±	0.12	0.73	±	0.03
Similarité K_2 - Stratégie Γ_{Max}									
CoBoC SLink	25.69	±	0.33	1.61	±	0.01	0.45	±	0.01
CoBoC ALink	24.24	±	0.26	1.67	±	0.01	0.41	±	0.01
CoBoC CLink	25.19	±	0.47	1.6	±	0.01	0.45	±	0.01
CoBoC KKM	71.85	±	4.6	0.47	±	0.09	0.8	±	0.03
CoBoC KFKM	36.4	±	5.35	1.6	±	0.22	0.45	±	0.09
CoBoC SC	77.84	±	4.38	0.34	±	0.05	0.83	±	0.02

Tableau 4.27 — Évaluation externe de CoBoC complémentaire avec plusieurs algorithmes FKM sur *mfeat* selon différentes fusions finales pour les noyaux K_1 et K_2.

Étude de la fusion finale par approche multi-vues. Une dernière étude intéressante est d'observer l'apport de la recherche de solutions locales consensus par CoBoC pour le *clustering* multi-vues, notamment pour l'utilisation de CoFKM. CoBoC est utilisé ici pour l'apprentissage de représentations optimales locales, dont on espère qu'elles seront de suffisamment bonne qualité pour une recherche de consensus par CoFKM. Pour rappel, soit $\{X^{*(r)}\}_{r\in[1..n_r]}$ l'ensemble des représentations optimales obtenues par $\{A^{(r)}\}_{r\in[1..n_r]}$:

- CoBoC consensus CoFKM et CoBoC complémentaire CoFKM sont appliqués sur le jeu de donnée multi-vues $\mathcal{X}$ représenté par $\{X^{*(r)}\}_{r\in[1..n_r]}$;
- CoBoC consensus CoKFKM et CoBoC complémentaire CoKFKM sont appliqués sur le jeu de donnée multi-vues $\mathcal{X}$ représenté par $\{K^{(r)}\}_{r\in[1..n_r]}$ où $K^{(r)}$ est défini par :

$$K^{(r)} = \frac{1}{Z} X^{*(r)} X^{*(r)\top}$$

avec $Z = \underset{(x_i,x_j)\in\mathcal{X}^2}{max} \langle x_i, x_j \rangle$

Les meilleures performances de CoBoC sont atteintes par l'adjonction de CoKFKM comme procédure de fusion finale, et avec les noyaux $\{K^{(r)}\}_{r\in[1..n_r]}$. Le résultat fort ici est l'amélioration de CoFKM (tableau 4.29), déjà très performant sur *mfeat*, par CoKFKM à partir des noyaux issus de l'application de CoBoC complémentaire avec la stratégie Γ_{Max}.

	% F-mesure			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC consensus CoFKM	52.41	±	9.22	1.04	±	0.3	0.62	±	0.1
CoBoC consensus CoKFKM	84.72	±	7.45	0.28	±	0.09	0.87	±	0.05
CoBoC complement CoFKM	48.74	±	5.28	1.08	±	0.13	0.58	±	0.05
CoBoC complement CoKFKM	55.64	±	8.36	0.85	±	0.21	0.65	±	0.08
Stratégie Γ_{Min}									
CoBoC consensus CoFKM	41.09	±	6.65	1.34	±	0.31	0.5	±	0.1
CoBoC consensus CoKFKM	50.11	±	6.49	1.04	±	0.18	0.59	±	0.06
CoBoC complement CoFKM	41.78	±	7.21	1.41	±	0.32	0.48	±	0.1
CoBoC complement CoKFKM	35.61	±	2.03	1.63	±	0.09	0.43	±	0.03
Stratégie Γ_{Max}									
CoBoC consensus CoFKM	41.34	±	3.73	1.44	±	0.19	0.53	±	0.05
CoBoC consensus CoKFKM	91.4	±	0.14	0.19	±	0	0.92	±	0
CoBoC complement CoFKM	48.66	±	2.6	1.16	±	0.15	0.6	±	0.03
CoBoC complement CoKFKM	87.31	±	3.55	0.23	±	0.03	0.89	±	0.02

Tableau 4.28 — Évaluation externe de CoBoC avec plusieurs algorithmes différents sur *mfeat* selon différentes fusions finales multi-vues.

4.8 Discussion

La plateforme de *clustering* collaboratif proposée se décline en deux variantes heuristiques selon l'objectif de recherche d'un ou plusieurs *clusterings* consensus ou de plusieurs *clusterings* alternatifs. Celles-ci peuvent être appliquées dans différents contextes comme :

- la combinaison de modèles, où plusieurs algorithmes de *clustering* peuvent être employés pour fouiller un jeu de donnée classique mono-vue ;
- le multi-vues, où un ou plusieurs algorithmes peuvent être employés pour fouiller les parties communes ou différentes parmi des données multi-représentées.

	% *F-mesure*			*AvgEnt*			*NMI*		
Stratégie Γ_{Random}									
CoBoC consensus CoFKM	50.73	±	12.13	1.09	±	0.37	0.6	±	0.13
CoBoC consensus CoKFKM	83.93	±	5.22	0.29	±	0.07	0.87	±	0.03
CoBoC complement CoFKM	48.63	±	4.84	1.11	±	0.15	0.58	±	0.06
CoBoC complement CoKFKM	49.21	±	7.66	1.06	±	0.16	0.59	±	0.07
Stratégie Γ_{Min}									
CoBoC consensus CoFKM	40.68	±	9.79	1.42	±	0.37	0.48	±	0.11
CoBoC consensus CoKFKM	48.34	±	10.72	1.2	±	0.34	0.59	±	0.09
CoBoC complement CoFKM	37.18	±	5.59	1.55	±	0.22	0.43	±	0.08
CoBoC complement CoKFKM	32.49	±	1.83	1.79	±	0.07	0.4	±	0.03
Stratégie Γ_{Max}									
CoBoC consensus CoFKM	39.71	±	2.79	1.45	±	0.16	0.52	±	0.04
CoBoC consensus CoKFKM	91.47	±	0	0.19	±	0	0.92	±	0
CoBoC complement CoFKM	37.26	±	4.46	1.58	±	0.32	0.47	±	0.07
CoBoC complement CoKFKM	93.26	±	0.55	0.15	±	0.01	0.93	±	0.01

Tableau 4.29 — Évaluation externe de CoBoC avec plusieurs algorithmes FKM sur *mfeat* selon différentes fusions finales multi-vues.

La collaboration proposée pour atteindre l'accord (consensus) ou le désaccord (alternatives) entre les différents algorithmes employés est basé sur un mécanisme d'échange de contraintes permettant localement de trouver simultanément un *clustering* atteignant l'objectif et un sous-espace de représentation des données menant à ces *clusterings*. Cette dernière facette n'est pas présente dans les différentes approches étudiées dans l'état de l'art, et permet des analyses d'un autre ordre. Par exemple, une question à laquelle la contribution proposée peut répondre est la suivante :

quelles sont localement les sous-espaces de représentation qui permettent d'atteindre un consensus quelquesoient les algorithmes de clustering *locaux employés ?*

La résolution de cette question peut permettre, pour des données multi-vues, d'identifier les attributs créant du bruit pour l'obtention d'un *clustering* cible. Ceux-ci sont alors de faibles contributeurs à la définition du sous-espace permettant par exemple d'atteindre des solutions de *clutering* proches.

L'approche a été évaluée empiriquement afin d'observer son comportement de manière interne, et de manière externe. Ces expériences ont permis de dégager des liens comme par exemple, l'importance de chercher une solution consensus entre les algorithmes locaux lorsque ceux-ci proposent des solutions de *clustering* de base très diverses.

L'approche proposée a néanmoins le défaut d'être assez fortement paramétrée, notamment par le volume de contraintes échangées et le nombre d'échanges envisagé. Nous avons observé notamment que les solutions les plus intéressantes étaient obtenues lors des quelques premiers échanges. Une observation intéressante serait de conserver l'historique des solutions trouvées à chaque étape de génération des contraintes afin d'observer, par exemple dans le cas de la recherche de consensus, si la solution maximisant l'information mutuelle normalisée durant une exécution de l'algorithme permet d'atteindre une solution vraiment meilleure au sens de l'évaluation externe. La variante AlterBoC, elle, manque de procédure d'évaluation externe, mais ceci est normal par essence. En effet si un *clustering* de bonne qualité peut être obtenu sur des données, au sens de cette évaluation externe, alors une alternative sera de mauvaise

qualité au sens de l'évaluation choisie et sera donc peu valorisable. En revanche, les techniques de recherche d'alternatives trouvent tout à fait leur place lors de la confrontation à de réelles données dont on ne connait pas du tout la classification de départ, ou bien lorsque celle-ci est connu de l'analyste qui préfère alors découvrir quelque chose de *différent*.

4.9 Conclusion

Ce chapitre a permis d'introduire la plateforme collaborative proposée, dont CoBoC et AlterBoC sont des instances particulières. Il reprend de façon synthétique des développements réalisés dans le cadre du *clustering* d'ensemble qui mène au *clustering* collaboratif pour la recherche de consensus. Des développement récents, et des interrogations sur la diversification des problèmes autour du *clustering* [Kriegel and Zimek, 2010] ont guidé la recherche bibliographique autour notamment du *clustering* alternatif, et laisse entrevoir les liens entre toutes les problématiques, avec en suspens l'éventualité de voir des approches susceptibles de les unifier et de proposer un mécanisme de résolution adéquat. La plateforme proposée tend vers cet objectif de pouvoir gérer simultanément la recherche d'un ou plusieurs *clusterings*, consensus ou alternatifs, à travers un même mécanisme de collaborations entre plusieurs classifieurs non supervisés.

Les études expérimentales proposées suggèrent de nombreuses applications, mais celles-ci n'ont pu être réalisées afin de valoriser davantage les approches. La plateforme présentée est bien entendue extensible, et d'autres heuristiques peuvent être proposées pour atteindre les différents objectifs fixés. En particulier, en perspective de l'approche proposée, une amélioration serait, plutôt que de fixer la stratégie de génération des contraintes pour chaque algorithme de *clustering* local, de trouver un moyen de déterminer automatiquement quelles contraintes serait les plus judicieuses pour chacun.

Conclusion et perspectives

Conclusion

Ce travail de thèse a proposé une vision restreinte mais constructive, de l'évolution de la problématique classique du *clustering*, dans un premier temps vers l'adaptation à des problématiques applicatives de multiplicité de données, puis dans un second temps vers les problématiques de multiplicité des analyses et leur combinaison.

La première problématique abordée est la classification non supervisée multi-vues. Nous avons proposé pour résoudre ce problème, une approche centralisée collaborative et floue, ainsi qu'une extension à noyaux, permettant de traiter des données décrites simultanément par des représentations vectorielles et relationnelles. L'élaboration de cette contribution est permise grâce aux travaux de [Pedrycz, 2002] (CoFC) et [Bickel and Scheffer, 2005] (CoEM). Partant de l'approche multi-vues non convergente CoEM, nous avons proposé, sur la base d'une extension des K-moyennes floues ([Bezdek, 1981]) à la manière de CoFC, un critère simple et intuitif menant à un algorithme également simple, intuitif, et convergent. L'utilisation éventuelle de noyaux permet d'adapter l'algorithme pour des questions de complexité algorithmique. De plus, l'approche proposée généralise complètement diverses solutions de fusion naïves, basées sur FKM : la concaténation ou fusion *a priori*, où FKM est directement appliqué à la représentation jointe des différentes vues, et la fusion *a posteriori*, lorsque FKM est appliqué indépendamment sur chaque vue.

Le développement des approches centralisées dédiées aux données multi-vues reposent sur le paradigme de la recherche de *clusterings* adaptés dans chaque vue, mais liés entre eux par la réduction d'un critère de désaccord. Ce paradigme implique la construction de différents clusterings locaux devant tendre ensemble vers une solution consensus. La contribution proposée relevant d'une approche de *clustering* connue et paramétrée, les clusterings locaux peuvent alors être construits explicitement pour répondre au critère objectif posé, celui-ci étant *simple*. Cet aspect peut être considéré comme une première approche faisant intervenir la multiplicité des traitements, dans la mesure où les *clusterings* locaux optimaux minimisant le désaccord constituent un ensemble de *clusterings* consensus, émanant tous de la collaboration entre les vues. L'évolution naturelle envisagée pour nos contributions a alors été de proposer un modèle permettant de s'abstraire des algorithmes utilisés dans chaque vue ainsi que de leurs paramètres. L'instanciation d'un tel modèle peut alors permettre d'adapter le traitement réalisé dans chaque vue après connaissance de caractéristiques particulières sur ces vues (*e.g.* les types des descripteurs). Ce constat a donné lieu aux dernières approches proposées, se fondant complètement sur des principes tirés du *clustering* semi-supervisé, problématique qui a été étudié également dans cette thèse.

Le second apport proposé concerne alors l'intégration de connaissances externes en classification non supervisée. Dans ce contexte, la contribution est double puisque nous proposons une approche fondée sur le *boosting* dans un contexte non supervisé : BoC, et une approche fondée sur un algorithme d'optimisation numérique adapté : UzaBoC. En particulier,

nous montrons comment une variante de la seconde approche (ADAUZABOC) peut s'interpréter en terme de *boosting*. Ces contributions suivent directement les travaux de [Liu et al., 2007] (BOOSTCLUSTER) fonctionnant par génération successive de sous-espaces de représentation des données dans lequel un algorithme quelconque de *clustering* permettrait de mieux regrouper les individus, et en particulier, les individus pour lesquels des connaissances externes sont disponibles.

Nous proposons pour chaque contribution un formalisme adapté basé sur deux principes que sont la cohérence vis à vis de la représentation d'origine, et la consistance vis à vis des connaissances externes. Nous montrons en particulier que l'approche BOOSTCLUSTER optimise un critère proche du critère de consistance proposé dans le cadre de la contribution BOC. La seconde contribution UZABOC et sa variante ADAUZABOC permettent d'apprendre simultanément un *clustering* respectant au mieux les connaissances externes, et la fonction de distance permettant d'obtenir ce *clustering*. En particulier cette fonction de distance est obtenue comme l'optimum d'un problème d'optimisation sous contraintes. Dans les cas où la convergence n'est pas atteinte, la sous-optimalité du sous espace de projection optimal définissant la fonction de distance peut être quantifiée.

Les contributions proposées sont suffisamment génériques pour pouvoir améliorer différents algorithmes de *clustering* étant données les connaissances externes. En particulier, elles n'utilisent aucune propriété caractérisant de tels algorithmes. La variante ADAUZABOC est alors opérationnelle pour pouvoir être étendu à un contexte de multiplicité des données à travers une plateforme collaborative fondée sur l'échange de contraintes entre vues, prises dans chacune comme des connaissances externes.

Le troisième apport proposé a permis de fonder les bases de la collaboration entre algorithmes de *clustering* quelconques pour atteindre l'objectif de consensus, ou minimisation du désaccord comme suggéré dans le cadre multi-vues. Nous montrons de plus que la collaboration peut être envisagée pour atteindre l'objectif, au contraire, de divergence entre les vues, comme suggéré par les approches dédiées au problème du *clustering* alternatif. En ce sens la plateforme permet, modulo le mécanisme de collaboration, de proposer des solutions au problème du *clustering* multi-vues, rejoignant dans ce contexte le *clustering* d'ensemble et le *clustering* collaboratif, et en même temps au problème du *clustering* alternatif. Les contributions proposées : COBOC et ALTERBOC, se fondant sur ADAUZABOC, permettent alors simultanément d'apprendre un ensemble de fonctions de distances (une par vue ou alternative) et au choix, un ensemble de *clusterings* consensus ou de *clusterings* alternatifs. L'approche nécessite cependant plusieurs paramètres pour espérer atteindre ces objectifs, qu'elle atteint alors de manière heuristique et peu contrôlée. Pour finir elle vise à constituer une contribution de base à l'édifice de la recherche d'une approche unifiée au *clustering* et ses problèmes satellites, préoccupation très actuelle dans la communauté de la fouille de données (figure 4.11).

Perspectives

Les perspectives de ce travail de thèse concernent essentiellement la dernière approche proposée : la plateforme collaborative déclinée en COBOC et ALTERBOC. Parmi les points qui ont été abordés en conclusion de ces approches, certains peuvent se retrouver dans les approches suggérées dans la figure 4.11. En particulier, on s'intéresse à la possibilité d'apprendre directement les contraintes pour tendre vers un objectif de consensus, ou d'obtention d'alternatives, et non de devoir fixer à l'avance la stratégie de génération de ces contraintes. Une autre perspective, beaucoup plus à court terme, est la valorisation expérimentale de l'approche, où la nécessité de l'appliquer sur différents jeux de données notamment multi-représentées.

Concernant les approches BOC, UZABOC et ADAUZABOC, la première perspective envisageable est de changer l'objectif de cohérence. Celui-ci est fondé sur l'ACP, or de nombreuses

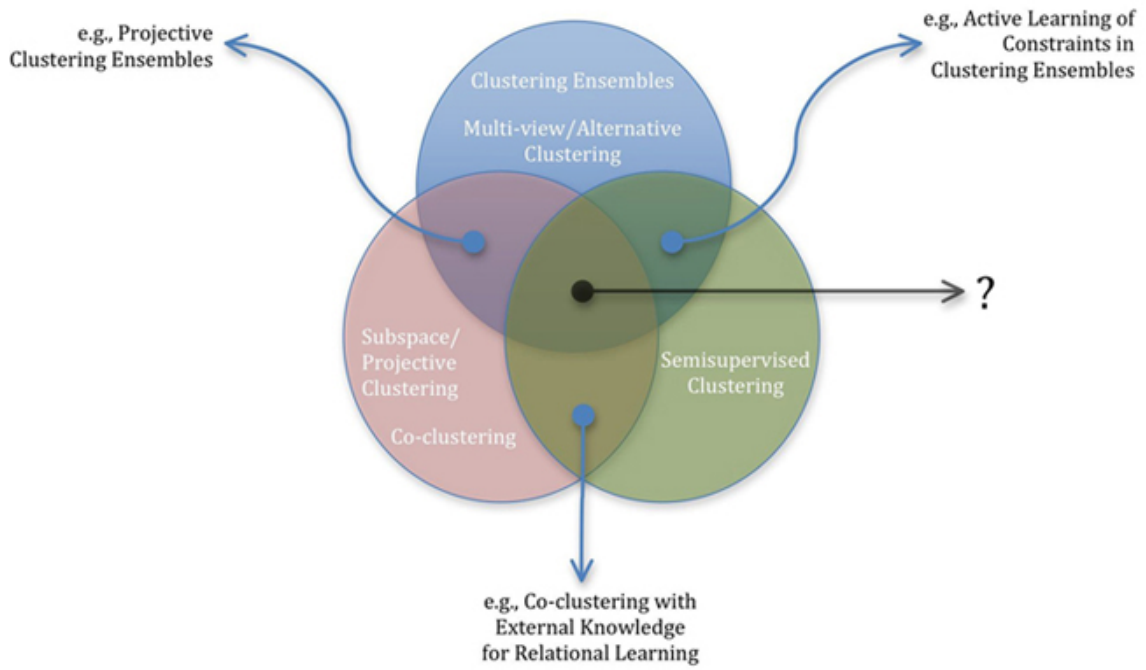

FIGURE 4.11 — L'unification des problèmes du *clustering*. L'objectif actuel est de proposer une approche intégrant un moyen de réaliser simultanément du *clustering* dans des sous-espaces (par exemple par ACP), du *clustering* semi-supervisé, du *clustering* multi-vues et alternatif.

autres techniques de recherche de sous-espaces ou variétés sur lesquels sont distribuées les données existent, et il serait important de tester l'impact de leur utilisation en lieu et place du critère de cohérence choisi. Ce changement aura également une influence sur les développements de CoBoC et AlterBoC, et ils peuvent aller à l'encontre des observations faites. D'un point de vue plus technique, la convergence des approches UzaBoC et AdaUzaBoC n'est pour l'heure qu'observée, et celle-ci n'est pas atteinte dans tous les cas. On peut alors s'interroger naturellement sur l'identification de propriétés sur les données, jointes aux contraintes, permettant de garantir une convergence vers la solution optimale. Notons toutefois que même si l'algorithme d'Uzawa utilisé dans ces approches n'atteint pas d'optimal au sens de la dualité forte, il permet d'obtenir une solution approchée, la meilleure possible et caractérisable par une notion de sous-optimalité qui est quantifiable.

En ce qui concerne l'approche originelle CoFKM, à partir de laquelle se sont fondés tous les développements ultérieurs, un problème solvable dans le modèle CoFKM, est celui de la correspondance entre les groupes. Tel que le modèle est proposé, la correspondance est posée dès l'initialisation des centres dans chaque vue (les mêmes individus sont tirés comme centre initiaux). Le critère de désaccord peut être modifié de sorte à identifier pour un groupe donné, la valeur de son indice dans chaque vue. Toujours pour l'approche CoFKM, il peut être intéressant d'observer la production de *clusterings* alternatifs en changeant le signe de la pénalisation du critère. En effet, comme il a été présenté dans cette thèse, il existe un lien étroit entre la recherche de plusieurs *clusterings* alternatifs à déterminer à partir de données mono-vue, et la

recherche d'un *clustering* consensus à partir de données multi-vues. Cette analogie est concrète dans les approches présentées basées sur le modèle de mélange : CoEM et CAMI. L'un pénalise la somme des critères de log-vraisemblance classiques par une divergence de Kullback-Leibler (KL) entre les *clusterings* locaux, l'autre par l'information mutuelle (MI) entre ceux-ci. Or la dualité entre les mesures KL et MI entre deux *clusterings* est admise, dans le sens où maximiser l'une des quantités revient à minimiser l'autre. La proposition d'une variante de CoFKM pour la recherche d'alternatives se justifie alors pleinement.

Liste des tableaux

2.1 Évaluation externe de CoFKM sur *mfeat* comparé aux approches mono-vues. 81
2.2 Évaluation externe de CoFKM sur *2D2K* comparé aux approches mono-vues. 82
2.3 Évaluation externe de CoFKM sur *mfeat* comparé aux approches centralisées multi-vues. 83
2.4 Évaluation externe de CoFKM sur *2D2K* comparé aux approches centralisées multi-vues. 83
2.5 Évaluation externe de CoFKM sur *mfeat* comparé aux différentes solutions de fusion. 84
2.6 Évaluation externe de CoFKM sur *2D2K* comparé aux différentes solutions de fusion. 84

3.1 Données pour le *clustering* semi-supervisé 123

4.1 Évaluation externe de CoBoC consensus sur *Iris* selon les résultats locaux. 183
4.2 Évaluation externe de CoBoC consensus sur *Iris* selon les résultats locaux. 184
4.3 Évaluation externe de CoBoC complémentaire sur *Iris* selon les résultats locaux. . . 185
4.4 Évaluation externe de CoBoC consensus sur *Wine* selon les résultats locaux. 185
4.5 Évaluation externe de CoBoC complémentaire sur *Wine* selon les résultats locaux. . 186
4.6 Évaluation externe de CoBoC consensus sur *parkinson* selon les résultats locaux. . . 186
4.7 Évaluation externe de CoBoC complémentaire sur *parkinson* selon les résultats locaux.187
4.8 Évaluation externe de CoFKM dans le contexte de la combinaison de modèles. . . . 187
4.9 Évaluation externe de CoBoC consensus sur *Iris* selon différentes fusions finales pour les noyaux K_1 et K_2. 189
4.10 Évaluation externe de CoBoC complémentaire sur *Iris* selon différentes fusions finales pour les noyaux K_1 et K_2. 190
4.11 Évaluation externe de CoBoC consensus sur *Wine* selon différentes fusions finales pour les noyaux K_1 et K_2. 191
4.12 Évaluation externe de CoBoC complémentaire sur *Wine* selon différentes fusions finales pour les noyaux K_1 et K_2. 192
4.13 Évaluation externe de CoBoC consensus sur *parkinson* selon différentes fusions finales pour les noyaux K_1 et K_2. 193
4.14 Évaluation externe de CoBoC complémentaire sur *parkinson* selon différentes fusions finales pour les noyaux K_1 et K_2. 194
4.15 Évaluation externe de CoBoC sur *Iris* selon différentes fusions finales multi-vues. . . 195
4.16 Évaluation externe de CoBoC sur *Wine* selon différentes fusions finales multi-vues. . 196
4.17 Évaluation externe de CoBoC sur *parkinson* selon différentes fusions finales multi-vues.196
4.18 Évaluation externe de CoBoC consensus sur *mfeat* selon les résultats locaux. 197
4.19 Évaluation externe de CoBoC consensus sur *mfeat* selon les résultats locaux obtenus par l'application de plusieurs FKM. 198
4.20 Évaluation externe de CoBoC complémentaire sur *mfeat* selon les résultats locaux obtenus par l'application de plusieurs FKM. 198
4.21 Évaluation externe de CoBoC consensus sur *mfeat* selon les résultats locaux obtenus par l'application d'algorithmes différents. 199

4.22 Évaluation externe de CoBoC complémentaire sur *mfeat* selon les résultats locaux obtenus par l'application d'algorithmes différents. 199
4.23 Évaluation externe de CoFKM sur *mfeat*. 200
4.24 Évaluation externe de CoBoC consensus avec plusieurs algorithmes différents sur *mfeat* selon différentes fusions finales pour les noyaux K_1 et K_2. 201
4.25 Évaluation externe de CoBoC complémentaire avec plusieurs algorithmes différents sur *mfeat* selon différentes fusions finales pour les noyaux K_1 et K_2. 202
4.26 Évaluation externe de CoBoC consensus avec plusieurs algorithmes FKM sur *mfeat* selon différentes fusions finales pour les noyaux K_1 et K_2. 203
4.27 Évaluation externe de CoBoC complémentaire avec plusieurs algorithmes FKM sur *mfeat* selon différentes fusions finales pour les noyaux K_1 et K_2. 204
4.28 Évaluation externe de CoBoC avec plusieurs algorithmes différents sur *mfeat* selon différentes fusions finales multi-vues. 205
4.29 Évaluation externe de CoBoC avec plusieurs algorithmes FKM sur *mfeat* selon différentes fusions finales multi-vues. 206

Table des figures

0.1 Données désordonnées avant *clustering* et ordonnées après *clustering*. 9
0.2 Analyse exploratoire des données 12
0.3 Différents types de données multi-vues. 16
0.4 Problématiques concernant la multiplicité des données et la multiplicité des analyses. 19

1.1 Dendrogramme d'un *clustering* hiérarchique 29
1.2 Résultats d'algorithme agglomératif hiérarchique 29
1.3 Déroulement de KM 31
1.4 Déroulement de DBSCAN 35

2.1 Les différentes fusions du *clustering* multi-vues. 52
2.2 Un modèle COMRAF et sa décompostion en plusieurs COMRAF*. 63
2.3 Évaluation interne de COFKM sur *2D2K*. 80
2.4 Évaluation interne de COFKM sur *mfeat*. 80
2.5 Influence des paramètres η et β sur COFKM. 84
2.6 Évaluation externe de COFKM selon le paramètre η. 85
2.7 Évolution du critère COFKM sur *mfeat* 85
2.8 Évaluation externe de COKFKM sur *WebKB*. 86

3.1 Intégration de contraintes dans le *clustering* semi-supervisé. 92
3.2 Réseau de Markov pour le *clustering* semi-supervisé. 97
3.3 Méta-algorithmes pour le *clustering* semi-supervisé. 108
3.4 Schéma du déroulement d'UZABOC. 118
3.5 Schéma du déroulement d'ADAUZABOC. 122
3.6 Illustration des méthodes de recherche UZABOC et ADAUZABOC. 124
3.7 Légende de l'évaluation interne des approches semi-supervisées. 126
3.8 Légende de l'évaluation externe des approches semi-supervisées. 126
3.9 Convergence de BOC avec KM sur *Iris* centré et réduit. 128
3.10 Convergence de UZABOC avec KM sur *Iris* centré et réduit. 129
3.11 Convergence de ADAUZABOC avec KM sur *Iris* centré et réduit. 130
3.12 Convergence de BOC avec CLINK sur *Iris* centré et réduit. 131
3.13 Convergence de UZABOC avec CLINK sur *Iris* centré et réduit. 132
3.14 Convergence de ADAUZABOC avec CLINK sur *Iris* centré et réduit. 133
3.15 Comparatifs des approches semi-supervisées sur *Iris* centré et réduit. 135
3.16 Comparatifs des approches semi-supervisées sur *Parkinson* centré et réduit. 135
3.17 Comparatifs des approches semi-supervisées sur *Wine* centré et réduit. 136
3.18 Comparatifs des approches semi-supervisées sur *WDBC* centré et réduit. 136
3.19 Comparatifs des approches semi-supervisées sur *Iris* centré et réduit avec contraintes bruitées. 138
3.20 Comparatifs des approches semi-supervisées sur *Parkinson* centré et réduit avec contraintes bruitées. 138

3.21 Comparatifs des approches semi-supervisées sur *Wine* centré et réduit avec contraintes bruitées. 139
3.22 Comparatifs des approches semi-supervisées sur *WDBC* centré et réduit avec contraintes bruitées. 139
3.23 Comparatifs des approches semi-supervisées sur *Iris* centré. 140
3.24 Comparatifs des approches semi-supervisées sur *Parkinson* centré. 141
3.25 Comparatifs des approches semi-supervisées sur *wine* centré. 141
3.26 Comparatifs des approches semi-supervisées sur *WDBC* centré. 142

4.1 *clustering* d'ensemble, *clustering* collaboratif et *alternative clustering*. 148
4.2 Légende pour l'évaluation interne de CoBoC et AlterBoC. 174
4.3 Évolution de l'$AvgNMI$ pour la combinaison de modèle et l'heuristique consensus. . 175
4.4 Évolution de l'$AvgNMI$ pour la combinaison de modèle et l'heuristique complémentaire. 176
4.5 Évolution de l'$AvgNMI$ pour le *clustering* multi-vues et l'heuristique consensus. . . . 177
4.6 Évolution de l'$AvgNMI$ pour le *clustering* multi-vues et l'heuristique complémentaire. 178
4.7 Évolution de l'$AvgNMI$ pour la combinaison de modèle et l'heuristique global. . . . 180
4.8 Évolution de l'$AvgNMI$ pour la combinaison de modèle et l'heuristique complémentaire. 181
4.9 Évolution de l'$AvgNMI$ pour le *clustering* multi-vues et l'heuristique global. 182
4.10 Évolution de l'$AvgNMI$ pour le *clustering* multi-vues et l'heuristique complémentaire. 182
4.11 L'unification des problèmes du *clustering*. 211

Liste des algorithmes

1 DIANA 28
2 AGNES 30
3 KM 31
4 SC 34
5 DBSCAN 35
6 *batch* SOM 37
7 FKM 38
8 EM 41
9 MVDBSCAN 54
10 COFC 56
11 FCPU 58
12 batch-MVADASOM 60
13 COMRAF* 62
14 COEM 65
15 COFKM 71
16 COKFKM 76
17 Cop K-moyennes 94
18 CCHC 95
19 EM contraint 98
20 PCKM 100
21 SSKM 101
22 LLMA 103
23 BC 106
24 BOC 113
25 UZABOC 121
26 ADAUZABOC 123
27 CE 151
28 FT 153
29 SAMARAH 156
30 MOCLE 158
31 COALA 160
32 ADFT 161
33 CAMI 163
34 COBOC consensus 168
35 COBOC complémentaire 169
36 ALTERBOC global 171
37 ALTERBOC complémentaire 172

Bibliographie

[Achtert et al., 2006] Achtert, E., Kriegel, H.-P., Pryakhin, A., and Schubert, M. (2006). Clustering multi-represented objects using combination trees. In Ng, W. K., Kitsuregawa, M., Li, J., and Chang, K., editors, *PAKDD*, volume 3918 of *Lecture Notes in Computer Science*, pages 174–178. Springer.

[Aikake, 1973] Aikake, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrov, B. N. and Csaki, F., editors, *Proceedings of 2nd International Symposium on Information Theory*, pages 267–281. Akademiai Kiado.

[Aupetit, 2006] Aupetit, M. (2006). Learning topology with the generative gaussian graph and the em algorithm. In *Advances in Neural Information Processing Systems*, page 2006.

[Bae and Bailey, 2006] Bae, E. and Bailey, J. (2006). Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *ICDM*, pages 53–62. IEEE Computer Society.

[Basu et al., 2004] Basu, S., Banerjee, A., and Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. In Berry, M. W., Dayal, U., Kamath, C., and Skillicorn, D. B., editors, *SDM*. SIAM.

[Bekkerman and Jeon, 2007] Bekkerman, R. and Jeon, J. (2007). Multi-modal clustering for multimedia collections. In *CVPR*.

[Bekkerman et al., 2006] Bekkerman, R., Sahami, M., and Learned-Miller, E. (2006). Combinatorial Markov Random Fields. In *Proceedings of ECML-06, the 17th European Conference on Machine Learning*, pages 30–41.

[Bezdek, 1981] Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. *Plenum Press, New York*.

[Bickel and Scheffer, 2004] Bickel, S. and Scheffer, T. (2004). Multi-view clustering. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, ICDM '04, pages 19–26, Washington, DC, USA. IEEE Computer Society.

[Bickel and Scheffer, 2005] Bickel, S. and Scheffer, T. (2005). Estimation of mixture models using co-EM. In *16th European Conference on Machine Learning ECML 2001*, volume 3720 of *Lecture Notes in Artificial Intelligence*, pages 35–46. Springer.

[Biernacki, 2009] Biernacki, C. (2009). Pourquoi les modèles de mélange pour la classification ? *Revue de MODULAD*, (40):1–22.

[Blum and Mitchell, 1998] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*.

[Celeux and Govaert, 1992] Celeux, G. and Govaert, G. (1992). A classification em algorithm for clustering and two stochastic versions. *Comput. Stat. Data Anal.*, 14(3):315–332.

[Chang and Yeung, 2004] Chang, H. and Yeung, D.-Y. (2004). Locally linear metric adaptation for semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 20–, New York, NY, USA. ACM.

[Cleuziou et al., 2009] Cleuziou, G., Exbrayat, M., Martin, L., and Sublemontier, J.-H. (2009). CoFKM : a Centralized Method for Multiple-View Clustering. In *ICDM 2009, The Ninth IEEE International Conference on Data Mining*, pages 752–757, Miami, United States.

[Dang and Bailey, 2010] Dang, X. H. and Bailey, J. (2010). Generation of alternative clusterings using the cami approach. In *SDM*, pages 118–129. SIAM.

[Davidson and Basu, 2007] Davidson, I. and Basu, S. (2007). A survey of clustering with instance level constraints. In *ACM Transactions on Knowledge Discovery from Data*, pages 1–41. ACM.

[Davidson and Qi, 2008] Davidson, I. and Qi, Z. (2008). Finding alternative clusterings using constraints. In *ICDM*, pages 773–778. IEEE Computer Society.

[Davidson and Ravi, 2005a] Davidson, I. and Ravi, S. S. (2005a). Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In Jorge, A., Torgo, L., Brazdil, P., Camacho, R., and Gama, J., editors, *PKDD*, volume 3721 of *Lecture Notes in Computer Science*, pages 59–70. Springer.

[Davidson and Ravi, 2005b] Davidson, I. and Ravi, S. S. (2005b). Clustering with constraints: Feasibility issues and the k-means algorithm. In *SDM*.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of Royal Statistical Society B*, 39:1–38.

[Dhillon et al., 2005] Dhillon, I. S., Guan, Y., and Kulis, B. (2005). A unified view of kernel k-means, spectral clustering and graph cuts. Technical Report TR-04-25, University of Texas Dept. of Computer Science.

[Ding et al., 2005] Ding, C., He, X., and Simon, H. D. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proc. SIAM Data Mining Conf*, pages 606–610.

[dos S. Dantas and de Carvalho, 2011] dos S. Dantas, A. B. and de Carvalho, F. (2011). Adaptive batch som for multiple dissimilarity data tables. In *ICTAI*, pages 575–578. IEEE.

[Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231.

[Faceli et al., 2009] Faceli, K., de Souto, M. C. P., de Araujo, D. S. A., and de Carvalho, A. C. P. L. F. (2009). Multi-objective clustering ensemble for gene expression data analysis. *Neurocomputing*, 72(13-15):2763–2774.

[Forestier, 2010] Forestier, G. (2010). Connaissances et classification multistratégie d'objets complexes multisources.

[Frey and Dueck, 2007] Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315:2007.

[Gan et al., 2007a] Gan, G., Ma, C., and Wu, J. (2007a). *Data clustering - theory, algorithms, and applications*. SIAM.

[Gan et al., 2007b] Gan, G., Ma, C., and Wu, J. (2007b). Grid-based clustering algorithms.

[Grozavu and Bennani, 2010] Grozavu, N. and Bennani, Y. (2010). Topological collaborative clustering. *Australian Journal of Intelligent Information Processing Systems*, 12(3). Machine Learning Applications (Part I).

[Grozavu et al., 2011] Grozavu, N., Ghassany, M., and Bennani, Y. (2011). Learning confidence exchange in collaborative clustering. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2011)*, pages 872–879, San Jose, California, USA. IEEE.

[Guénoche, 2011] Guénoche, A. (2011). Consensus of partitions : a constructive approach. *Adv. Data Analysis and Classification*, 5(3):215–229.

[Heer and Chi, 2002] Heer, J. and Chi, E. H. (2002). Mining the Structure of User Activity using Cluster Stability. In *proceedings of the Web Analytics Workshop, SIAM Conference on Data Mining*.

[Jain, 2008] Jain, A. K. (2008). Data clustering: 50 years beyond k-means. In Daelemans, W., Goethals, B., and Morik, K., editors, *ECML/PKDD (1)*, volume 5211 of *Lecture Notes in Computer Science*, pages 3–4. Springer.

[Kailing et al., 2004] Kailing, K., Kriegel, H.-P., Pryakhin, A., and Schubert, M. (2004). Clustering multi-represented objects with noise. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 394–403.

[Karypis and Kumar, 1998] Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, 20(1):359–392.

[Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data. An Introduction to Cluster Analysis.* John Wiley & Sons, Inc.

[Klein et al., 2002] Klein, D., Kamvar, S., and Manning, C. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering.

[Kohonen, 1988] Kohonen, T. (1988). Neurocomputing: foundations of research. chapter Self-organized formation of topologically correct feature maps, pages 509–521. MIT Press, Cambridge, MA, USA.

[Kriegel and Zimek, 2010] Kriegel, H.-P. and Zimek, A. (2010). Subspace Clustering, Ensemble Clustering, Alternative Clustering, Multiview Clustering: What Can We Learn From Each Other? In *Proceedings of MultiClustKDD*.

[Kulis et al., 2005] Kulis, B., Basu, S., Dhillon, I., and Mooney, R. (2005). Semi-supervised graph clustering: a kernel approach. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 457–464, New York, NY, USA. ACM.

[Lashkari and Golland, 2008] Lashkari, D. and Golland, P. (2008). Convex clustering with exemplar-based models. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 825–832. MIT Press, Cambridge, MA.

[Li, 2008] Li, T. (2008). Clustering based on matrix approximation: a unifying view. *Knowl. Inf. Syst.*, 17(1):1–15.

[Liu et al., 2007] Liu, Y., Jin, R., and Jain, A. K. (2007). Boostcluster: boosting clustering by pairwise constraints. In Berkhin, P., Caruana, R., and Wu, X., editors, *KDD*, pages 450–459. ACM.

[Luxburg, 2007] Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability*, volume 1, pages 281–297, Berkeley. University of California Press.

[Martin et al., 2006] Martin, C., grosse Deters, H., and Nattkemper, T. W. (2006). Fusing biomedical multi-modal data for exploratory data analysis. In *ICANN 2006, Part II, LNCS 4132*, pages 798–807.

[Mesghouni et al., 2011] Mesghouni, N., Ghedira, K., and Temani, M. (2011). Unsupervised horizontal collaboration based in som.

[Ng et al., 2001] Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849–856. MIT Press.

[Pedrycz, 2002] Pedrycz, W. (2002). Collaborative fuzzy clustering. *Pattern Recogn. Lett.*, 23(14):1675–1686.

[Regnier, 1965] Regnier, S. (1965). *Sur quelques aspects mathématiques des problèmes de classification automatique*.

[Reza et al., 2009] Reza, G., Md. Nasir, S., Hamidah, I., and Norwati, M. (2009). A survey: Clustering ensembles techniques. *Proceedings of World Academy of Science, Engineering and Technology*, 38:644–653.

[Schwarz, 1978] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464.

[Shental et al., 2003] Shental, N., Hertz, T., Bar-Hillel, A., and Weinshall, D. (2003). Computing gaussian mixture models with em using side-information. In *In Advances in Neural Information Processing Systems 16*. MIT Press.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

[Strehl and Ghosh, 2003] Strehl, A. and Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617.

[Sublemontier et al., 2009] Sublemontier, J.-H., Cleuziou, G., Exbrayat, M., and Martin, L. (2009). Regroupement de données multi-représentées : une approche par k-moyenne flou. In *EGC 2009, 9è Journées Francophones Extraction et Gestion des Connaissances, Actes des ateliers*, Strasbourg, France.

[Sublemontier et al., 2011a] Sublemontier, J.-H., Cleuziou, G., Exbrayat, M., and Martin, L. (2011a). Clustering multi-vues : une approche centralisée. *Revue des Nouvelles Technologies de l'Information, numéro spécial Fouille de Données Complexes : données multiples*.

[Sublemontier et al., 2011b] Sublemontier, J.-H., Martin, L., Cleuziou, G., and Exbrayat, M. (2011b). Integrating pairwise constraints into clustering algorithms: optimization-based approaches. In *ICDMW 2011, The Eleventh IEEE International Conference on Data Mining Workshops*, Vancouver, Canada.

[Sublemontier et al., 2011c] Sublemontier, J.-H., Martin, L., Cleuziou, G., and Exbrayat, M. (2011c). Intégration de contraintes must-link et cannot-link pour la classification : une approche indépendante de l'algorithme. In *XVIIIèmes Rencontres de la Société Francophone de Classification*, pages 153–156, Orléans, France.

[van Breukelen et al., 1998] van Breukelen, M. P. W., Tax, D. M. J., and den Hartog, J. E. (1998). Handwritten digit recognition by combined classifiers,. *Kybernetika*, vol. 34:381–386.

[Vega-Pons and Ruiz-Shulcloper, 2011] Vega-Pons, S. and Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *IJPRAI*, 25(3):337–372.

[Wagstaff and Cardie, 2000] Wagstaff, K. and Cardie, C. (2000). Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110.

[Wagstaff et al., 2001] Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 577–584, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Wemmert et al., 2000] Wemmert, C., Gançarski, P., and Korczak, J. J. (2000). A collaborative approach to combine multiple learning methods. *International Journal on Artificial Intelligence Tools*, 9(1):59–78.

[Wiswedel and Berthold, 2007] Wiswedel, B. and Berthold, M. R. (2007). Fuzzy clustering in parallel universes. *Int. J. Approx. Reasoning*, 45(3):439–454.

[Xing et al., 2002a] Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. (2002a). Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press.

[Xing et al., 2002b] Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. J. (2002b). Distance metric learning with application to clustering with side-information. In Becker, S., Thrun, S., and Obermayer, K., editors, *NIPS*, pages 505–512. MIT Press.

[Yamanishi et al., 2004] Yamanishi, Y., p. Vert, J., and Kanehisa, M. (2004). Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20(1):i363–i370.

[Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.

[Zeng et al., 2010] Zeng, E., Yang, C., Li, T., and Narasimhan, G. (2010). Clustering genes using heterogeneous data sources. *IJKDB*, 1(2):12–28.

[Zhang et al., 2003] Zhang, Z., Kwok, J. T., and Yeung, D.-Y. (2003). Parametric distance metric learning with label information. In *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 1450–1452.

Jacques-Henri SUBLEMONTIER

Classification non supervisée : de la multiplicité des données à la multiplicité des analyses

Résumé : La classification automatique non supervisée est un problème majeur, aux frontières de multiples communautés issues de l'*Intelligence Artificielle*, de l'*Analyse de Données* et des *Sciences de la Cognition*. Elle vise à formaliser et mécaniser la tâche cognitive de classification, afin de l'automatiser pour la rendre applicable à un grand nombre d'objets (ou individus) à classer. Des visées plus applicatives s'intéressent à l'organisation automatique de grands ensembles d'objets en différents groupes partageant des caractéristiques communes. La présente thèse propose des méthodes de classification non supervisées applicables lorsque plusieurs sources d'informations sont disponibles pour compléter et guider la recherche d'une ou plusieurs classifications des données. Pour la classification non supervisée multi-vues, la première contribution propose un mécanisme de recherche de classifications locales adaptées aux données dans chaque représentation, ainsi qu'un consensus entre celles-ci. Pour la classification semi-supervisée, la seconde contribution propose d'utiliser des connaissances externes sur les données pour guider et améliorer la recherche d'une classification d'objets par un algorithme quelconque de partitionnement de données. Enfin, la troisième et dernière contribution propose un environnement collaboratif permettant d'atteindre au choix les objectifs de consensus et d'alternatives pour la classification d'objets mono-représentés ou multi-représentés. Cette dernière contribution répond ainsi aux différents problèmes de multiplicité des données et des analyses dans le contexte de la classification non supervisée, et propose, au sein d'une même plate-forme unificatrice, une proposition répondant à des problèmes très actifs et actuels en *Fouille de Données* et en *Extraction et Gestion des Connaissances*.

Mots clés : Intelligence Artificielle, Apprentissage automatique, Classification non supervisée, Données multi-vues, Consensus de partitions, Co-Apprentissage, Recherche d'alternatives.

Clustering : from multiple data to multiple analysis

Abstract: Data clustering is a major problem encountered mainly in related fields of *Artificial Intelligence*, *Data Analysis* and *Cognitive Sciences*. This topic is concerned by the production of synthetic tools that are able to transform a mass of information into valuable knowledge. This knowledge extraction is done by grouping a set of objects associated with a set of descriptors such that two objects in a same group are similar or share a same behaviour while two objects from different groups does not. This thesis present a study about some extensions of the classical clustering problem for multi-view data, where each datum can be represented by several sets of descriptors exhibing different behaviours or aspects of it. Our study impose to explore several nearby problems such that semi-supervised clustering, multi-view clustering or collaborative approaches for consensus or alternative clustering. In a first chapter, we propose an algorithm solving the multi-view clustering problem. In the second chapter, we propose a boosting-inspired algorithm and an optimization based algorithm closely related to boosting that allow the integration of external knowledge leading to the improvement of any clustering algorithm. This proposition bring an answer to the semi-supervised clustering problem. In the last chapter, we introduce an unifying framework allowing the discovery even of a set of consensus clustering solution or a set of alternative clustering solutions for mono-view data and or multi-view data. Such unifying approach offer a methodology to answer some current and actual hot topic in *Data Mining* and *Knowledge Discovery in Data*.

Keywords: Artificial Intelligence, Machine Learning, Clustering, Multi-view data, Clustering ensemble, Co-Training, Alternative clustering.

Laboratoire d'Informatique Fondamentale d'Orléans
Bâtiment 3IA, rue Léonard de Vinci, B.P. 6759
45067 ORLEANS cedex 2, FRANCE