**Laboratoire des Sciences de l'Image,
de l'Informatique et de la Télédétection**

UMR 7005 UDS/CNRS

École Doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur

Thèse

présentée pour obtenir le grade de

**Docteur de l'Université de Strasbourg
Discipline : Informatique**

par

Germain Forestier

Connaissances et clustering collaboratif d'objets complexes multisources

Soutenue publiquement le 29 Septembre 2010

Membres du jury

Directeur de thèse :	Pierre Gançarski, Maître de Conférences HDR, Université de Strasbourg
Rapporteur externe :	Younès Bennani, Professeur, Université Paris Nord
Rapporteur externe :	Florence Sèdes, Professeur, Université de Toulouse III
Examinateur :	Christoph Eick, Professeur, Université de Houston, Texas
Examinateur :	Cédric Wemmert, Maître de Conférences, Université de Strasbourg
Invitée :	Anne Puissant, Maître de Conférences, Université de Strasbourg

Remerciements

Je tiens tout d'abord à remercier Pierre Gançarski qui a accepté de diriger cette thèse et qui m'a encadré de nombreuses années au cours de ma formation. Durant cette période, il a toujours été présent et a su me guider dans mon travail. Il m'a fourni un cadre de travail idéal pendant ma thèse qui m'a permis de développer mes idées. Il a toujours su apporter le regard critique nécessaire sur mes travaux tout en proposant des solutions et en m'indiquant des voies de recherche pertinentes.

Je remercie également Cédric Wemmert sans qui je n'aurais jamais commencé ce travail de thèse. Il m'a suivi depuis mon stage de DUT et m'a fait découvrir la recherche. Je tiens à saluer son investissement, sa constante disponibilité ainsi que son aide et son soutien dans toutes les pistes que j'ai explorées. Il m'a donné l'envie et la motivation de mener ces travaux à bien, et pour cela je lui dois beaucoup.

Merci à Anne Puissant pour son apport et son expertise thématique tout au long de ces travaux. Nos échanges m'ont permis de mieux cerner les enjeux et les challenges de la télédétection.

Merci à Younès Bennani, Florences Sèdes ainsi qu'à Christoph Eick pour l'intérêt qu'ils ont porté à mon travail et pour avoir accepté d'évaluer ma thèse.

Merci à Jordi Inglada pour m'avoir accueilli au CNES pour un séjour doctoral. Ce séjour m'a permis de mieux saisir les enjeux opérationnels de la télédétection et m'a apporté de nombreuses idées pour mes travaux. Merci également aux membres du service DCT/SI/AP du CNES pour leur accueil chaleureux.

Merci à toute l'équipe du département informatique de l'IUT Robert Schuman pour leur accueil et leur bonne humeur. J'ai pris beaucoup de plaisir à enseigner et à prendre part à la vie du département.

Merci à Sébastien Derivaux avec qui j'ai débuté ma thèse et avec qui j'ai collaboré. Nos discussions passionnées sur la recherche m'ont éclairé.

Merci à Alexandre Blansche pour sa disponibilité, son aide et ses idées lors nos discussions.

Merci à Jonathan Weber avec qui j'ai partagé mes idées, mes passions mais également mes doutes.

Merci à Camille Kurtz et François Petitjean qui ont apporté de la fraîcheur lors de ma dernière année de thèse.

Je tiens également à remercier ma mère qui a financé mes études et m'a toujours soutenu au cours de mes années de thèse.

Enfin, merci à tous mes ami(e)s et les personnes que j'aurais oublié de citer. Vous avez tous (et toutes...) contribué à mon équilibre.

Table des matières

Liste des tableaux

Table des figures

Chapitre 1

Introduction

Sommaire

1.1 Introduction

La création de groupes au sein d'un ensemble d'éléments est une opération omniprésente dans notre société. Les groupes peuvent posséder plusieurs significations, comme par exemple une signification sociale quand ils décrivent un ensemble de personnes qui partagent des caractéristiques communes. Il est naturel de faire appel aux groupes quand il s'agit de structurer, d'organiser ou de résumer un ensemble d'éléments. Ainsi, dans la vie de tous les jours, la notion de *groupe* est utilisée pour l'organisation et la description, comme par exemple des familles de plantes, des genres de musique, des groupes de produits, des groupes sanguins, etc.

Un groupe peut être défini de manière informelle comme *un ensemble d'éléments qui sont rassemblés en raison d'une relation particulière entre ces éléments.* La problématique consistant à former de tels groupes de manière automatique se pose dans de nombreux domaines. En marketing, on va chercher à regrouper des personnes ayant des comportements de consommation similaires pour cibler par exemple une campagne publicitaire. Dans l'étude des réseaux sociaux, on cherche à regrouper les différents membres du réseau pour y faire émerger des communautés. En biologie, on cherche à identifier des groupes de gènes ayant le même comportement pour mettre en place des thérapies géniques.

Deux types d'approches existent et sont à considérer quand la tâche est de regrouper des éléments de manière automatique. La première approche consiste à faire émerger des groupes au sein d'un ensemble d'éléments sans aucune information a priori. Dans ce cas, cette tâche est appelée, selon les domaines, classification non supervisée, classification automatique ou encore en utilisant l'anglicisme *clustering*. Les groupes créés sont appelés *clusters*. L'objectif de cette approche est de découvrir la structure sous-jacente des données pour en extraire de l'information.

La seconde approche intervient quand les groupes existent *a priori*, et le problème est de créer un modèle permettant d'assigner des éléments à ces groupes. Dans ce cas, la tâche est appelée classification supervisée et les groupes sont appelés des *classes* et possèdent une *étiquette* qui correspond au nom de la classe. La classification supervisée nécessite cependant, contrairement à la classification non supervisée, un ensemble d'exemples, c'est-à-dire un ensemble d'éléments dont la

classe est connue a priori. L'objectif, à partir de ces exemples, est de découvrir un modèle des classes pouvant être généralisé à un ensemble de données plus large sous la forme d'un modèle prédictif. Ce processus comporte deux étapes : une étape de construction du modèle à partir des exemples dont la classe est connue, suivie d'une étape de classement des objets dont la classe est inconnue. Cependant, les exemples nécessaires à la construction de ce modèle sont très souvent difficiles à obtenir car ils nécessitent généralement l'intervention d'un expert humain, qui va manuellement affecter une classe à un élément, c'est-à-dire lui affecter une *étiquette*.

De plus, les objets dont la classe est inconnue sont généralement disponibles en grand nombre. C'est pourquoi, les approches non supervisées sont massivement utilisées pour traiter des données de manière automatique. Néanmoins, de nombreux problèmes se posent lors du choix des traitements à appliquer, dont on peut citer les deux plus importants :

1. L'existence d'un grand nombre de méthodes différentes de classification non supervisée ayant des objectifs différents mais fournissant des résultats de bonne qualité ;
2. L'augmentation de la complexité et de la masse des données disponibles qui pousse les approches entièrement non supervisées à montrer leurs limites.

Le premier problème confronte l'expert au choix de la méthode mais aussi au choix de ses paramètres. Ces choix sont souvent liés au domaine d'application et à des connaissances *a priori* de celui-ci, mais aussi aux données à classer elles-mêmes. Le second problème rapporte la question plus conceptuelle de la représentation de connaissances et de leur utilisation dans les algorithmes de clustering pour améliorer le processus de fouille. L'objectif de cette thèse est d'étudier les solutions existantes et de proposer des solutions innovantes à ces deux problèmes via d'une part le clustering collaboratif, et d'autre part l'intégration de connaissances en clustering.

1.1.1 Approche collaborative pour le clustering

De très nombreuses méthodes de classification non supervisées existent et de nouvelles méthodes ou des améliorations de méthodes existantes sont proposées régulièrement par la communauté de la fouille de données. Cette prolifération de méthodes est principalement due au fait qu'il n'existe pas de méthode universelle pouvant s'appliquer de manière performante à tout type de données (*no free lunch theorem* [Wolpert et Macready, 1997]). Par conséquent, il est souvent nécessaire pour l'expert de faire un choix parmi ces méthodes pour traiter ses données. Cependant, une fois la méthode choisie, il reste encore à l'expert à choisir les paramètres de cette méthode qui sont nombreux (par exemple le nombre de groupes attendu, le type de distance utilisée, etc.) et qui influe souvent de manière significative sur le résultat final produit.

Dans la quête d'une solution à ce problème, la communauté scientifique s'est intéressée ces dernières années à l'utilisation conjointe de plusieurs algorithmes pour traiter les mêmes données. De nombreuses méthodes ont été proposées permettant la mise en œuvre d'approches utilisant plusieurs algorithmes et tirant parti des informations contenues dans les différents résultats produits. Chacune de ces approches utilise les informations fournies par les méthodes de clustering d'une manière spécifique. Néanmoins, l'idée affichée étant toujours que l'utilisation de plusieurs méthodes et résultats de clustering peut fournir une information plus pertinente et supplémentaire à celle fournie par une méthode unique. Cette démarche s'inspire et se rapproche du fonctionnement des comités d'experts dans les sociétés humaines.

Dans ce cadre, nous allons étudier plus particulièrement le clustering collaboratif qui consiste à faire collaborer plusieurs méthodes de clustering. Ces différentes méthodes vont partager des informations et vont remettre en cause leurs décisions en fonction des décisions proposées par les autres méthodes. Ainsi, une *discussion* est entreprise entre les méthodes afin de faire converger collectivement les différents résultats afin que ceux-ci soient suffisamment comparables pour qu'un mécanisme d'unification (basé par exemple sur un vote) puisse être effectué. Notre point de départ est la méthode collaborative SAMARAH, proposée par Cédric Wemmert lors de sa thèse [Wemmert, 2000]. Une étude détaillée de cette méthode est présentée ainsi que des expériences illustrant son fonctionnement. Nous présentons ensuite les avancées proposées lors du présent travail de thèse, qui

ont consisté à définir de nouvelles stratégies de résolution de désaccord de classification en clustering collaboratif ainsi qu'une nouvelle approche collaborative basée sur un algorithme génétique. Les expériences présentées illustrent comment nos propositions ont mené à une amélioration de la qualité de la collaboration entre les méthodes.

1.1.2 Intégration de connaissances dans le processus de clustering

Avec l'augmentation de la masse et de la complexité des données disponibles, les approches entièrement non supervisées peuvent montrer leurs limites et produire des résultats ne reflétant pas les attentes de l'expert. L'augmentation du nombre d'objets, du nombre d'attributs, du nombre de clusters ou encore du nombre d'objets atypiques sont des exemples des problèmes de plus en plus présents. Une des voies empruntées par la communauté scientifique pour y apporter une solution consiste à utiliser des connaissances lors du processus de clustering. Ces connaissances, parfois appelées *connaissances du domaine*, sont intégrées de manière croissante dans les processus de fouille de données. Elles peuvent se présenter sous différentes formes suivant les domaines (annotation, étiquetage partiel, pondération, ontologie du domaine, etc.). L'idée est que leur utilisation au sein du processus de fouille de données permettra d'améliorer les résultats en les rendant plus précis, plus robustes et en fournissant à l'expert une information plus riche et plus pertinente.

Les termes les plus communément utilisés pour caractériser ces approches qui se trouvent de fait à la frontière entre approches non supervisées et approches supervisées sont classification *semi-supervisées*, *partiellement supervisées* ou encore *hybrides*. Sous ces termes se regroupent un ensemble de méthodes et d'approches qui utilisent à la fois des données non étiquetées et des données étiquetées ainsi que d'autres types de connaissances (contraintes entre les objets, ontologie, etc.).

Cette problématique de l'intégration de connaissances pose de nouveaux problèmes et a ouvert un champ de recherche totalement novateur dont les fondements théoriques commencent lentement à être posés. De nombreuses approches ont été proposées pour résoudre ces problèmes, utilisant des types de connaissances différents ou les utilisant de manière différente pour améliorer la recherche de solutions à un problème.

Nous étudions dans cette thèse les différentes méthodes proposées dans la littérature pour intégrer des connaissances dans les algorithmes de clustering. Nous présentons plus en détail différents critères de pureté permettant l'intégration de connaissances sous forme d'objets étiquetés. Enfin, nous proposons une approche innovante d'intégration de connaissances dans le cadre du clustering collaboratif. Les connaissances seront utilisées pour guider la collaboration entre les méthodes. Les verrous scientifiques de l'intégration de connaissances sont explorés ainsi que les différents niveaux d'intégration au sein de la méthode collaborative.

1.1.3 Application en observation de la Terre

Les géosciences et plus particulièrement l'observation de la Terre via les images de télédétection ont été le domaine privilégié d'application des propositions faites lors de cette thèse. L'observation de la Terre consiste à étudier la surface terrestre afin par exemple de déterminer l'occupation du sol, les dynamiques urbaines ou encore d'effectuer du suivi de parcelles agricoles. Le domaine de la télédétection fourni de nombreuses données permettant d'effectuer ces analyses. Dans cette thèse, nous nous intéressons uniquement à l'utilisation d'images issues de capteurs optiques embarqués à bord de satellites ou aéroportés.

L'image de télédétection est le prototype même d'une donnée complexe de par sa structure physique mais aussi par le fossé sémantique entre les informations bas niveau (les valeurs radiométriques des pixels) et les informations à extraire (par exemple l'occupation du sol). Ce fossé sémantique est défini comme le manque de concordance entre l'information bas niveau et l'interprétation faite par un expert [Smeulders et al., 2000].

L'extraction de l'information contenue dans une image de télédétection peut être réalisée manuellement par un photo-interprète. Ce processus d'interprétation visuelle est cependant con-

sommateur de temps, en particulier avec l'augmentation de la précision des images qui multiplie la masse de données à traiter. L'automatisation de l'extraction d'informations devient alors une nécessité et un fort besoin a été exprimé de la part des experts géographes concernant le développement de nouveaux outils d'analyse adaptés à ces données. Le principal verrou scientifique réside dans la modélisation de la connaissance de l'expert sur l'interprétation de ces images en vue de l'automatisation du processus d'interprétation.

Dans ces images, chaque pixel d'une image est caractérisé par une information radiométrique sur différentes bandes spectrales et est représenté sous la forme d'un ensemble de valeurs numériques. Le clustering peut être appliqué directement au niveau des pixels, la tâche de clustering consistant alors à affecter un cluster à chaque pixel. On parlera d'approches *basées pixels*. L'objectif est de regrouper automatiquement les pixels qui ont un comportement radiométrique similaire et donc de créer des groupes de pixels similaires. Il est à la charge de l'expert géographe de faire correspondre les groupes identifiés automatiquement à une sémantique correspondante aux classes thématiques d'occupation du sol (par exemple la végétation, les zones urbaines, etc.).

L'arrivée à la fin des années 90 de la très haute résolution spatiale (THRS), qui correspond à une augmentation importante de la précision des images, a mis à mal les méthodologies classiquement employées par les experts géographes dans le cadre de l'observation de la Terre. En effet, alors que les images à faible résolution ne permettaient qu'une analyse de zones thématiques, les images à très haute résolution permettent une analyse directe des objets géographiques : maison, arbre, route, etc. Pour permettre le traitement de ces nouvelles données, de nouvelles méthodes d'analyse ont été proposées. Celles-ci sont communément appelées méthodes *basées régions*. Dans ces approches, l'image est découpée en segments par un processus de *segmentation*. Les segments sont alors caractérisés par des attributs variés (informations spectrales, de texture ou de forme). Ce sont ces régions (les segments caractérisés) qui sont ensuite regroupées par l'algorithme de clustering.

Dans cette thèse, nous présentons nos propositions pour la formalisation des connaissances de l'expert et leur mise en œuvre pour l'identification automatique d'objets géographiques dans les images de télédétection. Plus précisément, nous étudions comment ces connaissances peuvent, soit être utilisées en segmentation d'image, soit conjointement avec une méthode de clustering collaboratif pour l'étiquetage de clusters de régions. Ces approches innovantes ont permis de mettre en place un processus automatique d'interprétation d'image guidé par les connaissances de l'expert. Loin de vouloir fournir un système totalement automatique, le but est d'aider l'expert dans le processus d'interprétation en lui fournissant des outils lui permettant d'accélérer les traitements.

1.1.4 L'enjeu des données multisources

Avec la multiplication des sources et des formats de données, il est souvent possible d'obtenir plusieurs sources de données qui ne sont pas toujours utilisables dans un processus unique de fouille de données. Cependant, la majorité des approches de fouille de données ne considèrent souvent qu'une source de données à la fois. Différentes sources de données peuvent être traitées par un algorithme mais ce traitement est généralement effectué de manière séquentielle. Il n'existe donc pas, la plupart du temps, d'échange automatique d'informations entre les exécutions et les résultats obtenus lors des différentes exécutions. Ces informations sont généralement centralisées par l'expert qui va, par exemple, modifier les paramètres de son algorithme à appliquer sur un jeu de données en fonction des résultats obtenus sur un autre. Il est donc nécessaire d'envisager de nouvelles approches qui peuvent traiter plusieurs vues des données et tirer parti de ces informations hétérogènes.

Ainsi, pour pouvoir utiliser toutes les informations disponibles, des approches récentes se sont intéressées à l'utilisation conjointe de plusieurs sources de données dans un processus unique et conjoint de clustering. Ces approches, souvent qualifiées de *mutilsources* ou *multivues* cherchent à tirer parti des informations disponibles dans chacune des sources pour améliorer l'extraction de connaissances.

Les concepts auxquels ces méthodes multisources font appel sont intimement liés à l'utilisation

de plusieurs méthodes de clustering. En effet, il est possible que la nature des données oblige l'expert à utiliser plusieurs méthodes différentes pour traiter les différentes vues des données. Dans ce cadre, nous présentons dans cette thèse l'utilisation de données multisources en clustering collaboratif dans le cas de l'image de télédétection. Deux types d'approches sont étudiées. Dans la première, plusieurs images de télédétection hétérogènes sont utilisées simultanément dans un processus de clustering collaboratif pour effectuer une analyse multisource. Dans la seconde, nous faisons appel à l'utilisation de librairies spectrales pour effectuer de la simulation de capteurs. Plusieurs vues simulées de ces librairies sont ensuite utilisées lors d'un processus de clustering collaboratif.

1.2 Contenu du document

Ce document est structuré en trois parties. Dans une première partie, nous commençons par présenter la tâche de clustering ainsi que les méthodes les plus couramment mises en œuvre dans ce domaine. Nous présentons les limites et les problèmes posés qui motivent le travail de recherche effectué lors de cette thèse. Nous étudions ensuite les différentes approches proposées jusqu'alors concernant l'utilisation conjointe de plusieurs méthodes de clustering pour répondre aux verrous précédemment présentés. L'approche collaborative SAMARAH, initialement proposée par Cédric Wemmert lors de sa thèse [Wemmert, 2000], est le point départ d'une étude détaillée suivie de propositions novatrices dans le domaine du clustering collaboratif. Nous nous attachons particulièrement à étudier nos propositions en terme de résolution de désaccord de classification entre résultats de clustering.

Dans une seconde partie, nous étudions les mécanismes d'intégration des connaissances dans les algorithmes de clustering. Nous présentons ensuite nos propositions pour l'utilisation de critères de pureté en vue de l'intégration de connaissances dans les algorithmes. Enfin, nous détaillons comment l'intégration de connaissances a été effectuée en clustering collaboratif pour guider le processus de collaboration. Des expériences illustrent la pertinence des approches proposées.

Les géosciences et plus particulièrement l'observation de la Terre via les images de télédétection étant le domaine privilégié d'application des propositions faites lors de cette thèse, nous proposons dans une troisième partie un ensemble de méthodes permettant le traitement de ces données à l'aide des approches vues dans les deux parties précédentes. Enfin, nous nous intéressons au traitement de données multisources, problématique émergente dans ce domaine.

Chacune de ces parties est composée d'un ensemble de chapitres dont voici le contenu :

Première partie : Clustering collaboratif

Chapitre 2 : ce chapitre est une introduction aux principales approches de clustering sans se vouloir exhaustif. Il précise l'objectif et les enjeux du clustering ainsi que les différentes approches existantes pour l'évaluation de résultats de clustering.

Chapitre 3 : ce chapitre présente la problématique du clustering collaboratif. Les différentes approches existante dans la littérature qui utilisent plusieurs méthodes et résultats de clustering sont présentées et comparées. Les fondements du clustering collaboratif sont ensuite étudiés ainsi que différentes approches proposées lors de cette thèse pour améliorer la collaboration entre les méthodes.

Seconde partie : Connaissances et clustering

Chapitre 4 : ce chapitre est centré sur l'utilisation de connaissances en clustering. Un état de l'art des approches pour intégrer des connaissances dans les algorithmes de clustering est dressé. Un ensemble de critères permettant l'intégration de connaissances données sous la forme d'objets

étiquetés est ensuite présenté et ceux-ci sont comparés. Enfin, nos propositions pour l'intégration de connaissances à différents niveaux de la méthode de clustering collaboratif sont détaillées.

Troisième partie : Applications en télédétection

Chapitre 5 : ce chapitre s'intéresse à la problématique de l'observation de la Terre. Une introduction aux données disponibles en télédétection situe les enjeux de ces données complexes. Un formalisme permettant la modélisation des connaissances sur les objets géographiques est présenté. Ces connaissances structurées par ce nouveau formalisme sont ensuite utilisées dans un processus de segmentation d'image. Finalement, une méthode où ces connaissances sont utilisées en clustering collaboratif pour effectuer de l'étiquetage de clusters est présentée.

Chapitre 6 : ce chapitre présente la problématique de l'utilisation de données multisources. Les cas des images de télédétection et des librairies spectrales sont présentés pour illustrer les enjeux de ce type de données complexes. Une nouvelle méthode de clustering multisource adaptée au traitement d'images de télédétection hétérogènes est proposée. Enfin, l'adaptation de la méthode permettant son utilisation dans le cadre du traitement de données multisources est présentée en détail.

Conclusion et perspectives

Chapitre 7 : ce chapitre conclue la thèse en présentant le bilan général de celle-ci ainsi que les perspectives à court, moyen et long terme. Il détaille les apports des travaux de recherche menés lors de ce travail de thèse.

Première partie : Clustering collaboratif

Chapitre 2

Le clustering

Sommaire

2.1 Introduction

Le *clustering*, en français *regroupement* ou *partitionnement*, est une tâche dont l'objectif est de trouver des groupes au sein d'un ensemble d'éléments (appelés par la suite *objets*). Ces objets sont décrits par des *caractéristiques*, encore appelées *attributs*, qui décrivent les propriétés des objets. Les groupes recherchés, communément appelés des *clusters*, forment des ensembles homogènes d'objets du jeu de données partageant des caractéristiques communes. Un exemple de jeu de données décrites par deux attributs et contenant trois clusters identifiables visuellement est illustré sur la figure 2.1. Il existe de très nombreuses méthodes de clustering permettant de créer ces clusters de manière automatique, chacune utilisant une stratégie et un objectif propre pour construire les clusters.

Pour réaliser cette opération de regroupement, on fait fréquemment appel à la notion de *similarité* entre les objets dans les données. En effet, cette notion de similarité prend tout son sens en clustering car il s'agit d'évaluer à quel point deux éléments sont similaires (ou dissimilaires) pour les regrouper ou les séparer. Le choix de la mesure de similarité permettant de comparer les objets entre eux va induire la façon de les regrouper. En utilisant deux définitions de similarité différentes, les objets ne seront pas comparés, et de fait regroupés ou non, de la même façon.

Cette notion de similarité est une première étape pour définir un algorithme permettant de regrouper les objets, mais n'est pas suffisante. En effet, il est nécessaire de décrire la stratégie utilisant cette similarité et permettant la construction explicite des clusters. Plusieurs stratégies

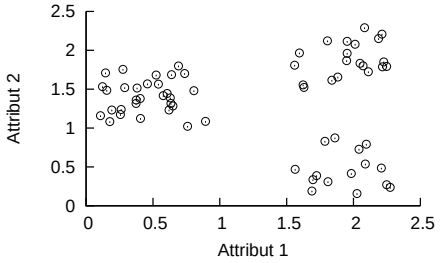


Fig. 2.1: *Exemple d'un jeu de données décrites par deux attributs et contenant trois clusters identifiables visuellement.*

peuvent être mises en place en utilisant une même mesure de similarité. Ces premiers constats sont déjà une explication du nombre important de méthodes de clustering existantes.

L'objectif de ce chapitre est d'introduire la problématique du clustering ainsi que les concepts de base qui seront utilisés dans la suite de ce mémoire de thèse. L'accent sera mis sur la complexité de la tâche de clustering, ainsi que sur le nombre important de méthodes et de variantes disponibles. Des états de l'art plus détaillés sont disponibles dans la littérature, le lecteur souhaitant une description plus avancée des méthodes pourra s'y référer [Berkhin, 2002; Jain et al., 1999; Rauber et al., 2000; Xu et Wunsch, 2005]. Dans ce chapitre nous décrivons les principaux types de résultats en clustering (section 2.2.1), puis nous présentons les principales approches existantes pour les obtenir (section 2.2.2). Nous étudions ensuite les différentes méthodes d'évaluation d'un clustering (section 2.3). Enfin, nous étudions également les principaux problèmes et limites du clustering (section 2.4) avant de conclure (section 2.5).

Notations :

- Soit $X = \{x_1, \ldots, x_n\}$ l'ensemble des n objets à classer ;
- Soit $\mathcal{C} = \{C_1, \ldots, C_K\}$ un résultat de clustering comportant K clusters ;
- Soit $|C_i|$ le nombre d'objets contenus dans le cluster C_i ;
- Soit $d(x_1, x_2)$ une distance entre les objets x_1 et x_2 ;
- Soit $D(C_1, C_2)$ une distance entre les clusters C_1 et C_2.

2.2 Différentes approches en clustering

Loin de vouloir faire un état de l'art exhaustif de toutes les méthodes existantes, nous présentons dans cette section les concepts clefs du clustering. Il est ainsi possible de regrouper les approches selon des caractéristiques communes. La description des méthodes de clustering utilisées dans cette thèse est présente en Annexe A.

La première distinction à faire concerne le type de résultat obtenu. Suivant les méthodes, les clusters obtenus peuvent être des ensembles durs ou flous. Certains objets peuvent ne pas être classés, et certains clusters peuvent se recouvrir. De plus, le résultat n'est pas forcément plat, et peut se présenter sous la forme d'une hiérarchie. Ces différents types de résultats sont présentés dans la section 2.2.1.

Les algorithmes de clustering diffèrent également par la stratégie mise en place pour construire les clusters. Comme introduit au début de ce chapitre, la notion de similarité est utilisée par une part importante des approches. Cependant, d'autres méthodes à base de densité ou de modèles probabilistes existent. Ces différentes approches sont présentées dans la section 2.2.2.

	C_1	C_2	C_3
x_1	1	0	0
x_2	0	1	0
x_3	0	0	1
x_4	0	0	1

Exemple de résultat dur.

	C_1	C_2	C_3
x_1	1	1	0
x_2	0	1	1
x_3	0	1	1
x_4	0	0	1

Exemple de résultat doux.

	C_1	C_2	C_3
x_1	0,9	0,1	0
x_2	0	0,8	0,2
x_3	0	0,3	0,7
x_4	0	0	1,0

Exemple de résultat flou.

Tab. 2.1: *Exemple des degrés d'appartenance des objets aux clusters pour un résultat dur, dou et flou.*

2.2.1 Structures des résultats de clustering

Le résultat d'un algorithme de clustering peut se présenter sous différentes formes selon qu'il est possible ou non que deux clusters se chevauchent, c'est-à-dire qu'un objet puisse appartenir ou non à plusieurs clusters en même temps.

2.2.1.1 Clustering dur, doux et flou

Le résultat le plus simple et le plus souvent rencontré est le clustering dur (*hard clustering*). Dans un clustering dur, chaque élément appartient à un et un seul cluster. L'ensemble des données X est divisé en un ensemble de K clusters, $\mathcal{C} = \{C_1, \ldots, C_K\}$, formant une partition de X, c'est-à-dire $\cup_{k=1}^{K} C_k = X$.

Ce type de résultat est le plus courant et le plus facilement interprétable par l'expert. Cependant il peut être nécessaire de donner plus de flexibilité aux clusters. En effet, il peut arriver que certains objets se distinguent de manière trop significative des autres objets, et leur affecter un cluster peut perturber le processus de clustering. Il arrive que ces objets soient rejetés et qu'aucun cluster ne leur soit affecté dans le résultat final. On parle alors de clustering dur partiel, c'est-à-dire que chaque objet appartient à un ou aucun cluster.

De plus, la frontière entre les clusters peut être difficile à définir, et il arrive que certains objets soient à la frontière de plusieurs clusters. Pour pouvoir refléter ce type d'appartenance, le clustering doux (*soft clustering*) permet à chaque objet d'appartenir à un ou plusieurs clusters. On peut alors parler de clustering doux partiel si dans le résultat, un élément peut appartenir à aucun, un ou plusieurs clusters.

L'appartenance à plusieurs clusters est cependant difficile à interpréter pour l'expert. En effet, plus les objets vont appartenir à de nombreux clusters, plus le résultat va perdre en précision et va rendre difficile son interprétation. La *clustering flou* apporte alors une solution, en permettant à chaque élément d'appartenir à chacun des clusters selon un certain degré d'appartenance. Il est toujours possible de revenir à un clustering dur en sélectionnant pour chaque objet le cluster dont l'appartenance est maximale. Le tableau 2.1 présente une illustration des degrés d'appartenance d'objets aux clusters pour un résultat dur, doux et flou.

2.2.1.2 Clustering hiérarchique

La majorité des méthodes proposent un résultat sous la forme d'une structure plate, c'est-à-dire sans lien entre les clusters. Il est cependant naturel pour certaines applications de représenter le résultat sous la forme d'une hiérarchie de clusters. On peut facilement imaginer des groupes relativement grossiers situés à un niveau élevé dans la hiérarchie, qui vont se spécialiser plus on descendra dans cette hiérarchie. Plus un cluster sera bas dans la hiérarchie plus il contiendra un faible nombre d'objets mais qui seront plus similaires. Dans un clustering hiérarchique, un cluster peut être divisé en sous clusters, l'ensemble des clusters étant généralement représenté par un arbre. Un objet appartient à une et une seule feuille dans la hiérarchie, mais également à son nœud père, et ainsi de suite jusqu'à la racine. Les méthodes de clustering hiérarchique permettent d'obtenir ce type de résultats. Deux grands types d'approches de clustering hiérarchique existent : les approches par agglomération (ou ascendantes) et les approches par division (ou descendantes).

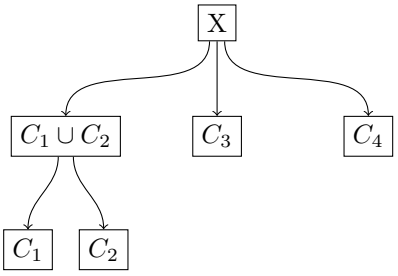


Fig. 2.2: *Exemple de résultat hiérarchique.*

Dans les approches par agglomération, l'algorithme part des objets et ceux-ci sont ensuite regroupés jusqu'à obtenir un cluster unique contenant tous les objets. Les approches divisives partent elles, de l'ensemble des données, et les divisent en clusters qui sont ensuite divisés à leur tour de manière récursive. La figure 2.2 montre un exemple de résultat de clustering hiérarchique à 4 clusters.

2.2.2 Méthodes de clustering

Dans la section précédente nous avons présenté les principales structures de résultats qui peuvent être obtenues en clustering. Dans cette section, nous présentons les principales familles de méthodes de regroupement des données en clusters. Cette taxonomie est inspirée des articles d'état de l'art dans le domaine [Berkhin, 2002; Jain et al., 1999; Rauber et al., 2000; Xu et Wunsch, 2005]. Les méthodes peuvent être séparées en quatre groupes :

- Les méthodes basées sur une distance (section 2.2.2.1)
- Les méthodes basées sur une grille (section 2.2.2.2)
- Les méthodes probabilistes (section 2.2.2.3)
- Les méthodes hiérarchiques (section 2.2.2.4)

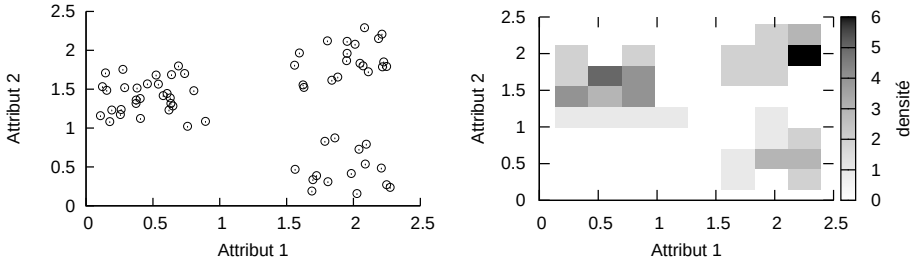
2.2.2.1 Méthodes basées sur une distance

Les méthodes basées sur les distances sont parmi les premières méthodes de clustering à avoir été proposées et sont toujours très populaires aujourd'hui. Ces méthodes se basent sur la notion de distance entre objets du jeu de données, en posant que si deux objets sont proches suivant cette distance, ils doivent être regroupés ensemble dans un même cluster. Les principales méthodes utilisant une distance sont les méthodes à base de prototypes, les méthodes neuronales et les méthodes à base de densité.

Les méthodes basées sur les prototypes définissent pour chaque cluster un objet représentant de ce cluster. Si ce représentant est calculé comme la moyenne des éléments du cluster, il est alors appelé centroïde. S'il s'agit d'un objet particulier du cluster alors celui-ci est appelé un médoïde. Les algorithmes KMEANS [Macqueen, 1967] et FUZZY-C-MEANS [Dunn, 1974] sont les algorithmes les plus connus de cette famille d'algorithme. Ces méthodes permettent de trouver des formes de clusters convexes et sont très utilisées notamment à cause de leur coût algorithmique faible.

Les méthodes neuronales s'inspirent du fonctionnement des neurones humains pour découvrir des clusters. On pourra citer l'algorithme SOM (*Self Organising Map*) [Kohonen, 1984] qui consiste à projeter une carte à deux dimensions de neurones inter-connectés sur les données. Ces neurones sont ensuite déplacés itérativement pour correspondre au mieux aux données. L'algorithme GNG (*Growing Neural Gas*) [Fritzke, 1995] est une méthode neuronale inspirée de SOM qui génère des neurones qui inter-agissent entre eux en simulant le fonctionnement de vrais neurones.

Les méthodes basées sur la densité considèrent que les clusters sont des régions denses au sein des données séparées par des régions moins denses. Cette notion de densité est fortement lié au calcul du voisinage d'un objet, c'est-à-dire les objets situés à une certaine distance de celui-ci. Plus un objet aura de voisins dans ce voisinage, plus il sera considéré comme appartenant à une région



(a) Exemple de données avec trois clusters. (b) Densité de la grille.

Fig. 2.3: *Un jeu de données à trois clusters et la grille de densité associée.*

dense. Ce type d'algorithme nécessite généralement la définition de seuils, par exemple le seuil du voisinage à considérer. Le voisinage V_ϵ d'un objet x est défini tel que $V_\epsilon(x) = \{y \in X | d(x, y) \leq \epsilon\}$ pour ϵ un seuil donné et $d(x, y)$ une mesure de distance entre x et y. Contrairement aux méthodes basées sur des prototypes, cette approche permet de découvrir des clusters concaves dans l'espace de données [Han, 2005]. Plusieurs méthodes basées sur le concept de densité existent, la plus connue étant DBSCAN [Ester et al., 1996].

2.2.2.2 Méthodes basées sur une grille

Les méthodes à base de grille sont fondées sur le principe de la discrétisation de l'espace des données. Celui-ci est décomposé en un ensemble de cellules qui forment l'unité de la grille. Ces méthodes ont été proposées pour réduire l'explosion combinatoire des méthodes à base de densité qui fait suite à l'augmentation du nombres d'objets. La densité d'une cellule est basée sur le rapport entre le nombre de points présents dans cette cellule et son volume. Ainsi, la relation de voisinage qui servait dans les méthodes à base de densité est remplacée par le voisinage entre les cellules, ce qui permet de réduire le nombre d'objets à regrouper. Un exemple du calcul de cette grille est donné sur la figure 2.3.

Le processus de clustering dans les méthodes à base de grille consiste à regrouper les cellules denses les plus proches. L'algorithme BANG [Schikuta et Erhart, 1997] effectue ce regroupement de manière hiérarchique, en partant de la grille et en fusionnant successivement les cellules denses voisines dont la différence de densité ne dépasse pas un certain seuil. L'algorithme CLIQUE [Aggarwal et al., 1999] est une méthode très populaire basée sur les grilles. Il consiste à partir des cellules, et à ne considèrer que les cellules dont la densité est supérieure à un seuil. La particularité de CLIQUE est d'explorer plusieurs sous-espaces, c'est-à-dire de considérer plusieurs sous-ensembles des attributs qui décrivent les objets. La grille et les densités sont calculées dans ces sous-espaces, ce qui permet d'effectuer une sélection d'attributs de manière implicite. Ceci permet de ne conserver que les attributs faisant ressortir la densité des cellules, et donc les clusters.

L'utilisation de grilles adaptatives dans ces algorithmes consiste à considérer des grilles non-uniformes, c'est-à-dire dont les cellules n'ont pas toutes la même géométrie. En effet, dans le cas où les clusters ont des densités différentes, il peut être intéressant d'avoir des cellules n'ayant pas la même résolution dans tout l'espace des données. Adapter la grille à la densité locale permet d'éviter ce problème. L'algorithme MAFIA [Nagesh et al., 2000] propose par exemple une évolution de CLIQUE en créant des grilles adaptatives.

2.2.2.3 Méthodes probabilistes

Les méthodes probabilistes supposent que les données suivent une certaine loi de probabilité. L'objectif est d'estimer les paramètres de cette loi et de définir un modèle de mélange de lois

pour représenter les différents clusters. Ces méthodes font l'hypothèse qu'à chaque cluster C_i est associée une loi de probabilité $P(x, \theta_i)$ de paramètres θ_i qui permet de déterminer la probabilité d'appartenance de x à C_i. Si on note π_i la proportion de la i-ème loi dans le mélange, les paramètres du modèle sont : $\Phi = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$ et la fonction de densité est :

$$P(x, \Phi) = \sum_{i=1}^{K} \pi_i P(x, \theta_i)$$

Les méthodes de clustering probabilistes cherchent à approximer les paramètres Φ du modèle. La probabilité d'appartenance aux clusters peut être interprétée comme un degré d'appartenance à ce cluster. On pourra notamment citer l'algorithme EM [Dempster et al., 1977]. Á noter qu'en général, les lois considérées sont supposées gausiennes.

2.2.2.4 Méthodes hiérarchiques

Les méthodes hiérarchiques construisent une hiérarchie de clusters, c'est-à-dire un arbre de clusters pouvant se présenter sous la forme d'un dendrograme. Chaque nœud contient ses clusters enfants, et les nœuds frères partitionnent les objets contenus dans leurs parents. Ce type d'approche permet d'explorer les données à différents niveaux de granularité. Les méthodes de clustering hiérarchique sont décomposées en deux types d'approches, les approches ascendantes et les approches descendantes. Dans les approches ascendantes, l'algorithme part d'un grand nombre de clusters et ceux-ci sont ensuite fusionnés jusqu'à n'obtenir plus qu'un unique groupe contenant tous les objets du jeu de données. Les approches descendantes partent, de l'ensemble des données, et le divisent en clusters qui sont ensuite divisés récursivement.

Dans les approches ascendantes, il est nécessaire de définir un critère de similarité entre les clusters, qui permet à chaque étape de l'algorithme de choisir les deux clusters à fusionner. Une hypothèse importante est l'hypothèse de monotonie. La monotonie signifie que si $s_1, s_2, \ldots, s_n$ sont les similarités des clusters fusionnés au cours du clustering hiérarchique alors $s_1 \geq s_2 \geq \ldots \geq s_n$. Un clustering hiérarchique non monotone contient au moins une inversion $s_i < s_{i+1}$ ce qui contredit l'hypothèse fondamentale de fusionner les deux meilleurs clusters candidats à chaque étape.

Le clustering hiérarchique ascendant ne nécessite pas de prédéfinir un nombre de clusters. Cependant, si un clustering plat des données est nécessaire, il est possible d'effectuer une coupe dans la hiérarchie au niveau qui propose le nombre de clusters demandé. Il est également possible de définir un niveau de similarité entre les clusters au-delà duquel on ne souhaite plus fusionner les clusters. Un certain nombre d'heuristiques existent pour choisir à quel niveau de la hiérarchie il est judicieux d'effectuer une coupe pour obtenir le meilleur partitionnement des données.

Il existe plusieurs stratégies pour calculer la similarité entre les clusters, les plus connues étant : *single-link*, *complete-link* et *average-link*. La stratégie *single-link* compare les deux clusters en considérant la distance minimale entre les objets des deux clusters :

$$D_s(C_1, C_2) = \min_{x \in C_1, y \in C_2} d(x, y) \tag{2.1}$$

La stratégie *complete-link* considère la distance maximale entre les objets des deux clusters :

$$D_c(C_1, C_2) = \max_{x \in C_1, y \in C_2} d(x, y) \tag{2.2}$$

Enfin, la stratégie *average-link* considère la moyenne des distances des objets des deux clusters :

$$D_a(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x \in C_1} \sum_{y \in C_2} d(x, y) \tag{2.3}$$

La figure 2.4 illustre graphiquement ces différentes stratégies.

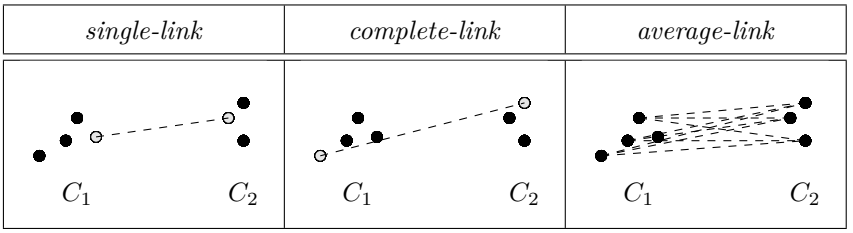


Fig. 2.4: *Illustration des différentes stratégies de regroupement de clusters en clustering hiérarchique.*

2.2.2.5 Méthodes spectrales

Les méthodes spectrales ont recontré un fort succès ces dernières années. Le cœur du clustering spectral consiste à considérer le problème de clustering comme un problème de coupe de graphe normalisée et à utiliser le Laplacien de la matrice d'adjacence du graphe qui représente alors les données. Une matrice de similarité est utilisée qui représente la similarité entre les couples d'objets. Ces algorithmes utilisent les vecteurs propres du Laplacien et sont de fait peu appropriés à de gros volumes de données en raison du coup important de son calcul. Cependant, il suffit de déterminer une matrice de noyau pour l'appliquer, ce qui facilite le traitement de données hétérogènes (définition de différentes matrices de noyau).

Une de ces techniques est l'algorithme de Shi-Malik [Shi et Malik, 2000]. Il partitionne les données en deux ensembles (X_1, X_2) en se basant sur le vecteur propre correspondant à la deuxième plus petite valeur propre du Laplacien de la matrice. Le partitionnement peut alors être effectué de plusieurs manières, par exemple en prenant la médiane des composantes du vecteur propre et en plaçant les points dont la composante est supérieure à un seuil. L'algorithme utilise une méthode hiérarchique de manière récursive pour créer des clusters.

Une autre approche appelée l'algorithme de Meila-Shi [Shi, 2000] prend les vecteurs propres correspondants aux k plus grandes valeurs propres de la matrice pour un k donné et utilise un algorithme de partitionnement (par exemple KMEANS) pour regrouper les points par rapport à leurs composantes respectives dans les vecteurs propres.

2.3 Critères d'évaluation de la qualité d'un clustering

L'évaluation de la qualité d'un résultat de clustering est un domaine de recherche actif et de nombreuses méthodes continuent d'être proposées régulièrement. Ceci est dû au fait que l'évaluation d'un clustering contient toujours une part de subjectivité et qu'il est impossible de définir un critère universel qui permettrait une évaluation sans biais de tous les résultats produits par toutes les méthodes de clustering existantes. Cependant, un certain nombre de critères existent et sont utilisés de manière récurrente par de nombreux chercheurs pour comparer les résultats obtenus. Comme il existe un nombre important de résultats de clustering possibles pour un même jeu de données, l'objectif est d'évaluer si un de ces résultats est meilleur qu'un autre. Cette notion de *meilleur* est à définir et est souvent dépendante de la méthode utilisée.

2.3.1 Taxonomie des méthodes d'évaluation

Plusieurs taxonomies des méthodes d'évaluation ont été proposées dans la littérature [Halkidi et al.; Jain et Dubes, 1988; Tan et al., 2005]. Elles les regroupent principalement en trois familles. La première famille contient les *mesures non supervisées* qui utilisent uniquement des informations internes aux données comme par exemple la distance entre les objets. Ces mesures sont également appelées mesures de qualité internes. La seconde famille contient les *mesures supervisées* qui calculent le degré de correspondance entre le clustering produit par l'algorithme et un partitionnement connu des données. Ces mesures sont aussi connues sous le nom de mesures de qualité externes. Le

dernier groupe contient les mesures dites *relatives*, qui permettent pour un même algorithme de comparer les clusterings produits par celui-ci. Les mesures relatives sont donc simplement l'utilisation de critères internes ou externes pour faire un choix parmi plusieurs résultats produit par un même algorithme. Dans cette section, nous allons voir les principales mesures de qualité internes qui permettent d'évaluer un clustering. Les mesures de qualité externes seront étudiées dans le Chapitre 3 sur l'intégration de connaissances dans les algorithmes de clustering et sont également présentées en Annexe B.

2.3.2 Critères d'évaluation non supervisés

Les critères d'évaluation non supervisés [Pakhira et al., 2004] se basent sur des informations internes au clustering comme par exemple la distance entre les objets d'un cluster et le centroïde de celui-ci. Ces mesures se basent souvent sur la définition la plus simple du clustering qui définit que les objets d'un même cluster doivent être les plus proches possible entre eux et que les objets de deux clusters distincts doivent être les plus éloignés possible. Pour évaluer si un clustering respecte cette définition intuitive, des mesures de distance sont calculées entre les représentants des clusters et les objets du résultat. Ces mesures non supervisées permettent d'évaluer la compacité ainsi que la séparabilité des clusters. La définition de la qualité d'un cluster n'étant pas définie formellement, il existe de nombreux critères évaluant de manière différente les résultats. Certains de ces critères peuvent être directement utilisés comme fonction objective et être optimisés par un algorithme de clustering. D'autres sont cependant trop coûteux à évaluer pour être calculés au cours de l'exécution d'un algorithme et sont par conséquent destinés à être calculés à l'issue de l'application de celui-ci. Nous présentons dans la suite les mesures d'évaluation les plus connues.

Somme des erreurs au carré (*SSE*) :

La somme des erreurs au carré est une des façons la plus simple d'évaluer la qualité d'un résultat. Elle est définie comme :

$$SSE(\mathcal{C}) = \sum_{i=1}^{K} \sum_{x \in C_i} d(x - \mu_i)^2 \tag{2.4}$$

avec μ_i le centroïde du cluster C_i, et d une mesure de distance entre les objets. Plus la valeur est petite plus les clusters sont compacts.

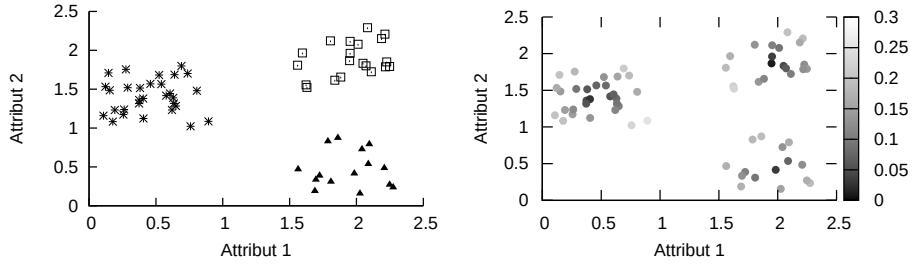
Coefficient silhouette (*CS*) :

Le coefficient silhouette [Kaufman et Rousseeuw, 1990] permet d'évaluer la compacité des clusters ainsi que la séparabilité de ceux-ci. Il peut être calculé pour chaque objet, pour chaque cluster et pour le clustering entier. Pour un objet x il est défini comme :

$$CS(x) = \frac{b_x - a_x}{max(a_x, b_x)}$$

avec a_x la distance moyenne entre l'objet x et tous les autres objets appartenant au même cluster que x, et b_x la distance moyenne entre x et tous les objets n'appartenant pas à ce même cluster. Le coefficient $CS(x)$ varie entre -1 et 1. Une valeur positive ($a_x < b_x$) signifie que les objets appartenant au même cluster que x sont plus proches de x que des objets des autres groupes. Pour un cluster, le coefficient silhouette est la moyenne des coefficients des objets appartenant à ce cluster :

$$CS(C_i) = \frac{1}{|C_i|} \sum_{x \in C_i} (CS(x))$$



(a) Clustering à trois clusters. (b) Coefficient silhouette pour chaque objet.

Fig. 2.5: *Illustration du calcul du coefficient silhouette pour chaque objet d'un clustering.*

Enfin, pour un clustering, le coefficient silhouette est égal à la moyenne des coefficients de ses clusters :

$$CS(\mathcal{C}) = \frac{1}{K}\sum_{i=1}^{K}(CS(C_i)) \tag{2.5}$$

Le coefficient pour le clustering varie également entre -1 et 1, une valeur positive indiquant que les clusters sont très compacts et bien séparés. Il est à noter que le calcul de cet indice est relativement coûteux en temps car de nombreux calculs de distance sont nécessaires à son évaluation. La figure 2.5 présente un exemple du calcul du coefficient silhouette pour chaque objet d'un clustering à trois clusters.

Indice de Dunn (*Du*) :

L'indice de Dunn [Dunn, 1974] permet d'évaluer la séparabilité ainsi que la compacité des clusters. Il est défini par :

$$Du(\mathcal{C}) = \frac{\min\limits_{i=1,\ldots,K}\left(\min\limits_{j=1,\ldots,K,i\neq j}\Big(D_s(C_i,C_j)\Big)\right)}{\max\limits_{i=1,\ldots,K}\Big(D_c(C_i,C_i)\Big)} \tag{2.6}$$

où $D_s(C_i, C_j)$ correspond à la distance entre les clusters C_i et C_j, définie ici comme la plus petite distance entre les objets des deux clusters (voir équation (2.1)) et $D_c(C_i, C_i)$ correspond à la distance maximale entre deux objets du cluster (voir équation (2.2)). Une faible valeur indique une forte compacité et une forte séparation des clusters.

Indice de Davies et Bouldin (*DB*) :

L'indice de Davies et Bouldin [Davies et Bouldin, 1979] mesure la compacité et la séparation des clusters en se basant sur le calcul de la moyenne de la similarité entre les clusters. Il se définit comme :

$$DB(\mathcal{C}) = \frac{1}{K}\sum_{i=1}^{K}\max_{j=1,\ldots,K,i\neq j}\left(\frac{S(C_i)+S(C_j)}{d(\mu_i,\mu_j)}\right) \tag{2.7}$$

où $d(\mu_i, \mu_j)$ représente la distance entre les centroïdes de C_i et C_j et $S(C_i)$ la distance moyenne entre chaque objet de C_i et son centroïde μ_i :

$$S(C_i) = \frac{1}{|C_i|} \sum_{x \in C_i} d(x, \mu_i)$$

Plus les clusters sont compacts et plus la moyenne de la distance au centroïde ($S(C_i)$) est petite. Plus les clusters sont séparés et plus la distance entre les groupes ($d(\mu_i, \mu_j)$) est élevée. Une valeur faible de l'indice de Davies et Bouldin indique un clustering de bonne qualité.

Indice de Wemmert et Gançarski (*WG*) :

L'indice de compacité de Wemmert et Gançarski [Wemmert, 2000] évalue la séparabilité et la compacité des clusters. Pour un cluster, il est défini par :

$$WG(C_i) = \begin{cases} 0 \quad \text{si } \frac{1}{|C_i|} \sum\limits_{x \in C_i} \frac{d(x, \mu_i)}{d(x, \mu_j)} > 1 \\ 1 - \frac{1}{|C_i|} \sum\limits_{x \in C_i} \frac{d(x, \mu_i)}{d(x, \mu_j)} \quad \text{sinon} \end{cases}$$

où $j = \arg\min\limits_{k \neq i} (dist(x, \mu_k))$

L'indice de Wemmert et Gancarski prend ses valeurs entre 0 et 1, 1 indiquant une très bonne compacité et séparabilité des clusters. Pour un clustering, il est défini par :

$$WG(\mathcal{C}) = \frac{1}{n} \sum_{i=1}^{K} |C_i| WG(C_i) \tag{2.8}$$

2.3.3 Bilan

Nous avons présenté dans cette section différents indices permettant l'évaluation de résultats de clustering de manière non supervisée. Ces critères sont dit internes car ils n'utilisent pas d'informations externes au clustering pour son évaluation. Plusieurs critères existent car plusieurs approches différentes peuvent être envisagées pour juger de la qualité d'un partitionnement. Un algorithme dont la fonction objective est définie par l'un de ces critères aura potentiellement plus de chance de produire de bons résultats pour cette évaluation, mais pas forcément par rapport aux autres critères. L'utilisation de plusieurs critères est donc recommandé lorsque des évaluations non supervisées sont effectuées. Pour observer si un algorithme optimise bien son critère objectif, il est possible d'étudier la progression de celui-ci. Cependant, si l'on souhaite une évaluation moins biaisée, il est courant d'utiliser des critères qui ne sont pas directement optimisés par la méthode. Ceci permet de s'assurer que la solution produite est de bonne qualité d'après différents critères (compacité des clusters, séparabilité des clusters, etc.).

2.4 Problèmes et limites du clustering

Malgré l'existence d'un grand nombre de méthodes de clustering ainsi que leur utilisation avec succès dans de nombreux domaines, le clustering pose encore de nombreux problèmes. Ces problèmes sont liés d'une part au manque de précision dans la définition de ce qu'est réellement un cluster mais également dans la difficulté de définir une mesure de similarité entre objets ou encore dans la définition d'une fonction objective pour un problème donné. Jain et Dubes [1988] ont listé un ensemble de questions qu'il est nécessaire de se poser lorsqu'on entreprend d'effectuer une tâche de clustering. Cette liste de questions met en avant la multiplicité et surtout la nature différente des paramètres à prendre en compte dans ce type d'approche :

– Qu'est-ce qu'un cluster ?
– Quels attributs doivent être utilisés ?
– Les données doivent-elles être normalisées ?
– Les données contiennent-elles des objets atypiques ?
– Comment est définie la similarité entre deux objets ?
– Combien de clusters sont présents ?
– Quelle méthode de clustering doit-on utiliser ?
– Est-ce que les données contiennent des clusters ?
– Est-ce que la partition découverte est valide ?

Nous allons étudier dans les sections suivantes certains des problèmes et limites récurrents en clustering qui sont soulevés par ces questions.

2.4.1 Choix du nombre de clusters

Définir le nombre de clusters est un des problèmes les plus difficiles en clustering. En effet, il est souvent nécessaire de fournir le nombre de clusters souhaité comme paramètre de l'algorithme. Le choix du nombre de clusters a souvent été étudié comme un problème de sélection de modèle. Dans ce cas, l'algorithme est généralement exécuté plusieurs fois indépendamment avec un nombre de clusters différent. Les résultats sont ensuite comparés en se basant sur un critère de sélection qui permet de choisir la meilleure solution. Ce choix est toujours subjectif et fortement dépendant du critère sélectionné pour comparer les résultats.

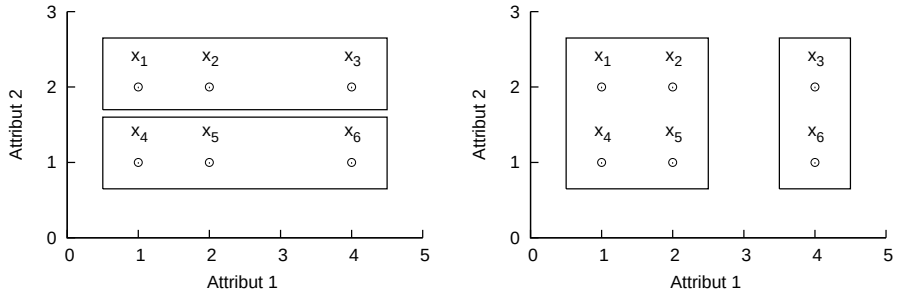
Deux approches moins subjective souvent utilisées se basent sur les critères de Minimum Message Length (MML) [Figueiredo et Jain, 2000] et Minimum Description Length (MDL) [Hansen et Yu, 1998]. Elles consistent à débuter avec un nombre de clusters relativement élevé, puis à fusionner itérativement deux clusters pour optimiser les critères (MML ou MDL). Les autres critères classiquement utilisés pour la sélection de modèle sont le Bayes Information Criterion (BIC) et le Akiake Information Criterion (AIC). Le Gap statistics [Tabachnick et Fidell, 2006] est également utilisé pour décider du nombre de clusters. Ces critères reposent généralement sur des bases statistiques fortes et s'appliquent de manière naturelle aux méthodes de clustering probabilistes (voir section 2.2.2.3). Elles peuvent être plus difficiles à mettre en place lors de l'utilisation d'autres types d'approches. De plus, elles sont relativement coûteuses et nécessitent d'effectuer de nombreuses exécutions des algorithmes. L'étude de la validité des clusters découverts peut également être un outil pour effectuer le choix du nombre de clusters. Dans l'idéal, le choix du nombre de clusters reste à l'appréciation de l'expert qui est à même, avec ou sans l'aide d'indices, de choisir le nombre de clusters qui lui paraît adapté [Xu et Wunsch, 2005].

L'algorithme ISODATA [Ball et Hall, 1965] est basé sur l'algorithme KMEANS mais permet une évolution du nombre de clusters pendant l'exécution. Des clusters sont fusionnés si le nombre d'objets dans un des clusters est inférieur à un seuil ou si deux clusters sont plus proches qu'un certain seuil. De plus, les clusters peuvent également être séparés en deux si leur dispersion (évaluée par exemple par l'écart type) dépasse un seuil. Cet algorithme permet, contrairement à KMEANS, d'avoir un nombre de clusters variables pendant l'exécution. Cependant, ce type d'approche nécessite la définition de nombreux seuils et ne fait souvent que déplacer le problème du choix du nombre de clusters à la définition de ces seuils.

2.4.2 Validité des clusters

Un algorithme de clustering tente de découvrir des clusters dans les données que celles-ci en contiennent ou non. Dans un cas idéal, il faudrait analyser les données a priori pour savoir si celles-ci ont une tendance à posséder des clusters avant d'appliquer un algorithme de recherche de clusters. Dans les faits, il est nécessaire d'étudier les résultats obtenus pour s'assurer de la pertinence des clusters identifiés. Vérifier la validité des clusters découverts consiste à utiliser une procédure qui évalue le résultat obtenu de manière quantitative et objective [Jain et Dubes, 1988].

Pour étudier la validité des clusters découverts, les critères présentés dans la section 2.3 peuvent



(a) Premier clustering possible. (b) Second clustering possible.

Fig. 2.6: *Exemple de données produisant des résultats différents suivant l'initialisation de* KMEANS.

être utilisés. Ils permettent d'évaluer la qualité des clusters selon certaines caractéristiques qui donnent des indications sur ceux-ci (compacité, séparabilité, etc.). Ces critères peuvent également servir de guide pour choisir le nombre de clusters idéal pour le jeux de données selon un critère donné.

La notion de stabilité des clusters [Lange et al., 2004] peut également être utilisée comme un critère de qualité et de validité. La stabilité des clusters peut être définie comme la variation entre deux résultats de clustering produits à partir de différents échantillonnages des données. Différentes mesures de variations peuvent être utilisées pour obtenir une évaluation de la stabilité. Par exemple, dans les algorithmes à base de modèle (les centroïdes pour KMEANS, ou les mixtures de gausiennes pour EM) la distance entre deux modèles trouvés sur différents échantillons peut être utilisée comme mesure de stabilité.

La notion de validation croisée utilisée en apprentissage supervisé peut également être utilisée pour étudier la stabilité des solutions. Elle a été adaptée en classification non supervisée en remplaçant la notion de précision par un critère de validité (par exemple les critères vus dans la section 2.3). Dans le cas de l'algorithme EM, les mixtures de gausiennes apprises sur un sous-ensemble peuvent être évaluées en étudiant leur vraisemblance sur les autres sous-ensembles. Cette information peut également être utilisée comme une indication pour sélectionner le nombre de clusters.

2.4.3 Paramétrage des algorithmes

Chaque algorithme de clustering possède un certain nombre de paramètres. Ces paramètres influent de manière plus au moins importante sur les résultats obtenus. Ces paramètres sont nécessaires pour que l'algorithme soit relativement générique et pour que celui-ci soit applicable dans plusieurs cas. Plus le nombre de paramètres augmente plus l'algorithme est adaptable et peut s'appliquer à une gamme de problèmes plus large. Cependant, l'augmentation des paramètres nécessite également de la part de l'expert une connaissance importante sur ses données et sur le fonctionnement de l'algorithme. Nous avons déjà soulevé le problème du nombre de clusters comme paramètre de nombreux algorithmes. Il existe bien d'autres paramètres qu'il serait impossible de citer avec exhaustivité. On pourra cependant noter le problème de l'initialisation qui est un problème important et qui concerne une grande partie des algorithmes existants. Prenons l'exemple de l'algorithme KMEANS où il est nécessaire de définir les centroïdes initiaux. En fonction du choix effectué pour cette initialisation, les résultats peuvent être fortement différents. Pour réduire ce problème, il est possible d'étudier les *formes fortes* ce qui consiste à identifier les groupes d'objets qui sont régulièrement regroupés ensemble lors de plusieurs exécutions différentes de l'algorithme. Il est cependant difficile de choisir comment faire varier les paramètres et combien d'exécutions sont nécessaires.

Prenons un exemple simple pour illustrer ce problème d'initialisation [Manning et al., 2008]. Sur la figure 2.6, six objets sont représentés et nous allons les regrouper en 2 clusters en util-

isant l'algorithme KMEANS. Si à l'initialisation les objets x_1 et x_5 sont sélectionnés comme centres initiaux, l'algorithme convergera vers la solution $\{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\}$ (figure 2.6(a)) qui n'est pas une solution optimale de la fonction objective de KMEANS (SSE). Si par contre les objets x_2 et x_3 sont sélectionnés comme centres initiaux, l'algorithme convergera vers la solution $\{\{x_1, x_2, x_4, x_5\}, \{x_3, x_6\}\}$ (figure 2.6(b)) qui est la solution optimale de la fonction objective de KMEANS (SSE) pour $K = 2$.

2.4.4 Comparaison des méthodes de clustering

Il est relativement difficile de faire un choix parmi de nombreuses méthodes de clustering, surtout quand l'expert a peu de connaissances sur ses données, ce qui est généralement le cas en classification non supervisée. Les critères d'évaluation de la qualité ou de la stabilité présentés précédemment apportent des éléments permettant d'aider l'expert à faire un choix.

Pour tenter d'apporter des réponses à ce problème, Jain et al. [2004] ont comparé des algorithmes entre eux par rapport aux résultats qu'ils produisent. Pour se faire, les auteurs ont utilisé 35 algorithmes de clustering différents sur 12 jeux de données. Les résultats produits ont ensuite été comparés, et les méthodes de clustering ont été regroupées en fonction de la similarité des résultats produits. Cette étude, qui ne donne pas de réponse à la question : *"Quel est le meilleur algorithme ?"*, permet tout de même d'obtenir des informations sur les méthodes ayant sensiblement le même comportement.

Des études plus théoriques d'algorithmes de clustering ont également été menées en tentant de comparer les fonctions objectives des algorithmes. D'après Jain [2009], il est nécessaire de faire la différence entre méthode de clustering et algorithme de clustering. Une méthode de clustering est une stratégie générale employée pour résoudre un problème de clustering. Un algorithme de clustering est alors une instance d'une de ces méthodes.

Toutes ces études arrivent cependant à la même conclusion : il n'existe pas de méthode de clustering meilleure que toutes les autres. Cette idée est notamment formalisée par Kleinberg [2002] qui montre qu'aucun algorithme de clustering ne satisfait de manière simultanée un ensemble d'axiomes de base. Van Ness [1973] a également défini un ensemble de critères auxquels est censé répondre un algorithme de clustering parfait, et a montré que certains de ces critères sont contradictoires et que par conséquent, il n'est pas possible de les satisfaire tous en même temps.

2.5 Bilan

Le clustering est une tâche dont l'objectif est de trouver des groupes au sein d'un ensemble d'objets. Dans ce chapitre, nous avons étudié les grands concepts du clustering, les principales méthodes existantes ainsi que leur évaluation et leur comparaison. Nous avons également présenté les nombreux problèmes et limites en clustering. Il en ressort qu'un nombre important de méthodes existent et qu'il est souvent difficile de faire un choix parmi celles-ci. Ce choix est crucial dans le processus de fouille de données et est conditionné par le type de résultat que l'expert veut obtenir. Pour tenter de résoudre ce problème, nous allons voir dans le chapitre suivant comment plusieurs algorithmes de clustering peuvent être utilisés de manière simultanée pour produire de meilleurs résultats. Ainsi, nous proposons d'utiliser le paradigme du clustering collaboratif pour tirer parti des informations provenant de plusieurs algorithmes de clustering différents.

Chapitre 3

Le clustering collaboratif

Sommaire

3.1 Introduction

Comme nous l'avons vu dans le chapitre précédent, il existe de nombreuses méthodes de clustering. Celles-ci peuvent, à partir des mêmes données, fournir des résultats différents. Cette multitude de méthodes impose à l'expert de choisir une méthode ainsi que ses paramètres. Ce choix est lourd de conséquences et va conditionner le type de résultats qu'il pourra obtenir. Une connaissance importante du problème est nécessaire pour choisir une méthode de clustering de manière éclairée. Bien sûr, si l'expert n'a aucune connaissance sur ses données et sur le fonctionnement des algorithmes, il lui sera difficile de faire un choix sensé. Cependant, même avec une expertise sur les données et les algorithmes, il est bien souvent difficile de faire le *bon* choix.

Pour tenter de résoudre ce problème, la communauté scientifique s'est intéressée depuis plusieurs années à la combinaison de plusieurs méthodes de clustering. L'objectif est d'utiliser l'avis de plusieurs méthodes dans le but d'en dégager un consensus ou une synthèse. Ces approches se basent sur l'intuition que combiner des informations fournies par différents acteurs, ou experts, peut améliorer la solution proposée à un problème. Le marquis de Condorcet, un philosophe, mathématicien et politologue français a été l'un des premiers à formaliser cette notion dans *Essai*

sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. Il décrit, ce qui sera par la suite appelé le théorème du jury de Condorcet. Dans ce théorème, il formalise la probabilité relative d'un groupe d'individus à arriver à une solution correcte. La version la plus simple du théorème stipule que le vote à la majorité d'un ensemble d'individus qui se prononcent indépendamment sur un problème à deux issues permet de réduire le risque d'erreur si ces individus font un choix correct avec une probabilité d'au moins 1/2. Dans ce cas, plus le nombre d'individus votant augmente, plus la probabilité de faire une erreur de jugement en utilisant un vote diminue, jusqu'à devenir nulle pour une infinité de votants. Cependant, si la probabilité de se tromper parmi les votants est potentiellement supérieure à 1/2, alors l'ajout d'individus votant ne fait qu'augmenter la probabilité d'erreur. Le tableau 3.1 montre un exemple de ce type de vote [Kuncheva, 2008].

Voici les notations utilisées dans les prochaines sections :

Notations :

- Soit $X = \{x_1, \ldots, x_n\}$ l'ensemble des n objets à classer ;
- Soit $\mathbb{C} = \{\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(N)}\}$ un ensemble de N résultats de clustering de X ;
- Soit $\mathcal{C}^{(i)} = \{C_1^{(i)}, \ldots, C_{K^{(i)}}^{(i)}\}$ un résultat de clustering de X en $K^{(i)}$ clusters ;
- Soit $n_k^{(i)} = |C_k^{(i)}|$ le nombre d'objets du cluster $C_k^{(i)}$;
- Soit $\alpha_{k,l}^{(i,j)} = |C_k^{(i)} \cap C_l^{(j)}|$ le nombre d'objets commun au cluster $C_k^{(i)}$ et au cluster $C_l^{(j)}$.

3.1.1 Approches supervisées

En classification supervisée, le théorème de Condorcet a pu facilement être utilisé et adapté, le domaine de définition de la variable cible, c'est-à-dire l'ensemble des classes, étant fixé, connu et commun à tous les classifieurs. Il est alors relativement simple de créer plusieurs modèles différents et de les combiner suivant leur prédiction. De nombreuses méthodes de combinaison ont été proposées [Kittler et al., 1998], ce domaine étant communément appelé classification par ensemble ou classification par méthode ensembliste. Ces méthodes de combinaison supervisées peuvent se contenter de fusionner les décisions de plusieurs classifieurs, mais certaines proposent également de modifier la manière dont l'algorithme de classification supervisée est appliqué (peu importe le choix de celui-ci). Ainsi, la méthode du STACKING [Wolpert, 1992] cherche à construire un méta-modèle en utilisant plusieurs méthodes de classification différentes. De cette manière, le STACKING tente de réduire le biais induit par la sélection d'une unique méthode. La méthode du BAGGING [Breiman, 1996] consiste à utiliser plusieurs échantillonnages des données et construire plusieurs classifieurs, dit faibles, c'est-à-dire que leur prédiction n'est que légèrement supérieure à l'aléatoire. Ces classifieurs faibles sont combinés par un méta-classifieur dit fort, qui va combiner les prédictions souvent en utilisant un vote [Kittler et al., 1998]. Enfin, la méthode de BOOSTING [Schapire, 1990] fonctionne de manière similaire mais pondère les classifieurs faibles lors de la combinaison suivant leur précision de classification ou un autre critère (par exemple une connaissance externe sur la qualité des sources). De plus, le choix des échantillons est dirigé afin de sélectionner les exemples difficiles à classer. De cette manière, l'algorithme de classification va se concentrer sur les exemples posant le plus de problèmes.

3.1.2 Approches non supervisées

Dans le cas non supervisé, ce type d'approches par combinaison est plus compliqué à mettre en place. En effet, lorsque l'on va générer plusieurs résultats de clustering à partir d'un même jeu de données, on disposera pour chaque objet uniquement de l'information de son appartenance aux clusters de chaque résultat. Comme il n'existe pas de lien évident entre les clusters d'un résultat et les clusters d'un autre résultat de clustering, il est difficile de comparer leur prédiction. Soient deux résultats de clustering $\{C_1^{(1)}, C_2^{(1)}, \ldots, C_{K^{(1)}}^{(1)}\}$ et $\{C_1^{(2)}, C_2^{(2)}, \ldots, C_{K^{(2)}}^{(2)}\}$, rien ne garantit que les deux clusters $C_k^{(.)}$ de ces deux résultats représentent sensiblement les mêmes groupes d'objets ($C_1^{(1)} \cap C_1^{(2)} = \varnothing$). De plus, les deux résultats peuvent tout à fait avoir un nombre de clusters différent ($K^{(1)} \neq K^{(2)}$).

Classifieur	Prédiction	Précision
classifieur 1	□■□■■□□□□■□□□□■	10/15 = 0.667
classifieur 2	■■□□□□■□□□□□■□■	10/15 = 0.667
classifieur 3	□□■□□■■□□□□□■■□	10/15 = 0.667
vote	□■□□□□■□□□□□■□■	11/15 = 0.733

□ = correct, ■ = erreur.

Tab. 3.1: *Exemple d'amélioration d'un résultat par l'utilisation d'un vote à la majorité en classification supervisée.*

Pour résoudre le fait que les clusters de différents résultats n'ont pas de lien direct, différentes méthodes permettant en partie de s'affranchir de cette correspondance ont été proposées dans la littérature. Les méthodes dites d'*ensemble clustering*, qui seront présentées en détail dans la section 3.2.1, tentent de trouver la partition *consensus*, résumant au mieux un ensemble de partitions donné. Seule la distribution des objets dans les clusters de chaque résultat est disponible. Les méthodes d'ensemble clustering s'intéressent uniquement à la construction de cette partition consensuelle, ne modifient pas les partitions données en entrée, et n'en créent pas de nouvelles. Par conséquent, pour que les méthodes soient réellement efficaces, les partitions fournies en entrée doivent être de qualité et de complémentarité suffisantes.

De manière sensiblement différente, les approches multiobjectives présentées dans la section 3.2.2, considèrent le processus de clustering comme l'optimisation de plusieurs objectifs. Dans ces méthodes, un nombre important de résultats de clustering sont explorés, et ceux qui répondent le mieux aux différents objectifs sont conservés. Un algorithme génétique est généralement utilisé, et de nouveaux résultats de clustering sont créés en mélangeant les différents résultats disponibles. Cependant, les objectifs utilisés par les méthodes d'optimisation ne sont pas toujours ceux utilisés pour créer les partitions initiales, ce qui introduit un nouveau biais.

Dans la collaboration de méthodes floues, présentée en détail dans la section 3.2.3, l'algorithme FUZZY-C-MEANS [Dunn, 1974] est utilisé pour créer des clusters de différentes vues des données de manière collaborative. Ces travaux [Pedrycz, 2002] se concentrent uniquement sur l'algorithme FUZZY-C-MEANS ce qui limite son utilisation. Une autre méthode a récemment été proposée, basée sur l'algorithme SOM [Grozavu et Bennani, 2010].

Dans la section 3.3 nous étudions la méthode de clustering collaboratif développée par Wemmert et Gançarski [Wemmert, 2000]. Nous présentons le fonctionnement de la méthode et nous proposons des exemples pour illustrer son fonctionnement. Enfin, nous détaillons les solutions proposées lors du présent travail de thèse sous la forme d'une nouvelle méthode de clustering collaboratif. Des évaluations sont présentées pour illustrer la pertinence de cette nouvelle approche.

3.2 Combinaison de plusieurs méthodes de clustering

3.2.1 Approches par ensemble

3.2.1.1 Applications

Le but du clustering par ensemble est de produire un unique clustering à partir d'un ensemble de plusieurs clusterings donné. Les approches par ensemble ont été utilisées dans de nombreux domaines, nous allons étudier les principales applications recensées dans la littérature [Punera et Ghosh, 2007].

Une des applications du clustering par ensemble est la réutilisation de connaissances acquises précédemment sous la forme de partitions existantes. Par exemple, lorsque des données sont re-

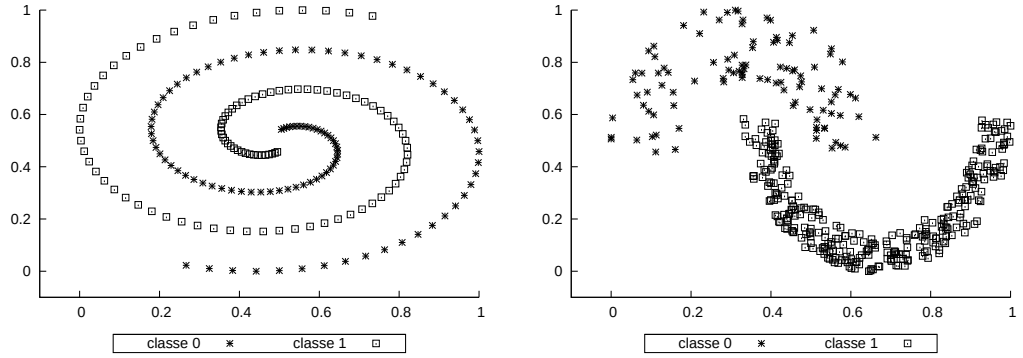


Fig. 3.1: *Exemples de jeux de données avec des classes de formes spécifiques.*

classées, un clustering antérieur peut être utilisé comme une connaissance disponible. De plus, les informations utilisées pour créer cet ancien résultat ne sont peut-être plus disponibles, les données ayant évoluées, ou la partition ayant été produite par un expert humain [Strehl et Ghosh, 2002].

Une autre application s'intéresse à la classification multivue. Un objet peut être décrit par différents attributs qui ne sont pas forcément compatibles et il peut être impossible de les utiliser dans un processus de clustering unique. Ces objets sont classés en fonction des différents sous-ensembles d'attributs, et les résultats obtenus sont combinés grâce à une méthode par ensemble [Kreiger et Green, 1999]. Par exemple, dans des domaines comme le clustering d'images ou de documents web basé sur des annotations fournies par des experts, le fait que plusieurs sources d'informations possède chacune son propre jeu d'annotations rend souvent difficile l'application d'un processus de clustering unique (les étiquettes des classes pouvant être différentes pour chaque expert). Le chapitre 6 abordera ces concepts en détail.

Récemment, les méthodes de clustering par ensemble ont trouvé des applications dans le cadre de la préservation de la vie privée (*privacy-preserving*) [Aggarwal et Yu, 2008] et en fouille de données distribuées [Merugu et Ghosh, 2003]. En effet, il n'est pas toujours possible de centraliser toutes les données sur un site unique pour effectuer le processus de classification. On pourra par exemple citer le cas d'entreprises possédant des données commerciales sur leurs activités. Chacune de ces entreprises possèdent des données propres et celles-ci ne sont pas forcément prêtes à les partager avec les autres acteurs du marché. Il est cependant envisageable d'effectuer des clusterings indépendants sur chacune de ces données, et d'utiliser une approche par ensemble pour obtenir une solution finale (il faut évidement que celles-ci partagent une liste d'objets commune comme par exemple une liste de clients).

Les méthodes par ensemble ont pour objectif l'amélioration de la qualité des résultats en réduisant le biais induit par chaque algorithme, et donc d'améliorer la qualité du résultat final [Hadjitodorov et Kuncheva, 2007; Hadjitodorov et al., 2006]. Certaines méthodes permettent de trouver des clusters de forme spécifique comme des spirales (voir figure 3.1). En effet, utiliser plusieurs clusterings permet à certaines méthodes par ensemble de mieux saisir la distribution spatiale des données.

L'utilisation de plusieurs résultats de clustering augmente aussi la robustesse des solutions. De nombreux algorithmes souffrent de problèmes d'initialisation et tombent dans des minima locaux de leur fonction objective. Le clustering par ensemble apporte une solution à ces problèmes en utilisant des résultats issus d'un même algorithme mais initialisé à chaque fois différemment [Fred et Jain, 2005; Fred, 2001].

3.2.1.2 Problématique

La classification par ensemble peut être formellement définie comme la recherche d'un résultat de clustering $\mathcal{C}^{(*)}$ représentant au mieux la structure des données X, à partir de l'ensemble des informations disponibles dans les N résultats de $\mathbb{C}$. Pour cela, il est nécessaire de définir une

fonction Φ prenant en entrée les N résultats de $\mathbb{C}$ et produisant un résultat de clustering $\mathcal{C}^{(*)}$ appelé communément *consensus* :

$$\Phi : \mathbb{C} \rightarrow \mathcal{C}^{(*)} \tag{3.1}$$

Le résultat $\mathcal{C}^{(*)}$ peut être vu comme une moyenne des résultats de l'ensemble, et donc comme étant le clustering le plus similaire (selon une fonction de similarité sim) à l'ensemble des N résultats de $\mathbb{C}$ parmi l'ensemble des résultats de clustering possibles $\breve{\mathbb{C}} = \{\breve{\mathcal{C}}^{(1)}, \breve{\mathcal{C}}^{(2)}, \ldots, \breve{\mathcal{C}}^{(m)}\}$ de X :

$$\mathcal{C}^{(*)} = \arg \max_{\breve{\mathcal{C}}^{(i)} \in \breve{\mathbb{C}}} \sum_{q=1}^{N} sim(\breve{\mathcal{C}}^{(i)}, \mathcal{C}^{(q)}) \tag{3.2}$$

Il est cependant impossible d'effectuer une recherche exhaustive parmi tous les résultats possibles $\breve{\mathbb{C}} = \{\breve{\mathcal{C}}^{(1)}, \breve{\mathcal{C}}^{(2)}, \ldots, \breve{\mathcal{C}}^{(m)}\}$ leur nombre étant dissuasif : $m = \frac{1}{K!} \sum_{k=1}^{K} \binom{K}{k} (-1)^{K-k} k^n$ (par exemple 171 798 901 possibilités de former 4 groupes à partir de 16 objets). De plus, quand ce problème est vu sous la forme d'un problème d'optimisation, la recherche d'un consensus devient un problème NP-complet [Megiddo et Supowit, 1984] car se rapportant à un problème de coloration de graphe.

Les travaux sur le clustering par ensemble s'intéressent essentiellement à deux aspects. Le premier est l'étude des méthodes permettant de générer les différents résultats composant $\mathbb{C}$. Le second est la définition de la fonction Φ permettant de générer le consensus. Dans les sections suivantes, nous verrons tout d'abord les principaux domaines d'application de la classification par ensemble. Nous verrons ensuite les principales méthodes pour générer l'ensemble $\mathbb{C}$ des résultats. Enfin les différentes approches abordées dans la littérature pour définir la fonction Φ permettant de trouver un consensus seront présentées.

3.2.1.3 Génération des résultats initiaux

Il existe différentes stratégies permettant de générer les résultats de clustering initiaux. Pour que le résultat final soit pertinent, il est nécessaire d'introduire une certaine diversité au sein des résultats initiaux. En effet, si tous les résultats de l'ensemble sont identiques, les combiner ne permettra pas d'améliorer le résultat final. Plusieurs heuristiques ont été proposées pour assurer cette diversité :

- utiliser différents algorithmes de clustering ;
- utiliser différentes initialisations d'un même algorithme ;
- utiliser différents sous-ensembles d'instances par échantillonnage ;
- utiliser différents sous-ensembles d'attributs ;
- utiliser des projections dans des sous-espaces différents.

Une description détaillée des différentes heuristiques permettant de générer ces résultats initiaux ainsi que leurs intérêts ont été donnés dans [Hadjitodorov et al., 2006] et [Hadjitodorov et Kuncheva, 2007]. Il a été montré qu'il est important d'obtenir une certaine diversité au sein de l'ensemble notamment en terme de nombre de clusters. En effet, combiner des résultats fortement corrélés réduit les possibilités d'une amélioration, le consensus ne pouvant être alors de qualité supérieure aux résultats de l'ensemble. Cependant ces approches nécessitent le calcul d'un nombre important de résultats pour être efficace, c'est-à-dire plusieurs dizaines voire plusieurs centaines. Si le jeu de données est volumineux, ces approches sont donc difficilement applicables.

Les techniques d'échantillonnage (*resampling*) sont souvent utilisées comme dans [Minaei-Bidgoli et al., 2004a,b]. L'intérêt de combiner des partitions créées par des clusterings *faibles* (*weak clustering*) a également été montré par Topchy et al. [2005]. Ces clusterings sont produits en utilisant des heuristiques simples sur les données comme par exemple une projection sur un axe aléatoire.

3.2.1.4 Méthodes de combinaison basées sur les graphes

L'une des premières approches pour le clustering par ensemble a été proposée par Strehl et Ghosh [2002] qui introduisent différents cadres d'application de ces méthodes et décrivent trois algorithmes permettant de calculer un consensus à partir d'un ensemble de résultats. La recherche d'un consensus est définie comme la recherche d'un résultat $\mathcal{C}^{(*)}$ qui partage le plus d'information avec les résultats de l'ensemble $\mathbb{C}$. Pour quantifier ce partage d'information, les auteurs utilisent la notion d'information mutuelle (*mutual information*), qui est une mesure symétrique pour quantifier statistiquement l'information partagée par deux distributions [Cover et Thomas, 2006]. L'objectif du clustering par ensemble tel que défini par Strehl et Ghosh [2002] est de trouver le consensus maximisant l'information mutuelle normalisée (NMI) avec les membres de l'ensemble :

$$\mathcal{C}^{(*)} = \arg\max_{\breve{\mathcal{C}}^{(i)} \in \breve{\mathbb{C}}} \sum_{q=1}^{N} NMI(\breve{\mathcal{C}}^{(i)}, \mathcal{C}^{(q)}) \tag{3.3}$$

Pour trouver ce consensus, les trois algorithmes proposés utilisent la notion d'hypergraphe pour représenter l'ensemble des résultats de clustering. Un hypergraphe est un ensemble de sommets et d'hyper-arêtes, une hyper-arête étant une généralisation du concept d'arête pouvant être connectée à un ensemble de sommets. Pour chaque résultat $\mathcal{C}^{(i)} \in \mathbb{C}$, une matrice binaire d'appartenance $H^{(i)}$ est créée, composée d'une colonne pour chaque cluster du résultat (voir exemple tableau 3.2). La concaténation de l'ensemble des matrices $H = (H^{(1)} \dots H^{(N)})$ représente la matrice d'adjacence d'un hypergraphe à N sommets et $\sum_{i=1}^{N} K^{(i)}$ hyper-arêtes. Chaque colonne h_i définit une hyper-arête où 1 indique que cette hyper-arête contient le sommet et 0 qu'elle ne le contient pas. Les trois algorithmes proposés utilisent cette représentation.

La première méthode appelée CSPA (*Cluster-based Similarity Partioning Algorithm*), est basée sur la création d'une mesure de similarité entre les objets à partir des informations contenues dans les résultats de l'ensemble. Pour représenter cette similarité, une matrice S de taille $n \times n$ (n étant le nombre d'objets) est calculée à partir de la matrice H tel que $S = \frac{1}{N} H H^T$. Cette matrice est ensuite utilisée dans un algorithme de clustering classique pour générer le consensus.

Le seconde approche appelée HPGA (*HyperGraph Partitioning Algorithm*), utilise directement la méthode de partitionnement d'hypergraphe HMETIS [Karypis et al., 1997] pour partitionner l'hypergraphe H représentant l'ensemble des clusterings. L'objectif de cette méthode est de créer un clustering consensus qui coupe le moins d'hyper-arêtes.

Enfin, la troisième approche appelée MCLA (*Meta-CLustering Algorithm*) crée des méta-groupes au sein de l'hypergraphe des clusterings en fusionnant des clusters similaires. Ces méta-groupes sont ensuite utilisés pour déterminer le clustering final. La similarité entre deux clusters est calculée à partir du nombre d'instances qui sont regroupées ensemble dans ces deux clusters en utilisant l'indice de Jaccard (voir Annexe B). Le graphe où les clusters ont été fusionnés est ensuite partitionné en utilisant la méthode de partitionnement d'hypergraphe HMETIS. Un vecteur d'association entre les instances et les clusters est créé au sein de chaque méta-groupe. Les instances sont ensuite classées dans le meta-cluster ayant le plus fort degré d'association.

Dans une autre approche proposée par Fern et Brodley [2004], nommée HBGF (*Hybrid Bipartite Graph Formulation*), le problème est ramené à trouver une partition d'un graphe bipartite pour la formation du consensus. L'algorithme CSPA modélise l'ensemble comme un graphe dont les sommets représentent les instances, alors que l'algorithme MCLA modélise l'ensemble comme un graphe de clusters. L'algorithme HBGF combine ces deux approches et représente l'ensemble par un graphe bipartite où les instances ainsi que les clusters forment des sommets. Le graphe est bipartite car il n'y a pas d'arête entre les deux différents types de sommets (instance et cluster). Des règles sont définies pour affecter un poids $W(i,j)$ entre deux sommets (i,j), sur les arêtes du graphe :

- $W(i,j) = 0$ si i et j sont deux clusters ou deux instances
- $W(i,j) = 0$ si l'instance i n'appartient pas au cluster j
- $W(i,j) = 1$ si l'instance i appartient au cluster j

Le graphe bipartite est ensuite partitionné pour former le consensus. Cette classification est

	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(3)}$
x_1	1	3	1
x_2	1	3	1
x_3	2	1	1
x_4	2	1	1
x_5	3	2	2
x_6	3	2	2

Résultats de clustering

	$H^{(1)}$			$H^{(2)}$			$H^{(3)}$	
	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
x_1	1	0	0	0	0	1	1	0
x_2	1	0	0	0	0	1	1	0
x_3	0	1	0	1	0	0	1	0
x_4	0	1	0	1	0	0	1	0
x_5	0	0	1	0	1	0	0	1
x_6	0	0	1	0	1	0	0	1

Matrice de transition du graphe.

Tab. 3.2: *Exemple de représentation par hypergraphe de trois résultats de clustering.*

effectuée en utilisant l'algorithme HMETIS ou une méthode de clustering spectral [Dhillon, 2001].

3.2.1.5 Méthodes de combinaison basées sur la matrice de co-association

Une autre approche dans le clustering par ensemble est présentée par Fred et Jain [2005] qui introduisent le concept d'accumulation de preuves (*evidence accumulation*). L'idée principale de cette approche est d'utiliser la *matrice de co-association* calculée à partir des résultats de l'ensemble. Cette matrice, de taille $n \times n$ (n étant le nombre d'objets), donne l'information du nombre de fois où deux objets ont été classés ensemble dans le même cluster dans les différents résultats de l'ensemble. Un exemple de calcul de cette matrice est donné dans le tableau 3.3. La matrice de co-association contient à l'indice (i, j) le nombre de fois où les objets (x_i, x_j) ont été classés ensemble dans les différents résultats :

$$\texttt{co-assoc}(i,j) = \frac{1}{K}\sum_{k=0}^{K} V(\mathcal{C}^{(k)}, x_i, x_j) \tag{3.4}$$

où $V(.)$ renvoie 1 si les instances (x_i, x_j) ont été classées ensemble dans le résultat $\mathcal{C}^{(k)}$ et 0 sinon. Une méthode de clustering hiérarchique est ensuite utilisée en prenant la matrice de co-association comme une matrice de distance avec un seuil t qui détermine le similarité minimale à partir duquel deux clusters seront fusionnés. Ce type d'approche permet notamment de trouver des clusters ayant des formes non conventionnelles comme des spirales (voir la figure 3.1, page 40). Il est cependant nécessaire que les clusters soient clairement séparables dans l'espace des données sous peine de ne former qu'un cluster unique lors du clustering final. De plus, la construction et l'utilisation de la matrice de co-association implique une complexité quadratique en temps et en mémoire en fonction du nombre d'observations $O(n^2)$, ce qui rend l'utilisation de ces méthodes peu attractive pour traiter de grands volumes de données.

Une méthode pour introduire une pondération dans la matrice de co-association a également été proposée par Duarte et al. [2005]. Elle permet de pondérer la décision de chaque résultat de l'ensemble en fonction d'un critère de qualité interne affecté à chaque résultat (voir section 2.3). Cet indice est ensuite utilisé pour mettre à jour la matrice de co-association pondérée :

$$\texttt{w-co-assoc}(i,j) = \frac{1}{K}\sum_{k=0}^{K} V(\mathcal{C}^{(k)}, x_i, x_j) \times Q(\mathcal{C}^{(k)}) \tag{3.5}$$

où $V(.)$ renvoie 1 si les instances (x_i, x_j) ont été classées ensemble dans le résultat $\mathcal{C}^{(k)}$ et 0 sinon, et $Q(.)$ renvoie l'indice de qualité du résultat $\mathcal{C}^{(k)}$.

	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(3)}$
x_1	1	3	1
x_2	1	3	1
x_3	2	1	1
x_4	2	1	1
x_5	3	2	2
x_6	3	2	2

Résultats de clustering.

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	-	3	1	1	0	0
x_2	3	-	1	1	0	0
x_3	1	1	-	3	0	0
x_4	1	1	3	-	0	0
x_5	0	0	0	0	-	3
x_6	0	0	0	0	3	-

Matrice de co-association.

Tab. 3.3: *Exemple de calcul de la matrice de co-association.*

3.2.1.6 Méthodes de combinaison basées sur la définition d'un nouvel espace de données

Une autre proposition [Topchy et al., 2003] consiste à construire un nouvel espace de données à partir de l'ensemble des clusterings. Le problème du clustering par ensemble est modélisé sous la forme d'un problème de clustering à attributs catégoriels. Chaque résultat de l'ensemble donne lieu a un attribut catégoriel représentant l'objet. Ce nouvel espace de données est ensuite utilisé pour effectuer le clustering final. Une fonction d'utilité est introduite qui permet d'évaluer la qualité du consensus en fonction des différents attributs. He et al. [2005] ont défini formellement le lien entre le clustering par ensemble et le clustering de données catégorielles. Dans des travaux similaires, Hu et al. [2006] utilisent un champ de Markov aléatoire ainsi qu'une estimation du maximum de vraisemblance pour définir une métrique entre les résultats de clustering. Ils présentent deux méthodes basées sur cette nouvelle similarité permettant de calculer un consensus.

3.2.1.7 Méthodes basées sur le ré-étiquetage et le vote

Une autre approche dans le clustering par ensemble consiste à effectuer un vote au sein des différents résultats pour produire le consensus. Ainsi, Ayad et Kamel [2008] présentent un algorithme de vote cumulatif qui permet de trouver un consensus parmi plusieurs partitions à nombre différent de clusters. Ils décrivent plusieurs approches de vote pondéré qui permettent de calculer une densité de probabilité résumant les clustering initiaux. Une étude empirique montre que leur approche semble donner de meilleurs résultats que les approches par graphe. D'autres approches par vote sont présentées par Nguyen et Caruana [2007] : IVC (*Iterative Voting Consensus*), IPVC (*Iterative Probabilistic Voting Consensus*) et IPC (*Iterative Pairwise Consensus*). Ces algorithmes utilisent une carte de caractéristiques calculée à partir de l'ensemble de résultats de clustering et utilisent une approche de type *Expectation-Maximisation* pour calculer le consensus. Un problème important dans ces approches par vote est la nécessité de mettre en correspondance les différents clusters des clusterings initiaux. La plupart des méthodes [Ayad et Kamel, 2008; Zhou et Tang, 2006] sélectionnent un membre de l'ensemble servant de *base* pour le ré-étiquetage des autres résultats. Une fois cette base choisie, les clusters des autres résultats sont mis en correspondance en observant le recouvrement des clusters. Cependant, le choix de l'heuristique de sélection de la base a d'importantes conséquences sur les résultats obtenus et détermine notamment le choix du nombre de clusters. Pour résoudre ce problème, Long et al. [2005] proposent une méthode basée sur un mécanisme de correspondance floue entre les différents clusters des résultats. L'algorithme proposé produit un consensus mais également une matrice de correspondance qui fournit les liens entre les clusters des différents résultats.

Une autre méthode proposée par Zhou et Tang [2006] ne s'intéressent qu'aux partitions ayant le même nombre de clusters (voir l'exemple tableau 3.4). Une fois les clusters ré-étiquetés, un vote à la majorité est effectué pour construire le consensus. Des pondérations sont appliquées à chacune des décisions en fonction de son taux d'accord avec les autres méthodes. Dimitriadou et al. [2002] présentent un algorithme basé sur la minimisation de l'erreur au carré entre les résultats de l'ensemble. L'algorithme de vote trouve une solution approchée, la recherche de la meilleure solution du problème de minimisation étant impossible, car il est nécessaire d'énumérer toutes les permutations possibles. L'algorithme effectue une recherche séquentielle, où pour chaque partition de l'ensemble,

	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(3)}$
x_1	1	3	1
x_2	1	3	1
x_3	2	1	1
x_4	2	1	3
x_5	3	2	3
x_6	3	2	2

Résultats initiaux.

	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(3)}$
x_1	1	1	1
x_2	1	1	1
x_3	2	2	1
x_4	2	2	2
x_5	3	3	2
x_6	3	3	3

Résultats ré-étiquetés.

Objet	x_1	x_2	x_3	x_4	x_5	x_6
Vote	(**3**,0,0)	(**3**,0,0)	(1,**2**,0)	(0,**3**,0)	(0,1,**2**)	(0,0,**3**)
Résultat	1	1	2	2	3	3

Résultat du vote.

Tab. 3.4: *Exemple de ré-étiquetage suivi d'un vote.*

la meilleure permutation d'étiquette est effectuée. Une méthode multivue est proposée par Wemmert et Gancarski [2002a] pour construire un consensus parmi les résultats de l'ensemble. Dans cette méthode, chaque objet va voter pour le cluster auquel il appartient et pour un cluster dans chacun des autres résultats. Les clusters ayant obtenu le maximum de votes seront retenus pour construire le consensus. Cette approche est présentée en détails dans la section 3.3.2.3.

Une autre approche proposée par Oliveira et Pedrycz [2007] consiste à utiliser une correspondance douce entre les clusters pour refléter le fait qu'un cluster d'un résultat peut être lié avec plusieurs clusters d'un autre résultat. Une nouvelle fonction de consensus appelée ITK (*Information Theoretic* KMEANS) est proposée. Celle-ci prend en entrée le degré d'appartenance de chaque objet aux clusters des résultats et non pas une appartenance dure comme dans les autres travaux sur l'ensemble clustering. Le fait de posséder cette information d'appartenance permet de définir une fonction de consensus plus précise qui prend en compte l'information floue proposée par les algorithmes de clustering. Il est cependant nécessaire d'utiliser des algorithmes de clustering proposant ce type de résultat en sortie. L'algorithme EM est par exemple bien adapté car il est possible d'obtenir un degré d'appartenance à chaque loi composant le résultat.

3.2.1.8 Bilan

Les méthodes par ensemble consistent à créer un résultat de clustering appelé consensus à partir d'un ensemble de résultats de clustering. Ces méthodes s'intéressent principalement à deux aspects. Le premier est la création des résultats de l'ensemble (différents algorithmes, différentes initialisations, etc.). La seconde est la définition d'une fonction permettant de trouver la partition consensuelle finale. Ces méthodes nécessitent généralement beaucoup de résultats en entrée pour être efficaces et il est souvent difficile de faire un choix sur la fonction de consensus à utiliser.

3.2.2 Approches multiobjectives

Le clustering multiobjectif a pour but d'optimiser simultanément plusieurs critères de clustering. L'idée est de mieux saisir la notion de cluster en définissant explicitement différentes fonctions objectives. Les algorithmes sont ainsi capables de produire un ensemble de solutions qui sont des compromis des différents objectifs utilisés.

Ainsi, la méthode MOCK (*Multi-Objective Clustering with automatic K-determination*) [Handl et Knowles, 2007] utilise deux objectifs : le premier est de maximiser la compacité des clusters, et le second leur connectivité. Un algorithme évolutionnaire multiobjectif est utilisé pour optimiser ces deux critères simultanément. La méthode utilise un front de Pareto [Konak et al., 2006] qui consiste à sélectionner les solutions non dominées, c'est-à-dire celles respectant le mieux les deux

objectifs de manière simultanée. À la fin de l'évolution, les solutions présentes sur le front de Pareto forment l'ensemble des solutions fournies par l'algorithme. Une heuristique est ensuite utilisée pour sélectionner la meilleure solution potentielle en utilisant le nombre de clusters des solutions présentes sur le front. Dans [Handl et Knowles, 2006], les auteurs présentent un moyen d'intégrer des connaissances du domaine (voir chapitre 4) à travers un troisième objectif basé sur un ensemble d'objets étiquetés. Cette version semi-supervisée donne de meilleurs résultats que la version sans connaissance.

Faceli et al. [2006] ont proposé une autre méthode appelée MOCLE (*Multi-Objective Clustering Ensemble*) qui intègre les deux mêmes fonctions objectives (maximisation de la compacité et la connectivité des clusters) que MOCK. Cependant, un opérateur de croisement spécial est ajouté qui utilise des techniques de clustering par ensemble. Le but de la méthode MOCLE est de produire un ensemble de solutions représentant des compromis entre les deux objectifs alors que MOCK cherchait à trouver une unique solution. Une extension semi-supervisée de MOCLE a également été proposée par Faceli et al. [2007]. La connaissance à propos de la structure des données est également intégrée à l'aide d'un objectif additionnel. Enfin, une autre approche proposée par Law et al. [2004] utilise également plusieurs objectifs. Cependant, la solution finale est produite en sélectionnant les meilleurs clusters dans les résultats produits. Une méthode d'échantillonnage est utilisée pour estimer la qualité des clusters en se basant sur la stabilité de leur présence sur plusieurs exécutions.

3.2.3 Approches par combinaison de méthodes floues

Une architecture de clustering flou a été introduite par Pedrycz [2002] dans laquelle plusieurs sous-ensembles d'objets d'un jeu de données initial sont traités dans le but de trouver une structure commune. Les différents sous-ensembles sont tout d'abord traités indépendamment, puis, chaque partition est modifiée en accord avec les autres partitions : chaque résultat produit à partir d'un sous-ensemble est modifié par rapport aux informations obtenues sur les autres sous-ensembles. Des expériences ainsi que plus de détails sur la méthode sont fournis dans [Pedrycz et Rai, 2008]. Une application de cette approche pour l'analyse de contenu web est proposée par Loia et al. [2007]. Les auteurs présentent une méthode collaborative basée sur la proximité des objets et montrent comment cette information peut être utilisée pour découvrir la structure d'informations sur le web dans des espaces sémantiques différents.

Une autre plateforme de clustering collaboratif flou est proposée par Mitra et al. [2006] où les ensembles bruts (*rough sets*) sont utilisés pour créer un paradigme collaboratif. Un algorithme de clustering est développé en intégrant les avantages des ensembles flous et des ensembles bruts. Une analyse quantitative et des résultats expérimentaux sont aussi présentés sur des données artificielles et réelles. Ces approches collaboratives floues apportent des fondements théoriques intéressants mais sont limitées sur de nombreux aspects : chaque partition doit avoir le même nombre de clusters ; la question difficile de la correspondance entre les clusters est supposée résolue ; la distance entre chaque point et le centre des clusters dans chaque solution doit être connue. Malgré ces contraintes, il a été montré, au moins sur des exemples simples à deux ou trois clusters que la collaboration a un effet positif sur la qualité des clusters.

Des travaux récents [Grozavu et Bennani, 2010] utilisent l'algorithme SOM pour effectuer cette tâche de clustering distribué de plusieurs sous-ensembles d'objets. La méthode est décomposée en deux phases. La première consiste à appliquer l'algorithme SOM sur les différentes données indépendamment. La seconde phase consiste à faire collaborer ces différentes cartes pour les enrichir. Des travaux similaires sont proposés par [Cleuziou G., 2009] avec une approche permettant de traiter des données multi-représentées, c'est-à-dire un même ensemble d'individus décrits par plusieurs représentations.

3.2.4 Bilan

Dans cette section nous avons présenté les principales approches de la littérature concernant l'utilisation de plusieurs algorithmes ou résultats de clustering. Les méthodes de clustering par

ensemble (section 3.2.1) ne s'intéressent généralement pas à la génération des résultats initiaux et ne s'intéressent qu'à la création du résultat consensuel final. Ainsi les algorithmes utilisés pour créer les résultats initiaux ne sont pas ou peu étudiés. Ces méthodes introduisent un nouveau biais, relatif à la fonction objective choisie pour fusionner les différents résultats. De plus, il est souvent nécessaire de générer de nombreuses partitions (de quelques dizaines à plusieurs centaines) pour obtenir des résultats pertinents. Ces problèmes sont partagés avec les méthodes multiobjectives (section 3.2.2) qui utilisent des objectifs qui ne sont pas forcément ceux utilisés pour la création des solutions initiales. De plus, les méthodes multiobjectives génèrent de très nombreuses solutions pour s'assurer de trouver des résultats pertinents. Enfin, les méthodes multiobjectives proposent en général en sortie un ensemble de résultats hétérogènes et le choix de la solution finale est laissé à l'expert. Les méthodes par combinaison de méthodes floues (section 3.2.3) proposent des fondements théoriques solides sur la collaboration entre méthodes, mais ces méthodes ne proposent pas de solution à de nombreux problèmes (nombre de clusters différents, correspondance entre les clusters, passage à l'échelle, etc.). Dans la section suivante nous allons aborder une autre approche du clustering collaboratif, plus générique, et indépendante de la méthode de clustering utilisée.

3.3 Le clustering collaboratif

3.3.1 Problématique

Dans le cadre du travail collectif, trois approches existent et sont utilisées notamment en management et en science de l'éducation. La première, la *coordination* est une activité collective qui consiste à construire un tout à partir de parties plus petites fournies par différents individus. La seconde, la *coopération* est une action conjointe où les acteurs se mettent d'accord sur un ensemble défini de règles à suivre qui permettent d'agréger le travail de chaque individu de manière efficace. Et enfin, la *collaboration*, qui est un processus de création collective et de partage de connaissances entre plusieurs individus. L'utilisation du terme *collaboratif* pour qualifier les méthodes présentées dans cette section, se justifie par la volonté de créer un système dans lequel les individus partagent activement leurs informations locales dans le but d'une recherche de solution globale. L'objectif est de trouver une solution que les méthodes n'aurait pas pu, ou auraient éprouvé des difficultés à trouver seules.

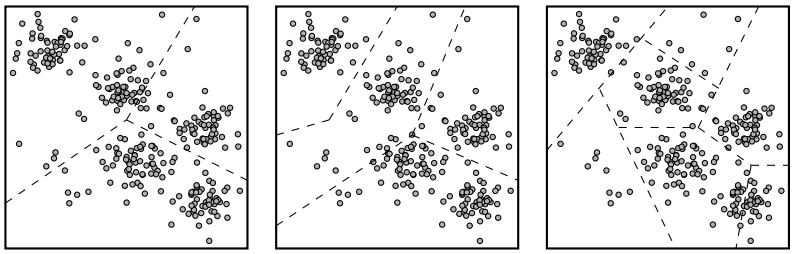
Dans cette section nous allons étudier la méthode de clustering collaboratif proposée par Wemmert [2000] appelée SAMARAH. Cette méthode consiste en la collaboration de différentes méthodes de clustering pour tenter de trouver un consensus sur le clustering d'un jeu de données. L'objectif de cette collaboration est de réduire l'impact du choix d'une méthode et de ses paramètres sur le résultat. Étant donné un ensemble de N résultats de clustering, l'idée est de modifier de façon itérative et collaborative les résultats initiaux afin d'en améliorer la similarité et la qualité, autorisant ainsi un consensus final plus pertinent et de meilleure qualité.

3.3.2 Présentation de l'approche samarah

La méthode SAMARAH est basée sur le principe d'un raffinement mutuel et itératif de plusieurs résultats de clustering. Le système peut être décomposé en trois grandes étapes :

1. La génération des résultats initiaux ;
2. Le raffinement des différents résultats ;
3. La combinaison des résultats raffinés.

La première étape de la méthode consiste à générer les résultats initiaux qui seront ensuite utilisés par celle-ci. Dans cette étape, des algorithmes de clustering sont appliqués sur les données. Différents algorithmes ou paramétrages du même algorithme peuvent être utilisés. La figure 3.2 donne un exemple de trois clusterings différents d'un même jeu de données.



(a) 3 clusters (b) 4 clusters (c) 7 clusters

Fig. 3.2: *Exemple de trois clusterings différents du même jeu de données.*

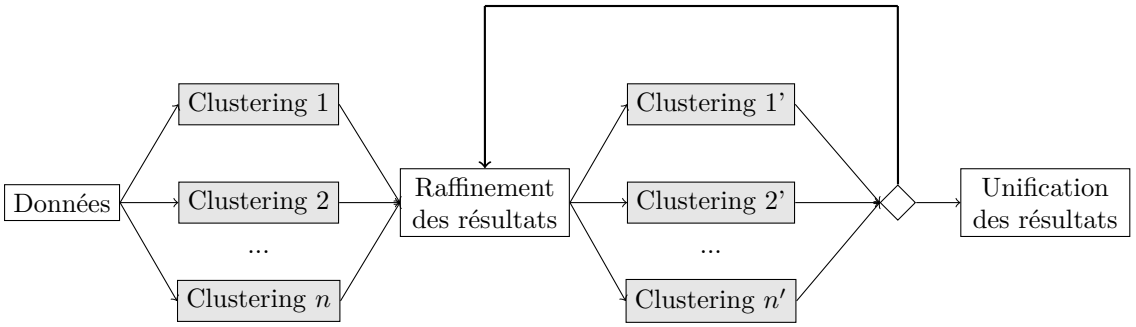


Fig. 3.3: *Schéma illustrant les différentes étapes du clustering collaboratif.*

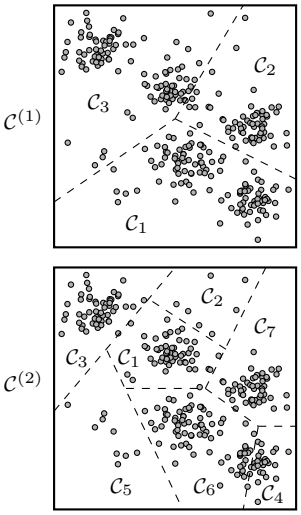
Lors de la deuxième étape, chacun des résultats va être comparé avec l'ensemble des résultats proposés par les autres méthodes. Le but est d'évaluer la similarité entre les différents résultats pour observer les différences dans les regroupements des objets. Une fois ces différences (appelées par la suite *conflits*) identifiées, l'objectif est de modifier les résultats pour tenter de réduire ces différences, c'est-à-dire résoudre les conflits. En fonction des informations obtenues sur les différences entre les résultats, des modifications (fusion de clusters, scission de clusters, reclustering de clusters) sont appliquées de manière itérative aux résultats. Cette étape peut être vue comme une remise en cause en fonction des informations fournies par les autres acteurs de la collaboration. Après plusieurs itérations de raffinement, il est attendu des résultats qu'ils soient plus similaires qu'avant cette étape collaborative.

Lors de la troisième étape, les différents résultats raffinés sont combinés si nécessaire pour proposer un résultat unique. Ce calcul d'un résultat consensuel est simplifié par la forte similarité des résultats.

La figure 3.3 présente un schéma des différentes étapes du clustering collaboratif. Dans les sections suivantes nous allons étudier en détail les différentes étapes de la méthode.

3.3.2.1 Génération des résultats initiaux

La génération des résultats initiaux consiste à appliquer plusieurs méthodes de clustering aux données. Il est nécessaire lors de cette étape de faire un choix sur le nombre de méthodes impliquées dans la collaboration ainsi que sur leur type. Nous avons vu dans le chapitre 2 qu'il existe un grand nombre de méthodes de clustering. La plupart des méthodes de clustering les plus courantes peuvent être utilisées dans la méthode SAMARAH. Nous verrons toutefois qu'il est nécessaire de modifier légèrement les méthodes pour pouvoir les utiliser de manière efficace dans le système. On obtient donc à l'issue de cette étape, un ensemble de clusterings différents $\mathbb{C} = \{\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(N)}\}$, chaque clustering $\mathcal{C}^{(i)}$ ayant été généré à partir d'une méthode et/ou d'un paramétrage spécifique.



	$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$
$C_1^{(2)}$		0.11	0.42
$C_2^{(2)}$		0.05	0.03
$C_3^{(2)}$			0.50
$C_4^{(2)}$	0,45		
$C_5^{(2)}$	0,05		0,05
$C_6^{(2)}$	0,50	0,02	
$C_7^{(2)}$		0,82	

$\Omega^{(1,2)}$

	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	$C_4^{(2)}$	$C_5^{(2)}$	$C_6^{(2)}$	$C_7^{(2)}$
$C_1^{(1)}$				1	0,55	0,98	0,02
$C_2^{(1)}$	0,13	0,43				0,02	0,98
$C_3^{(1)}$	0,87	0,57	1		0,45		

$\Omega^{(2,1)}$

Fig. 3.4: *Exemple de calcul de matrice de confusion à partir de deux résultats de clustering.*

3.3.2.2 Raffinement des résultats

Le raffinement des résultats consiste à comparer les différents résultats et à observer la répartition des objets dans les différents clusters des différents résultats. L'objectif est d'identifier sur quelle partie du regroupement des données les méthodes sont en désaccord (et respectivement en accord). Pour se faire il est nécessaire de comparer les résultats entre eux. Pour pouvoir observer les similitudes et les différences de chaque résultat par rapport à tous les autres résultats, ceux-ci sont comparés deux à deux. Pour effectuer cette comparaison, la matrice de confusion (Ω) est calculée pour chaque couple de clusterings $(\mathcal{C}^{(i)}, \mathcal{C}^{(j)})$:

$$\Omega^{(i,j)} = \begin{pmatrix} \alpha_{1,1}^{(i,j)} & \cdots & \alpha_{1,K^{(j)}}^{(i,j)} \\ \vdots & \ddots & \vdots \\ \alpha_{K^{(i)},1}^{(i,j)} & \cdots & \alpha_{K^{(i)},K^{(j)}}^{(i,j)} \end{pmatrix} \text{ où } \alpha_{k,l}^{(i,j)} = \frac{\left|C_k^{(i)} \bigcap C_l^{(j)}\right|}{\left|C_k^{(i)}\right|} \tag{3.6}$$

La matrice de confusion $\Omega^{(i,j)}$ contient la répartition des objets dans les clusters de deux résultats de clustering. Cette matrice permet d'observer si les objets d'un cluster d'un résultat ont été regroupés de manière similaire dans l'autre résultat ou au contraire ont été répartis dans plusieurs clusters et dans quelles proportions.

Cette matrice de confusion est le cœur de la comparaison des résultats en clustering collaboratif. À partir de cette information, il est possible d'évaluer la similarité des résultats, et plus encore, d'identifier avec précision leurs désaccords. Un autre intérêt de cette matrice est qu'elle est totalement indépendante de la méthode utilisée pour créer les clusters, car celle-ci ne nécessite que l'information de l'affectation des objets aux clusters pour être calculée. La figure 3.4 présente un exemple de calcul de matrice de confusion pour deux résultats.

Nous allons voir dans la suite de cette section un ensemble de critères statistiques utilisant cette matrice de confusion et qui permettent d'obtenir des informations sur la similarité de deux résultats de clustering. Le premier critère est un critère de similarité inter-cluster qui permet la comparaison de deux clusters de deux résultats différents. Pour se faire deux matrices de confusion sont calculées. En effet, comme les résultats n'ont, d'une part, pas forcément le même nombre de

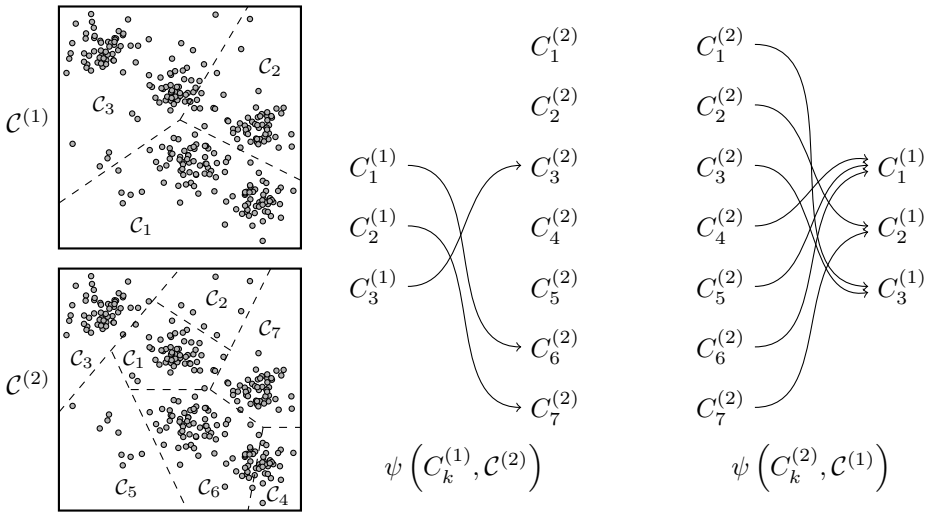


Fig. 3.5: *Exemple de fonction de correspondance entre les clusters de deux résultats.*

clusters, et que d'autre part, les coefficients de la matrice sont normalisés par la taille des clusters d'un des deux résultats, on obtient $\Omega^{i,j} \neq \Omega^{j,i}$. Il est donc nécessaire de calculer cette matrice dans les deux *sens* ($\Omega^{i,j}$ et $\Omega^{j,i}$) pour pouvoir comparer les deux résultats. Pour calculer la similarité S entre les deux clusters, on peut ensuite utiliser ces deux matrices, en observant les deux coefficients correspondants aux deux clusters à comparer :

$$S\left(C_k^{(i)}, C_l^{(j)}\right) = \alpha_{k,l}^{(i,j)} \alpha_{l,k}^{(j,i)} \tag{3.7}$$

Cependant, ce critère ne prend en compte que l'intersection entre les deux clusters considérés sans prendre en compte la répartition des objets dans les autres clusters. Si par exemple les objets d'un cluster d'un résultat se retrouve à 50% dans le cluster d'un autre résultat, la répartition des 50% d'objets restants n'est pas prise en compte. Que ces objets soient répartis dans un autre cluster ou dans plusieurs autres clusters, la valeur du coefficient est identique. Pour prendre en compte cette distribution, la similarité entre deux clusters est définie telle que :

$$S\left(C_k^{(i)}, C_l^{(j)}\right) = \rho_k^{(i,j)} \alpha_{l,k}^{(j,i)} \tag{3.8}$$

où

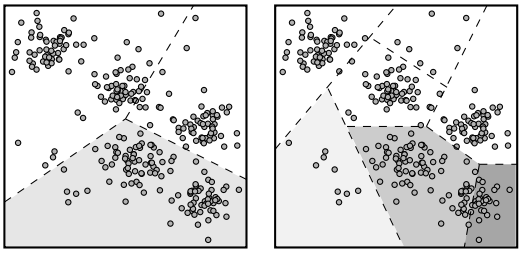
$$\rho_k^{(i,j)} = \sum_{r=1}^{K^{(j)}} (\alpha_{k,r}^{(i,j)})^2 \tag{3.9}$$

Une fois cette similarité définie entre les clusters des différents résultats, il est possible de définir une fonction qui va mettre en correspondance un cluster $C_k^{(i)}$ d'un résultat $\mathcal{C}^{(i)}$ et son cluster le plus similaire dans un autre résultat $\mathcal{C}^{(j)}$:

$$\psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right) = \arg \max_{C_l^{(j)} \in \mathcal{C}^{(j)}} S\left(C_k^{(i)}, C_l^{(j)}\right) \tag{3.10}$$

La figure 3.5 montre un exemple du calcul de cette fonction de correspondance entre deux résultats. Grâce à cette fonction de correspondance, il est possible d'identifier si les différents résultats sont en accord sur le regroupement des données. Nous définissons le concept de *conflit* entre un cluster d'un résultat et un autre résultat, si ce cluster ne se retrouve pas de manière parfaite dans l'autre résultat.

L'étape de détection des conflits consiste à chercher dans $\mathbb{C}$ tous les couples $\left(C_k^{(i)}, \mathcal{C}^{(j)}\right)$, $i \neq j$,



(a) $\mathcal{C}^{(1)}$, 3 clusters (b) $\mathcal{C}^{(2)}$, 7 clusters

Importance du conflit : $CI\left(\mathcal{K}_2^{(1,2)}\right) = 0.573$

Fig. 3.6: *Exemple de conflit entre deux résultats de clustering.*

tel que $S\left(\mathcal{C}_k^{(i)}, \psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right)\right) < 1$, ce qui signifie que le cluster $C_k^{(i)}$ ne peut pas être trouvé exactement dans le résultat $\mathcal{C}^{(j)}$. La liste des conflits de $\mathbb{C}$ peut se définir comme :

$$\texttt{conflits}(\mathbb{C}) = \left\{(C_k^{(i)}, \mathcal{C}^{(j)}) : i \neq j, S\left(C_k^{(i)}, \psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right)\right) < 1\right\} \tag{3.11}$$

Chaque conflit $\mathcal{K}_k^{(i,j)}$ est lié a un cluster $C_k^{(i)}$ et un résultat $\mathcal{C}^{(j)}$. Son importance CI, est évaluée grâce à la similarité entre le cluster du premier résultat et les clusters du deuxième résultat (équation (3.8)).

$$CI\left(\mathcal{K}_k^{(i,j)}\right) = 1 - S\left(C_k^{(i)}, \psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right)\right) \tag{3.12}$$

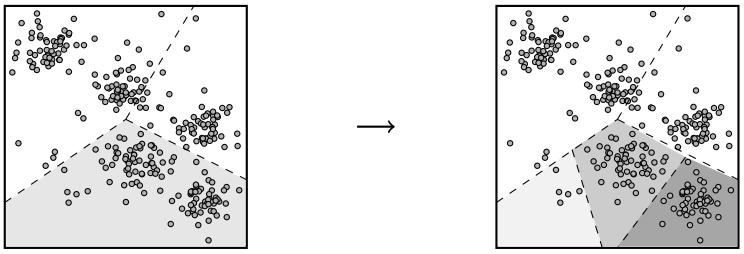
La figure 3.6 donne un exemple de conflit entre deux résultats. Plus un cluster est dispersé dans plusieurs clusters d'un autre résultat, plus il est en conflit avec ce résultat et plus l'importance du conflit est élevée. La suite du processus collaboratif consiste à essayer de résoudre ces conflits. Dans sa version initiale, la méthode SAMARAH cherche toujours à résoudre le conflit le plus important. C'est cette approche que nous allons étudier dans un premier temps. Plus loin dans ce chapitre (section 3.4) nous présentons différentes stratégies pour résoudre ces conflits, en étudiant notamment leur ordre de résolution.

Le conflit le plus important est sélectionné parmi tous les conflits détectés entre tous les couples de résultats. La résolution d'un conflit $\mathcal{K}_k^{(i,j)}$ consiste à appliquer un opérateur sur chacun des résultats impliqués dans le conflit : $\mathcal{C}^{(i)}$ et $\mathcal{C}^{(j)}$. Les opérateurs qui peuvent être appliqués aux résultats sont :

- la **fusion** de clusters : plusieurs clusters sont fusionnés ensemble. La figure 3.7 donne un exemple de cet opérateur ;
- la **scission** de cluster en sous-clusters : un clustering est appliqué aux objets d'un cluster pour créer des sous-clusters. La figure 3.8 donne un exemple de cet opérateur ;
- la **reclustering** d'un cluster : un cluster est retiré et ses objets sont distribués dans les autres clusters restants. La figure 3.9 donne un exemple de cet opérateur.

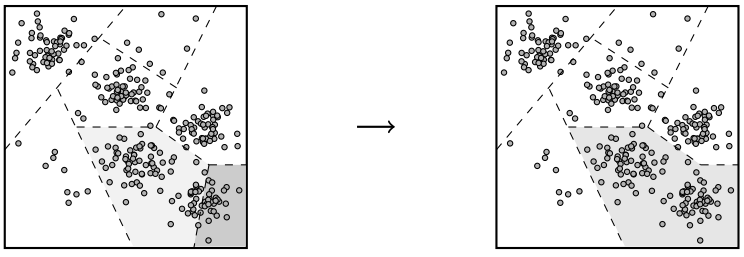
L'opérateur à appliquer est choisi grâce au nombre de clusters impliqués dans le conflit, c'est-à-dire tel que $S(C_k^{(i)}, C_l^{(j)}) > p_{cr}$, où $0 \leq p_{cr} \leq 1$ est un paramètre choisi. Par exemple si $p_{cr} = 0.2$ cela signifie que si $C_k^{(i)} \cap C_l^{(j)}$ représente moins de 20% des objets de $C_k^{(i)}$, $C_l^{(j)}$ n'est pas considéré comme un représentant de $C_k^{(i)}$.

Les détails du processus de sélection de l'opérateur à appliquer sont présentés dans l'algorithme 1. Un opérateur est appliqué à chacun des deux résultats impliqués dans le conflit. Cependant, l'application de ces deux opérateurs (un sur chaque résultat) n'est pas toujours pertinente. En effet, cela n'entraine pas toujours une augmentation de la similarité des résultats impliqués dans le conflit. De plus, une itération de cette résolution de conflit peut mener à des solutions triviales



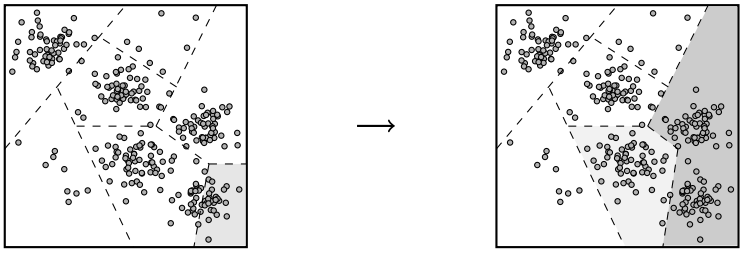
$\mathcal{C}^{(1)}$ avant l'application de l'opérateur. $\mathcal{C}^{(1)}$ après l'application de l'opérateur.

Fig. 3.7: *Exemple d'application de l'opérateur de scission.*



$\mathcal{C}^{(1)}$ avant l'application de l'opérateur. $\mathcal{C}^{(1)}$ après l'application de l'opérateur.

Fig. 3.8: *Exemple d'application de l'opérateur de fusion.*



$\mathcal{C}^{(1)}$ avant l'application de l'opérateur. $\mathcal{C}^{(1)}$ après l'application de l'opérateur.

Fig. 3.9: *Exemple d'application de l'opérateur de reclustering.*

et non voulues. Par exemple, tous les résultats ayant un unique cluster, ou un cluster pour chaque objet. Ces solutions non pertinentes doivent être évitées.

Algorithme 1: Résolution d'un conflit

Entrées : $\mathbb{C}$ les différents résultats de clustering, $\mathcal{K}_k^{i,j}$ le conflit à résoudre
Sorties : $\mathbb{C}^\star = \texttt{conflictResolution}\left(\mathbb{C}, \mathcal{K}_k^{i,j}\right)$ le nouvel ensemble après la résolution
soit $\kappa = \{C_l^{(j)}, \forall 1 \leq l \leq K^{(j)} : S(C_k^{(i)}, C_l^{(j)}) > p_{cr}\}$
si $|\kappa| > 1$ **alors**
  $\mathcal{C}^{(i')} = \mathcal{C}^{(i)} \setminus \{C_k^{(i)}\} \cup \texttt{scission}(C_k^{(i)}, |\kappa|)$
  $\mathcal{C}^{(j')} = \mathcal{C}^{(j)} \setminus \kappa \cup \texttt{fusion}(\kappa, \mathcal{C}^{(j)})$
sinon
  $\mathcal{C}^{(i')} = \texttt{reclustering}(\mathcal{C}^{(i)} \setminus \{C_k^{(i)}\})$
$\{\mathcal{C}^{(i\star)}, \mathcal{C}^{(j\star)}\} = \arg\max \gamma^{(I,J)}$ avec $I \in \{i, i'\}, J \in \{j, j'\}$
$\mathbb{C}^\star = \mathbb{C} \setminus \{\mathcal{C}^{(i)}, \mathcal{C}^{(j)}\} \cup \{\mathcal{C}^{(i\star)}, \mathcal{C}^{(j\star)}\}$

Pour cela, la méthode SAMARAH utilise un *coefficient local d'évaluation*, qui permet de contrôler la convergence de la collaboration entre deux clusterings. Ce coefficient est basé sur la similarité entre tous les clusters de deux résultats, et prend en compte un coefficient de qualité des clusters eux-mêmes (δ), sélectionné par l'expert. Ce coefficient de qualité évalue la qualité du clustering pour éviter d'obtenir une solution triviale :

$$\gamma(\mathcal{C}^{(i)}, \mathcal{C}^{(j)}) = \frac{1}{2}\left(p_s.\left(\frac{1}{K^{(i)}}\sum_{k=1}^{K^{(i)}}\omega_k^{(i,j)} + \frac{1}{K^{(j)}}\sum_{k=1}^{K^{(j)}}\omega_k^{(j,i)}\right) + p_q.\left(\delta^{(i)} + \delta^{(j)}\right)\right) \tag{3.13}$$

où

$$\omega_k^{(i,j)} = S\left(C_k^{(i)}, \psi\left(C_k^{(i)}, \mathcal{C}^{(j)}\right)\right) \tag{3.14}$$

et, p_q et p_s sont des paramètres du système ($p_q + p_s = 1$). Nous verrons en détail la définition de l'indice de qualité δ qui nous permettra notamment d'intégrer des connaissances dans la méthode dans le chapitre suivant (Chapitre 4).

Soit $\mathcal{C}^{(i')}$ (respectivement $\mathcal{C}^{(j')}$) le résultat de l'application d'un opérateur sur $\mathcal{C}^{(i)}$ (respectivement $\mathcal{C}^{(j)}$). Le coefficient local d'évaluation (équation (3.13)) est calculé pour chacun des couples : $(\mathcal{C}^{(i)}, \mathcal{C}^{(j)})$, $(\mathcal{C}^{(i')}, \mathcal{C}^{(j')})$, $(\mathcal{C}^{(i')}, \mathcal{C}^{(j)})$, $(\mathcal{C}^{(i)}, \mathcal{C}^{(j')})$. Le couple ayant la meilleure évaluation est conservée comme solution à la résolution du conflit.

Après cette étape de résolution locale d'un conflit (locale car effectuée pour un couple de résultats alors que plus de deux clusterings peuvent être impliqués dans la collaboration), une étape d'évaluation globale est engagée. Cette étape globale d'évaluation a pour but de valider ou non les décisions prises au niveau local. Le *coefficient global d'évaluation* est calculé en fonction de tous les coefficients locaux d'évaluation pour chaque couple de résultats :

$$\mathbf{\Gamma}(\mathbb{C}) = \frac{1}{N}\sum_{i=1}^{N}\Gamma(\mathcal{C}^{(i)}) \tag{3.15}$$

où

$$\Gamma(\mathcal{C}^{(i)}) = \frac{1}{N-1}\sum_{\substack{j=1 \\ j\neq i}}^{N}\gamma(\mathcal{C}^{(i)}, \mathcal{C}^{(j)}) \tag{3.16}$$

Nous montrons section 3.4.4 sur des exemples que cet indice est un bon indicateur de la qualité du résultat global. Trois cas peuvent survenir concernant l'évolution de ce critère pendant la phase de collaboration :

1. Cette étape de résolution de conflit permet d'obtenir une meilleure solution au niveau global. Dans ce cas, la solution courante est remplacée par cette nouvelle solution. La liste des conflits est recalculée, et une nouvelle itération est engagée ;
2. L'étape de résolution propose la même solution qu'avant l'application des opérateurs, ce qui signifie que la résolution de ce conflit n'est pas pertinente. Ce conflit est retiré de la liste, et une nouvelle itération est engagée ;
3. Si la solution proposée par la résolution du conflit donne un résultat globalement moins pertinent, ce résultat est tout de même considéré pour éviter de tomber dans un extremum local. Cependant, si dans la suite des itérations, aucune résolution de conflit ne permet d'obtenir une meilleure solution (après avoir épuisé la moitié des conflits à résoudre), tous les résultats sont réinitialisés à la meilleure solution courante et le conflit est retiré.

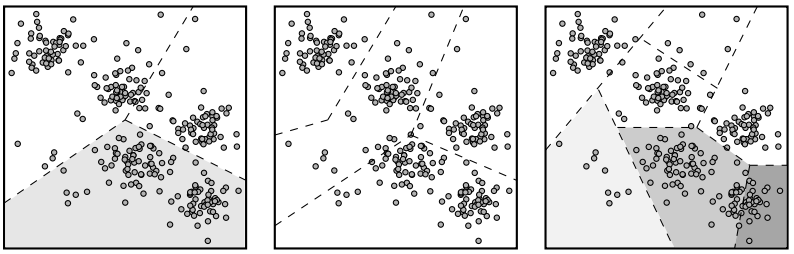
Ce processus itère jusqu'à épuisement de la liste des conflits, c'est-à-dire que tous les conflits ont été résolus, ou que la résolution des conflits restants ne permet pas d'améliorer les résultats. L'algorithme 2 présente ce processus en détail. La figure 3.10 présente un ensemble de résultats avant et après une itération où un conflit est résolu.

Algorithme 2: Clustering collaboratif

soit $\breve{\mathbb{C}} = \{\mathbb{C}^i\}_{1 \leq i \leq m}$ l'ensemble initial de résultats de clustering
soit $\breve{K} = \texttt{conflicts}(\breve{\mathbb{C}})$ l'ensemble des conflits sur $\breve{\mathbb{C}}$ (équation (3.11))
soit $\breve{\mathbb{C}}^{\texttt{best}} = \breve{\mathbb{C}}$ la meilleure solution temporaire
soit $\breve{K}^{\texttt{best}} = \breve{K}$ les conflits de la meilleure solution temporaire
tant que $|\breve{K}| \geq 0$ **faire**
  $\mathcal{K}_k^{i,j} = \arg\max_{\mathcal{K}_l^{r,s} \in \breve{K}} CI\left(\mathcal{K}_l^{r,s}\right)$
  $\breve{\mathbb{C}} = \texttt{conflictResolution}\left(\breve{\mathbb{C}}, \mathcal{K}_k^{i,j}\right)$ (Algorithme 1)
  si $\mathbf{\Gamma}(\breve{\mathbb{C}}) > \mathbf{\Gamma}(\breve{\mathbb{C}}^{best})$ **alors**
    $\breve{\mathbb{C}}^{\texttt{best}} = \breve{\mathbb{C}}$
    $\breve{K}^{\texttt{best}} = \breve{K} = \texttt{conflicts}(\breve{\mathbb{C}})$
    $bt = 0$
  sinon si $\breve{\mathbb{C}}^{t+1} = \breve{\mathbb{C}}^t$ **alors**
    $\breve{K} = \breve{K} \setminus \mathcal{K}_k^{i,j}$
  sinon
    $bt := bt + 1$
    $\breve{K} = \breve{K} \setminus \mathcal{K}_k^{i,j}$
    si $bt > |\breve{K}|$ **alors**
      $\breve{\mathbb{C}} = \breve{\mathbb{C}}^{\texttt{best}}$
      $\breve{K} = \breve{K}^{\texttt{best}} \setminus \mathcal{K}_k^{i,j}$
calcul du consensus

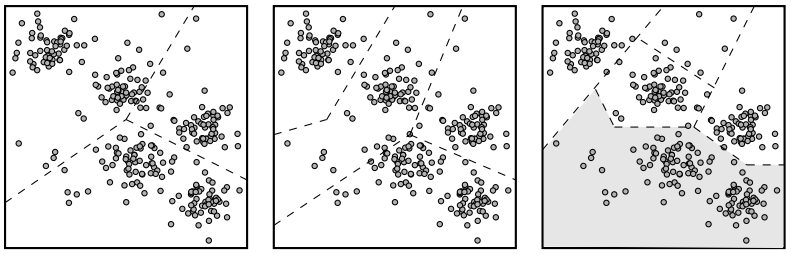
3.3.2.3 Combinaison des résultats raffinés

Après l'étape de raffinement, tous les résultats sont relativement similaires, avec un nombre de clusters proche. Chaque résultat peut être individuellement intéressant pour l'expert, comme chaque résultat a été créé d'une part grâce à son algorithme propre, et d'autre part par les informations obtenues des autres méthodes pendant la collaboration. Cependant, si l'expert est intéressé



(a) $\mathcal{C}^{(1)}$ (b) $\mathcal{C}^{(2)}$ (c) $\mathcal{C}^{(3)}$

Coefficient global d'évaluation avant résolution d'un conflit : $\mathbf{\Gamma} = 0.285$



(d) $\mathcal{C}^{(1)}$ (e) $\mathcal{C}^{(2)}$ (f) $\mathcal{C}^{(3')}$

Coefficient global d'évaluation après résolution d'un conflit : $\mathbf{\Gamma} = 0.318$

Fig. 3.10: *Trois résultats avant (première ligne) et après (seconde ligne) la résolution d'un conflit entre* $\mathcal{C}^{(1)}$ *et* $\mathcal{C}^{(3)}$.

par un résultat unique, représentant le mieux possible l'ensemble des résultats, il est possible d'appliquer des algorithmes d'ensemble clustering vue en section 3.2.1. Nous présentons ici l'une de ces méthodes développée par Wemmert et Gancarski [2002b] et qui utilise les concepts et les méthodes mis en place en clustering collaboratif.

Dans cet algorithme de vote, chaque résultat $\mathcal{C}^{(i)}$ effectue un vote pour chaque objet x. Il va tout d'abord voter pour le cluster $C_k^{(i)}$ auquel est associé x dans son résultat, ainsi que pour tous les clusters correspondants $\psi(C_k^{(i)}, \mathcal{C}^{(j)})$ dans les autres résultats de clustering impliqués dans la collaboration. Une fois tous les votes effectués pour chaque résultat et chaque objet, la valeur maximale indique le meilleur cluster pour l'objet x, par exemple $C_l^{(j)}$. Cela signifie que l'objet x doit être dans le cluster $C_l^{(j)}$ d'après l'opinion majoritaire des résultats.

Pour chaque objet x, un ensemble de vecteurs de vote est calculé :

$$\mathcal{V}(x) = \left\{ (v_1^{(i)}(x), \ldots, v_{K^{(i)}}^{(i)}(x)), 1 \leq i \leq N \right\} \tag{3.17}$$

où

$$v_k^{(i)}(x) = \sum_{j=1}^{N} vote(x, C_k^{(i)}, \mathcal{C}^{(j)}) \tag{3.18}$$

et

$$vote(x, C_k^{(i)}, \mathcal{C}^{(m)}) = \begin{cases} 1 & \text{si } (i = m \text{ et } x \in \mathcal{C}_k^i) \\ & \text{ou } x \in \psi(C_k^{(i)}, \mathcal{C}^{(m)}) \\ 0 & \text{sinon} \end{cases} \tag{3.19}$$

L'objet x est affecté au cluster $\breve{\mathcal{V}}$, défini comme :

$$\breve{\mathcal{V}}(x) = \arg\max_{C_k^{(i)}} v_k^{(i)}(x) \tag{3.20}$$

Le tableau 3.5 présente un exemple de l'application de cet algorithme de vote. Une fois les différents vecteurs de $\mathcal{V}$ calculés, le maximum est cherché pour assigner un cluster d'un résultat à

	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(3)}$
x_1	1	3	1
x_2	1	3	1
x_3	2	1	1
x_4	2	1	1
x_5	3	2	2
x_6	3	2	2

Résultats initiaux.

	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(3)}$
$\mathcal{V}(x_1)$	$\{(3,1,0),$	$(1,0,3),$	$(1,2,0)\}$
$\mathcal{V}(x_2)$	$\{(3,1,0),$	$(1,0,3),$	$(1,2,0)\}$
$\mathcal{V}(x_3)$	$\{(1,3,0),$	$(3,0,1),$	$(1,2,0)\}$
$\mathcal{V}(x_4)$	$\{(1,3,0),$	$(3,0,1),$	$(1,2,0)\}$
$\mathcal{V}(x_5)$	$\{(0,0,3),$	$(0,3,0),$	$(0,0,3)\}$
$\mathcal{V}(x_6)$	$\{(0,0,3),$	$(0,3,0),$	$(0,0,3)\}$

Votes pour les différents objets.

Objet	x_1	x_2	x_3	x_4	x_5	x_6
Résultat	1	1	2	2	3	3

Résultat du vote.

Tab. 3.5: *Exemple de l'application de l'algorithme de vote à la fin du processus collaboratif.*

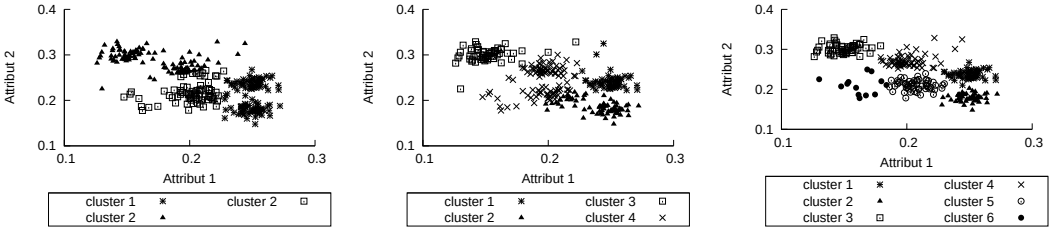
chaque objet (voir équation (3.20)). En cas d'égalité, le premier cluster trouvé pour un objet est conservé. Ces égalités rendent possible le fait que plusieurs clusters soient créés dans le résultat final mais qu'ils représentent en fait des clusters très similaires dans deux résultats différents. Il est possible d'utiliser des heuristiques pour réduire le nombre de clusters dans le résultat final, comme par exemple établir une correspondance entre les clusters très similaires et effectuer un ré-étiquetage en post-traitement.

3.3.3 Applications

Dans cette section nous allons présenter des applications de la méthode collaborative pour illustrer son fonctionnement et le type de résultats obtenus. Dans une première expérience nous avons utilisé la méthode collaborative sur un jeu de données à deux dimensions en utilisant l'algorithme KMEANS et en faisant varier le nombre de clusters en entrée ainsi que l'initialisation des centroïdes. Les figures 3.11 (a), (b) et (c) présentent les trois résultats initiaux obtenus sur le jeu de données. La méthode a été paramétrée pour trouver des résultats entre 2 et 10 clusters et la qualité a été évaluée en utilisant la compacité des clusters avec le critère de Wemmert et Gançarski (voir section 2.3). Les trois résultats ainsi calculés ont été utilisés en entrée de la méthode collaborative. À l'issue de la phase de collaboration, les trois résultats ont convergé vers un nombre de clusters égal à cinq. Les figures 3.11 (b), (c) et(d) présentent les trois résultats obtenus à l'issue de la collaboration. Les résultats obtenus sont fortement similaires et presque identiques à quelques objets près. Le tableau 3.6 présente la valeur des attributs des centroïdes pour les deux attributs avant et après collaboration. Les centroïdes n'ont pas été triés car l'ordre des clusters n'a a priori pas de sens. Il est cependant aisé de faire le lien entre les clusters des résultats.

Une expérience similaire a également été menée en utilisant l'algorithme EM. La figure 3.12 représente les gaussiennes trouvées pour le premier attribut par l'algorithme, avant ((a),(b) et (c)) et après ((b), (c) et (d)) la collaboration. On peut observer que les résultats initiaux sont nettement différents en fonction du nombre de gaussiennes sélectionnées. Enfin, les gaussiennes obtenues à la fin de la collaboration permettent de constater que les algorithmes ont atteint un consensus. En effet, les cinq gaussiennes trouvées sont fortement similaires.

De manière similaire, nous avons également fait collaborer trois instances de l'algorithme SOM. Les figures 3.13 (a), (b) et(c) présentent les trois cartes de neurones initiales, la première avec une carte 4x4, la seconde avec une carte 3x4 et enfin, la dernière avec une carte 4x2. Les figures 3.13 (b), (c) et (d) présentent les résultats obtenus à l'issue de l'application de la méthode collaborative. Les trois instances ont convergé vers un résultat similaire avec 6 neurones. Nous pouvons également constater que la forme rectangulaire de la carte, imposée à l'initialisation de l'algorithme, n'est plus respectée. Ceci s'explique par l'application des opérateurs (scission, fusion et reclustering) durant le processus collaboratif. Ces opérateurs modifient la structure de la carte et ne garantissent pas le respect de la topologie initiale.

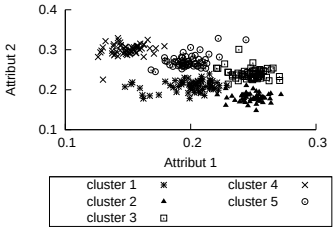


(a) $\mathcal{C}^{(1)}$ avec 3 clusters.

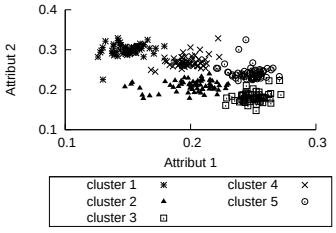
(b) $\mathcal{C}^{(2)}$ avec 4 clusters.

(c) $\mathcal{C}^{(3)}$ avec 6 clusters.

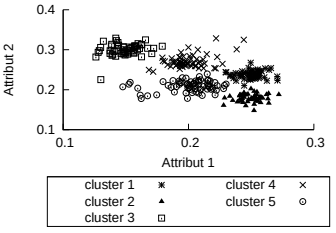
Résultats initiaux avant collaboration.



(d) $\mathcal{C}^{(1')}$ avec 5 clusters.



(e) $\mathcal{C}^{(2')}$ avec 5 clusters.



(f) $\mathcal{C}^{(3')}$ avec 5 clusters.

Résultats finaux après collaboration.

Fig. 3.11: *Résultats de clustering avant et après collaboration (algorithme* KMEANS*).*

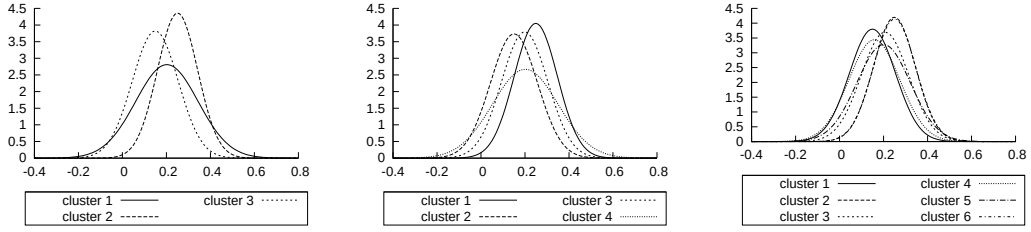
	$\mathcal{C}^{(1)}$		$\mathcal{C}^{(2)}$		$\mathcal{C}^{(3)}$	
	Att 1	Att 2	Att1	Att 2	Att 1	Att 2
C_1	0.2489	0.2094	0.2491	0.2412	0.2497	0.2375
C_2	0.1720	0.2881	0.2388	0.1859	0.2492	0.1789
C_3	0.2009	0.2166	0.1522	0.3004	0.1499	0.3016
C_4	-	-	0.1960	0.2430	0.2011	0.2717
C_5	-	-	-	-	0.2071	0.2102
C_6	-	-	-	-	0.1625	0.2100

Centroïdes des résultats avant la collaboration.

	$\mathcal{C}^{(1')}$		$\mathcal{C}^{(2')}$		$\mathcal{C}^{(3')}$	
	Att 1	Att 2	Att1	Att 2	Att 1	Att 2
C_1	0.1994	0.2091	0.1496	0.3002	0.2497	0.2375
C_2	0.2482	0.1801	0.2006	0.2088	0.2492	0.1789
C_3	0.2479	0.2397	0.2498	0.1812	0.1496	0.3002
C_4	0.1496	0.3002	0.1973	0.2701	0.2001	0.2709
C_5	0.1982	0.2711	0.2475	0.2423	0.2006	0.2088

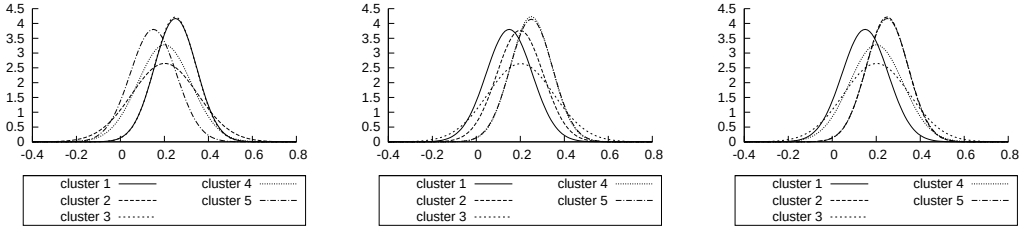
Centroïdes des résultats après la collaboration.

Tab. 3.6: *Centroïdes de* KMEANS *avant et après la collaboration.*



(a) $\mathcal{C}^{(1)}$ avec 3 clusters. (b) $\mathcal{C}^{(2)}$ avec 4 clusters. (c) $\mathcal{C}^{(3)}$ avec 6 clusters.

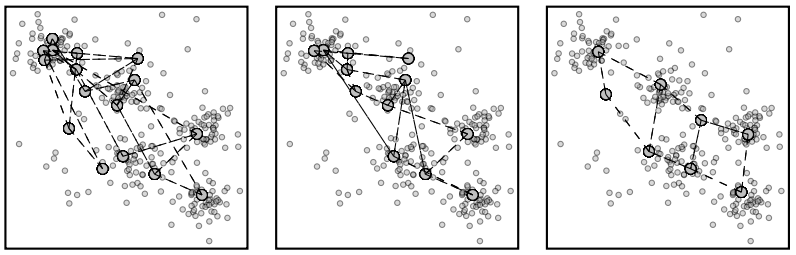
Gausiennes initiales avant collaboration.



(d) $\mathcal{C}^{(1')}$ avec 5 clusters. (e) $\mathcal{C}^{(2')}$ avec 5 clusters. (f) $\mathcal{C}^{(3')}$ avec 5 clusters.

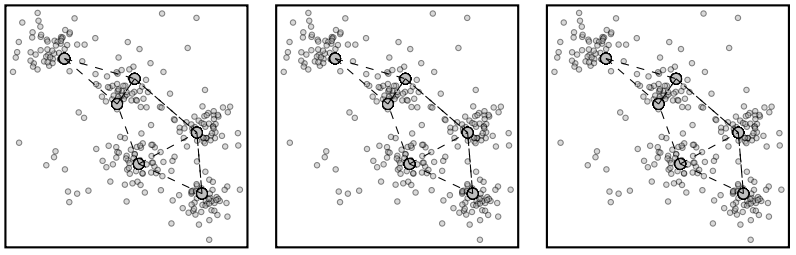
Gausiennes finales après collaboration.

Fig. 3.12: *Gausiennes avant et après la collaboration en utilisant l'algorithme* EM.



(a) SOM avec une carte 4x4. (b) SOM avec une carte 4x3. (c) SOM avec une carte 4x2.

Résultat initiaux avant collaboration.



(d) SOM avec 6 neurones. (e) SOM avec 6 neurones. (f) SOM avec 6 neurones.

Résultat finaux après collaboration.

Fig. 3.13: *Résultats avant et après collaboration avec l'algorithme* SOM.

3.4 Résolution de conflits en clustering collaboratif

Dans cette section, nous allons présenter nos propositions concernant la résolution de conflits en clustering collaboratif. De nouvelles stratégies de résolution de conflits développées lors de cette thèse vont être présentées.

3.4.1 Problématique

Comme nous l'avons vu dans la section précédente, le clustering collaboratif consiste à résoudre des conflits entre différents résultats de clustering en vue d'améliorer leur qualité et leur similarité. Pour résoudre ces conflits, la méthode SAMARAH propose de tenter de résoudre à chaque étape le conflit le plus important (voir section 3.3.2.2). Cette stratégie, bien que pouvant apparaître comme la plus efficace, peut mener à des solutions non optimales. En effet, en résolvant à chaque itération le conflit le plus important, il est possible que le système converge très rapidement vers un extremum local. La stratégie proposée, qui consiste à autoriser une perte de qualité pendant un certain nombre d'itérations permet de réduire ce risque, mais n'assure toujours pas un parcours optimal de l'espace de recherche. De plus, il peut être difficile pour l'expert de choisir le nombre d'itérations pendant lesquelles la qualité peut diminuer. Rappelons que la détection de conflits s'effectue au niveau local, c'est-à-dire entre couples de résultats de clustering. L'information locale (la répartition des objets dans les clusters d'un couple de résultats) permet de choisir le couple de résultats qui est le plus en désaccord, et par conséquent celui qui tirera le mieux parti de la résolution d'un conflit. Cependant, la validation au niveau global est sujette à une amélioration globale de la solution, c'est-à-dire une amélioration de l'ensemble des couples, ou au moins d'une partie importante. Il est donc difficile de s'assurer que les modifications faites au niveau local vont améliorer la solution globalement.

Il pourrait être envisagé de calculer l'évaluation globale lors de la résolution de chaque conflit. Ceci permettrait de s'assurer que la résolution du conflit considéré au niveau local permet la meilleure amélioration au niveau globale. Cependant, cette évaluation globale nécessite un nombre de calculs important car le coefficient local d'évaluation (voir équation (3.13)) est calculé pour chaque couple de résultats et est ensuite moyenné. On obtient donc $4(N(N-1))$ (N étant le nombre de résultats) calculs à effectuer pour choisir parmi les quatre couples résultant de la tentative de résolution d'un conflit. Il ne parait donc pas raisonnable de calculer ce coefficient pour chaque possibilité de résolution de conflit, surtout si le nombre de résultats impliqués dans le processus collaboratif est important.

En étudiant plus en détails l'espace des solutions possibles en clustering collaboratif, on se rend rapidement compte de l'importance du nombre potentiel de solutions. Soit $\breve{\mathbb{C}}$ l'ensemble de toutes les combinaisons de résultats de clustering possibles de N résultats de clustering des données X. L'objectif est de trouver à partir d'un ensemble $\mathbb{C} \in \breve{\mathbb{C}}$, l'ensemble $\mathbb{C}\star$ tel que :

$$\mathbb{C}\star = \arg\max_{\mathbb{C}\in\breve{\mathbb{C}}}(\mathbf{\Gamma}(\mathbb{C})) \tag{3.21}$$

En fonction du nombre de résultats de clustering impliqués dans la collaboration, l'espace de recherche, c'est-à-dire le nombre de solutions possibles, peut croître très rapidement. En effet, le nombre de clusterings possibles pour un jeu de données étant déjà très élevé (voir section 3.2.1.2), le fait de faire collaborer plusieurs méthodes fait encore augmenter cette complexité. Soient p le nombre de clusterings possibles et N le nombre de clusterings prenant part dans la collaboration, on obtient p^N solutions potentielles pour $\mathbb{C}\star$.

Dans cette section, nous allons étudier le problème de la résolution de conflit en clustering collaboratif. Différentes stratégies de résolution de conflits sont présentées. Nous étudions tout d'abord un premier groupe de stratégies, appelées stratégies itératives qui consistent toujours à sélectionner un conflit candidat et son éventuelle résolution dans un processus itératif. Chaque stratégie propose une manière différente d'effectuer le choix du conflit candidat. Ces stratégies conservent en grande partie les principes de la méthode SAMARAH. Seul l'ordre dans lequel les

conflits sont résolus est modifié. Par contre, dans la dernière stratégie développée lors de cette thèse, le clustering collaboratif est étudié comme un processus d'optimisation. Dans cette stratégie, une adaptation d'un algorithme génétique est utilisée pour optimiser la fonction objective $\Gamma(\mathbb{C})$. Cette approche modifie plus profondément la méthode collaborative existante et propose un formalisme et une approche totalement nouveaux.

3.4.2 Approche itérative pour la résolution de conflits

Lors de la description de la méthode collaborative SAMARAH, nous avons vu qu'il était nécessaire de définir l'ordre dans lesquels les conflits sont considérés. Nous allons maintenant présenter nos propositions pour explorer d'autres stratégies que la sélection du conflit le plus important. De nombreuses stratégies ont été explorées, nous présentons ici les plus intéressantes. Rappelons que les conflits sont évalués en fonction de la similarité des clusters des résultats. Un conflit implique le cluster d'un résultat et des clusters d'un autre résultat. Á noter que quelque soit la stratégie adoptée, le conflit sélectionné sera résolu de la même manière et son résultat ne sera pris en compte que s'il améliore le résultat global.

3.4.2.1 Conflit le plus important (WCC)

Cette stratégie (appelée WCC) consiste à sélectionner le conflit le plus important, c'est-à-dire celui qui implique le couple de résultats ayant le conflit avec un coefficient de conflit le plus élevé (voir équation (3.12)). C'est la stratégie utilisée dans la version originelle de SAMARAH.

Dans ce cas, la sélection du conflit $\mathcal{K}$ à résoudre peut s'écrire :

$$\mathcal{K} := \arg \max_{\mathcal{K}_{(i)} \in \breve{\mathcal{K}}} CI\left(\mathcal{K}_{(i)}\right) \tag{3.22}$$

La figure 3.14 illustre la mise en œuvre de ce type de sélection sur un exemple de liste de conflits. La probabilité de sélection des conflits est de 1 pour le premier conflit et de 0 pour les autres conflits de la liste. Si on considère la liste des conflits triés par leur importance associé croissante, seul le premier conflit peut être sélectionné comme candidat à la résolution.

3.4.2.2 Conflit aléatoire (RCC)

Cette stratégie (appelée RCC) consiste à considérer que tous les conflits ont le même potentiel d'amélioration au niveau global. Elle ne tient pas compte de l'importance du conflit dans la sélection du conflit à résoudre et choisit un conflit aléatoirement dans la liste des conflits.

Dans ce cas, la sélection de conflit peut s'écrire :

$$\mathcal{K} := \texttt{aléatoire}\left(\breve{\mathcal{K}}\right) \tag{3.23}$$

La fonction `aléatoire` sélectionne un conflit aléatoirement dans la liste des conflits. À la manière de la stratégie mise en place lors de la sélection du conflit le plus important, ce conflit aléatoirement choisi n'est appliqué que si il améliore la similarité globale de la solution. Par conséquent, la probabilité $p(\mathcal{K}_{(i)})$ que le i-ème conflit soit sélectionné est de :

$$p(\mathcal{K}_{(i)}) = \frac{1}{N_c} \tag{3.24}$$

avec N_c le nombre de conflits. La figure 3.15 illustre la mise en œuvre de ce type de sélection sur un exemple.

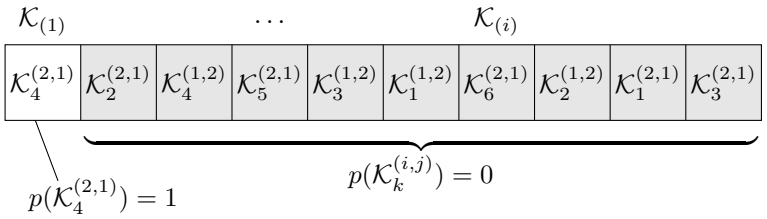


Fig. 3.14: *Sélection du conflit le plus important (WCC).*

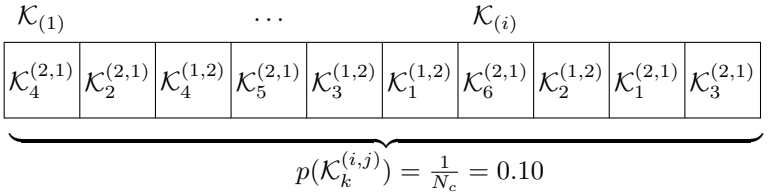


Fig. 3.15: *Sélection aléatoire du conflit à résoudre (RCC).*

3.4.2.3 Pondération des conflits (R-WCC)

Cette approche (appelée R-WCC) utilise le principe de la roue de la roulette qui consiste à effectuer une sélection proportionnelle à l'importance du conflit. Cette approche est inspirée des méthodes de sélection développées pour la sélection des solutions dans les algorithmes génétiques [Goldberg, 1989] (voir Annexe C). L'importance du conflit est utilisée pour associer une probabilité de sélection à chaque conflit. Dans ce cas, la probabilité de sélection d'un conflit peut s'écrire :

$$p(\mathcal{K}_{(i)}) = \frac{CI\left(\mathcal{K}_{(i)}\right)}{\sum_{j=1}^{N_c} CI\left(\mathcal{K}_{(j)}\right)} \tag{3.25}$$

avec N_c le nombre de conflits. Bien que les conflits ayant une importance élevée aient plus de chance d'être sélectionnées, il y a toujours une possibilité que des conflits moins importants soient sélectionnés.

Dans ce cas, la sélection de conflit peut s'écrire :

$$\mathcal{K} := \mathcal{K}_{(i)} | \left(\sum_{j=0}^{i-1} p(\mathcal{K}_{(j)}) \leq \nu \wedge \sum_{k=0}^{i+1} p(\mathcal{K}_{(k)}) > \nu \right) \tag{3.26}$$

où ν est un réel aléatoire dans $[0; 1]$. La figure 3.16 illustre la mise en œuvre de ce type de sélection sur un exemple de liste de conflits.

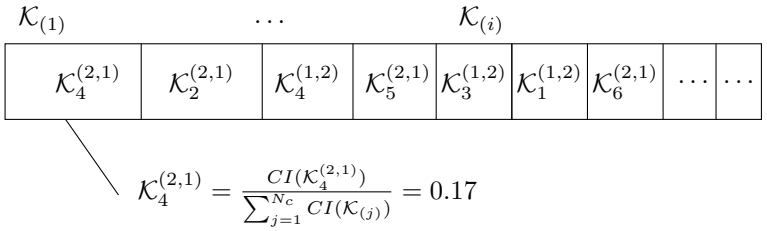


Fig. 3.16: *Sélection pondérée du conflit à résoudre (R-WCC).*

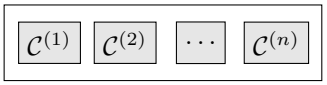


Fig. 3.17: *Représentation d'une solution dans le cadre de l'algorithme génétique composée de plusieurs résultats de clustering.*

3.4.2.4 Conclusion

Nous avons vu dans cette section différentes stratégies pour sélectionner des conflits à résoudre lors du processus collaboratif. Ces différentes stratégies consistent toujours en la sélection d'un conflit et sa résolution, et cela de manière itérative jusqu'à avoir résolu l'ensemble des conflits. Dans la section suivante nous allons voir une autre stratégie, légèrement différente, qui utilise un algorithme génétique. Nous comparons ensuite ces différentes stratégies sur des jeux de données artificiels et de l'UCI.

3.4.3 Approche évolutionnaire pour la résolution de conflits

3.4.3.1 Problématique

La classification collaborative peut être vue comme la recherche, à partir d'un ensemble de résultats de clustering, d'un autre ensemble de résultats de clustering qui maximise la similarité ainsi que la qualité des résultats (voir équation 3.21). Pour trouver ce nouvel ensemble de résultats nous allons voir dans cette section comment nous avons utilisé et adapté un algorithme génétique. Cette stratégie (appelée GR) implique de redéfinir complètement l'approche collaborative et bien que les concepts et les critères d'évaluation soient conservés, la stratégie mise en place pour la recherche d'une solution maximisant la similarité et la qualité des résultats est totalement différente et innovante.

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnaires. Leur but est d'obtenir une solution approchée à un problème d'optimisation, lorsqu'il n'existe pas (ou qu'on ne connaît pas) de méthode exacte pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle développée au $20^{\text{ème}}$ siècle par le scientifique Charles Darwin et l'appliquent à une population de solutions potentielles au problème donné. Cette population évolue au cours des générations, ce qui va simuler l'évolution de la population au fil du temps. Au cours de ces générations, les solutions qui répondent le mieux au problème sont conservées et sont utilisées pour en construire de nouvelles.

3.4.3.2 Formalisation

Dans le cadre du clustering collaboratif, une solution dans notre algorithme génétique sera un ensemble de résultats de clustering $\mathbb{C}$ (voir figure 3.17). Contrairement aux approches itératives, nous allons manipuler un ensemble d'ensembles de résultats. Au lieu de faire évoluer un ensemble de résultats qui est modifié itérativement en résolvant un conflit à la fois, nous allons utiliser plusieurs ensembles de résultats qui vont évoluer conjointement et vont pouvoir mieux explorer l'espace des solutions. Ces différentes solutions vont évoluer au cours de générations de l'algorithme génétique. Elles vont être utilisées pour créer de nouvelles solutions qui répondent mieux au problème, c'est-à-dire dont la similarité et la qualité sont supérieures. Il est donc important de noter qu'une solution est un ensemble de résultats de clustering et non pas un unique clustering.

Pour initialiser l'algorithme génétique, il est nécessaire de générer une population initiale qui va servir de base pour la création de nouvelles solutions au cours des générations. Cette population est composée d'un ensemble de solutions au problème considéré. Pour créer cette population initiale, nous avons fait le choix de ne partir que de la solution disponible initialement dans les approches itératives, et de modifier celle-ci par l'application d'opérateurs paramétrés aléatoirement sur les résultats la composant. À partir de la solution initialement disponible, un ensemble d'opérateurs

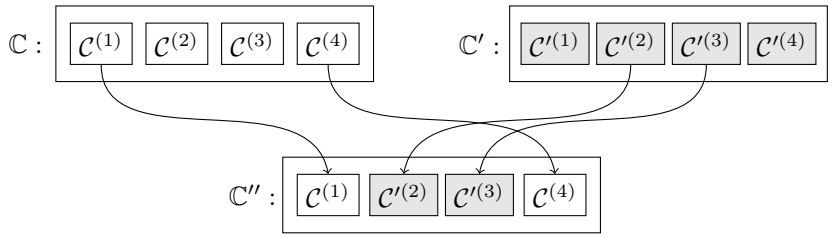


Fig. 3.18: *Illustration de l'opérateur de croisement.*

(scission, fusion et reclustering) est appliqué aux différents résultats pour créer des solutions alternatives. Ce choix a été fait pour pouvoir se comparer de manière équitable avec les approches itératives qui ne considèrent qu'une solution initiale.

Une fois cette population créée, l'algorithme génétique repose sur les étapes suivantes qui représentent la transition entre les générations :

1. *évaluation des solutions de la population :* chaque solution est évaluée grâce à son coefficient global d'évaluation défini dans SAMARAH (voir équation (3.15)). Cette mesure reflète la similarité et la qualité des résultats et est utilisée pour juger de la qualité d'une solution. Plus cette valeur est élevée, meilleure est la solution.
2. *sélection des solutions par rapport à leur évaluation* : une solution avec un fort coefficient global d'évaluation a plus de chance d'être sélectionnée qu'une solution avec une valeur plus faible. La probabilité de sélection d'une solution est donc proportionnelle à sa qualité.
3. *croisement :* l'opérateur de croisement est utilisé pour créer de nouvelles solutions à partir des solutions existantes dans la population. Deux solutions (appelées *parents*) sont sélectionnées et donnent naissance à une nouvelle solution en mélangeant les résultats de clustering présents dans les deux solutions (*slicing cross-over*). La figure 3.18 illustre ce processus de croisement. Il est important de noter que l'ordre des résultats dans chaque solution est important et doit être conservé. En effet, chaque résultat est lié à la méthode utilisée pour générer ce résultat particulier, qui est ensuite utilisé pour résoudre les conflits. Lors de la création de la nouvelle solution, à chaque rang, il est équiprobable de sélectionner un résultat du premier parent ou du deuxième parent.
4. *mutation :* à chaque génération, un conflit sélectionné aléatoirement dans la liste des conflits associé à un résultat selon une probabilité fixée a priori par l'expert est résolu. Ainsi, nous cherchons à éviter à l'algorithme d'être bloqué dans un optimum local et nous créons de nouvelles solutions en maintenant de la diversité au sein de la population. La meilleure solution de la population n'est pas affectée par l'opérateur de mutation car il est possible que l'application de cet opérateur diminue la qualité du résultat.

L'utilisation d'un algorithme génétique permet d'effectuer une recherche plus complète dans l'espace des solutions. En effet, beaucoup plus de solutions vont être explorées ce qui augmente les chances de trouver une solution proposant une meilleure similarité ainsi qu'une meilleure qualité. Cependant, comme beaucoup plus de solutions vont être manipulées, cette approche est plus gourmande en mémoire et en temps que les approches itératives.

Pour améliorer la rapidité de la convergence de l'algorithme génétique, nous avons utilisé le principe dit de l'optimisation *Lamarckienne* [Blansché, 2006]. Cette optimisation consiste, à chaque génération, à appliquer à chaque solution une optimisation locale qui a pour but d'améliorer légèrement la solution. Dans notre cas, l'opération à appliquer à chaque solution est la résolution du conflit le plus important. Si la résolution de conflit améliore la solution, alors la modification est conservée et elle est appliquée à la solution. En utilisant cette approche l'ensemble des solutions s'améliore légèrement à chaque génération. Ce processus permet d'améliorer la rapidité de convergence de l'algorithme génétique. Néanmoins, il est toujours difficile de trouver un bon compromis entre rapidité de convergence et maintien de la diversité dans la population. Cependant, d'après

Jeux de données	Classes	Attributs	Objets
2d-4c-no0	4	2	1070
2d-4c-no1	4	2	3080
2d-4c-no2	4	2	1070
2d-4c-no3	4	2	1130
10d-4c-no0	4	10	1290
10d-4c-no1	4	10	958
iris	3	4	150
wine	3	13	178
ionosphere	2	34	351
segment	7	19	2310
vehicle	4	18	846

Tab. 3.7: *Informations sur les différents jeux de données utilisés dans les expériences.*

nos expériences, l'utilisation de l'approche Lamarckienne est bénéfique et permet d'améliorer les performances de l'algorithme sans perdre de qualité au niveau des solutions.

3.4.4 Comparaison des différentes stratégies

Dans cette section, nous allons mener des expériences pour comparer les différentes stratégies présentées dans les sections précédentes dans le but d'étudier leur comportement pour le traitement de données synthétiques et réelles.

Jeux de données utilisés

Nous avons utilisé des jeux de données fournis avec le logiciel *Cluster Generators* [1] qui propose un ensemble de jeux de données présentant différentes caractéristiques en terme de nombre de clusters, nombre d'attributs et nombre d'objets. Les 6 jeux de données sélectionnés contiennent chacun 4 clusters de forme gaussienne décrits par deux attributs pour 4 jeux de données (*2d-4c-no0*, *2d-4c-no1*, *2d-4c-no2*, *2d-4c-no3*) et 10 attributs pour 2 jeux de données (*10d-4c-no0*, *10-4c-no1*).

Nous avons également utilisé des données de la base de données de l'UCI [Asuncion et Newman, 2007]. Dans tous les jeux de données sélectionnés les objets sont décrits par des attributs à valeurs réelles. Le tableau 3.7 résume les informations sur les jeux de données utilisés lors des expériences.

Critères d'évaluation

Nous avons utilisé différents critères supervisés d'évaluation de la qualité externe d'un résultat de clustering : l'indice de Rand, l'indice de Jaccard, l'indice de Folks & Mallows et la F-Mesure (voir Annexe B pour une description détaillée des indices). Le choix de plusieurs critères a été fait pour limiter le biais induit par le choix d'une unique mesure. L'information de l'appartenance des objets aux vraies classes dans les jeux de données a été utilisée pour calculer la valeur de ces indices par rapport aux résultats obtenus.

Configuration des méthodes

Dans ces expériences, nous avons utilisé l'algorithme KMEANS comme méthode de base. Cinq instances de l'algorithme ont été initialisées aléatoirement avec un nombre de clusters choisi aléatoirement entre 2 et 10 pour générer l'ensemble initial de résultats $\mathbb{C}$ ($N = 5$). Puis, les différentes stratégies de résolution de conflits vues dans les sections précédentes ont été appliquées. Chacune des stratégies a été appliquée individuellement mais avec le même ensemble de résultats $\mathbb{C}$ en entrée. Dans le cadre de la stratégie génétique (GR), nous avons utilisé les paramètres suivant : une

[1] `http://dbkgroup.org/handl/generators/`

population de 20 individus, un taux de mutation égal à 10% et un nombre de générations égal à 100. Ce nombre de générations a été choisi empiriquement, les expériences ayant montré que la qualité des résultats n'augmentait plus par la suite. Le taux relativement élevé de mutation est motivé par le fait que la population initiale est créée à partir d'un unique ensemble de résultats de clustering. Ce fort taux de mutation est utilisé pour augmenter la diversité au sein de la population.

Ces expériences ont été menées 100 fois pour chaque jeu de données et les résultats ont été moyennés. L'ensemble des résultats est présenté dans le tableau 3.8. Les valeurs dans ce tableau correspondent aux moyennes parmi les cinq méthodes à la fin du processus de collaboration, les écarts-types étant présentés entre parenthèses. Le coefficient global d'évaluation (voir équation 3.15), qui reflète la similarité et la qualité des résultats est également présenté en $2^{\text{ième}}$ colonne (Γ). Il montre à quel point la stratégie a réussi à optimiser la qualité et la similarité des résultats. En plus des cinq stratégies comparées, les valeurs des évaluations des indices de qualité ont également été calculés avant l'application des stratégies. Ces valeurs sont données par les lignes nommées BRUT dans le tableau.

Analyse des résultats

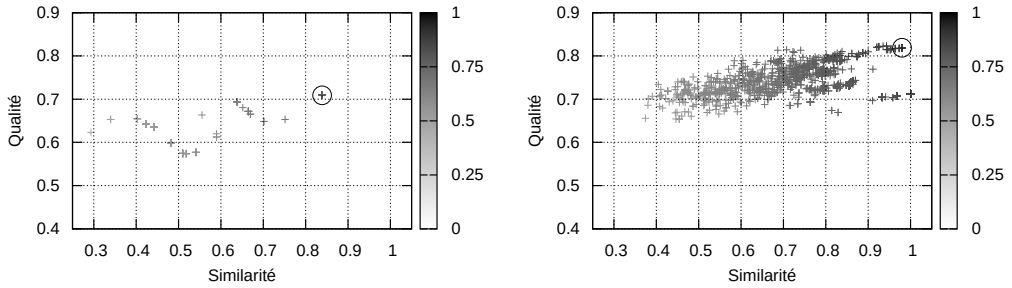
Les résultats obtenus indiquent que l'approche utilisant un algorithme génétique (GR) donne presque toujours les meilleurs résultats. Sur la quasi-totalité des jeux de données, la sélection aléatoire des conflits (RCC) donne les moins bons résultats, ce qui corrobore le fait que l'ordre dans lequel sont sélectionnés les conflits a une importance. De plus, la stratégie consistant à pondérer le choix des conflits par l'importance de ceux-ci (R-WCC) donne de bons résultats, d'ailleurs souvent meilleurs que la stratégie consistant à sélectionner toujours le conflit le plus important (WCC). Ceci peut être expliqué par le fait que la stratégie qui sélectionne le conflit le plus important (WCC) est plus susceptible d'être bloqué dans un optimum local. Les perturbations sont souvent plus importantes et éloignent le couple de résultats des autres résultats. Au contrainte, la stratégie avec pondération des conflits (R-WCC) a des chances d'éviter ce problème en sélectionnant des conflits dont l'importance locale est inférieure mais qui au final donne de meilleures améliorations au niveau global. La stratégie utilisant l'algorithme génétique (GR) produit de très bons résultats car celle-ci parcourt l'espace des solutions plus en détail. Elle trouve même la partition idéale (c'est-à-dire correspondant exactement aux classes) pour deux jeux de données (*10d-4c-no0* et *10-4c-no1*).

Les résultats obtenus sur les jeux de données artificiels sont globalement identiques à ceux obtenus sur les données provenant de l'UCI, notamment sur *iris, wine* et *segment* où la stratégie génétique (GR) donne les meilleurs résultats. Sur ces jeux de données réels, l'écart est moins net entre les résultats produits sans pondération de conflit (WCC) et avec pondération de conflit (R-WCC), la stratégie du conflit le plus important donnant même de meilleurs résultats sur *iris* et *segment*. On note également que le coefficient global d'évaluation (voir équation 3.15) est fortement corrélé à la qualité des résultats obtenus. En effet, plus celui-ci est élevé, meilleurs sont les résultats.

Ces résultats montrent que globalement l'importance des conflits (voir équation 3.12) est un indicateur intéressant pour choisir quel conflit résoudre. Les résultats révèlent également que résoudre de manière aléatoire les conflits n'est pas judicieux. Enfin, la stratégie proposée qui consiste à utiliser un algorithme génétique (GR) permet d'améliorer la découverte de meilleures solutions.

Visualisation des solutions explorées

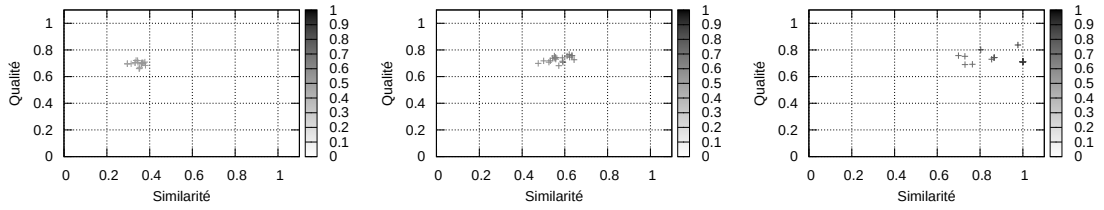
La stratégie utilisant un algorithme génétique (GR) explore mieux l'espace des solutions en évaluant plus de solutions potentielles que les stratégies itératives. Pour illustrer ce phénomène nous avons représenté graphiquement les solutions explorées par la stratégie du conflit le plus important (WCC) ainsi que pour la stratégie génétique (GR) sur une expérience sur le jeu de données *iris*. Les résultats sont présentés sur la figure 3.19 où chaque point représente une solution (c'est-à-dire un ensemble de résultats de clustering $\mathbb{C}$) en fonction de sa similarité moyenne et sa qualité moyenne (c'est-à-dire les deux composantes du coefficient global d'évaluation (voir équation (3.15)). On peut voir sur cette figure que la stratégie génétique évalue bien plus de solutions potentielles et est donc plus encline à trouver une meilleure solution.



(a) Solutions explorées par la stratégie du conflit le plus important (WCC).

(b) Solutions explorées par la stratégie utilisant un algorithme génétique (GR).

Fig. 3.19: *Solutions explorées sur une exécution sur le jeu de données iris avec la stratégie WCC (a) et la stratégie GR (b). La meilleure solution trouvée est encerclée.*



(a) 1ère génération

(b) 50ème génération

(c) 100ème génération

Fig. 3.20: *État de la population de la stratégie utilisant un algorithme génétique à différentes générations.*

La figure 3.20 illustre, lors d'une autre expérience, l'état de la population de l'algorithme génétique à la 1ère, 50ème et 100ème génération, en représentant les différentes solutions en fonction de leur qualité et de leur similarité. À la 1ère génération, la diversité des individus est relativement faible. Ceci est dû au fait que tous les individus de la population initiale sont générés à partir du même ensemble de résultats de clustering. Au cours des générations, les individus de la population vont accroître leur qualité et leur similarité, comme cela est visible à la 50ème génération. Enfin, plus on progresse dans les générations, meilleures sont les solutions composant la population.

Les bons résultats de la méthode génétique (GR) sont à relativiser par rapport à sa complexité en temps et en mémoire. Comme nous l'avons déjà mentionné, l'algorithme génétique explore bien plus de solutions que les méthodes itératives, et est par conséquent plus consommateur de ressources. Dans toutes nos expériences, les trois solutions itératives ont pris sensiblement le même temps d'exécution (de 1 à 100 secondes) en fonction du jeu de données, alors que l'algorithme génétique prenait environ 10 fois plus de temps (10 à 1000 secondes). Ces résultats ont été obtenus sur un processeur Intel Core2Duo 6300 avec 4GB de mémoire vive et sans optimisation du code. Il est important de noter que ces temps d'exécution pourraient être améliorés, par exemple, en parallélisant la méthode génétique.

Il serait également envisageable d'utiliser un algorithme génétique multiobjectif pour tenter d'optimiser parallèlement les deux objectifs (qualité et similarité) qui pour l'instant sont optimisés conjointement dans un unique critère. Cependant, ce type d'algorithme ne produira pas une solution unique mais un ensemble de solutions qui seront des compromis entre qualité et similarité des résultats. Les solutions étant elles-mêmes des ensembles de résultats, ce type d'approche imposerait à l'expert de faire de nombreux choix à la fin du processus collaboratif.

	Stratégie	Γ	Rand	Jaccard	Folks & Mallows	F-Mesure
2d-4c-no0	BRUT	0,655	0,898 (±0,020)	0,667 (±0,015)	0,805 (±0,010)	0,786 (±0,013)
	WCC	0,913	0,961 (±0,019)	0,874 (±0,037)	0,932 (±0,022)	0,930 (±0,025)
	RCC	0,908	0,956 (±0,019)	0,846 (±0,061)	0,916 (±0,036)	0,914 (±0,039)
	R-WCC	0,941	0,972 (±0,004)	0,896 (±0,015)	0,945 (±0,009)	0,945 (±0,009)
	GR	**0,956**	**0,975 (±0,009)**	**0,907 (±0,033)**	**0,951 (±0,018)**	**0,951 (±0,018)**
2d-4c-no1	BRUT	0,677	0,873 (±0,019)	0,577 (±0,035)	0,743 (±0,023)	0,726 (±0,029)
	WCC	0,826	0,889 (±0,018)	0,648 (±0,039)	0,789 (±0,025)	0,784 (±0,030)
	RCC	0,828	0,854 (±0,090)	0,622 (±0,090)	0,775 (±0,059)	0,763 (±0,076)
	R-WCC	0,837	0,894 (±0,023)	0,662 (±0,047)	0,799 (±0,030)	0,794 (±0,037)
	GR	**0,887**	**0,912 (±0,006)**	**0,700 (±0,015)**	**0,824 (±0,010)**	**0,823 (±0,011)**
2d-4c-no2	BRUT	0,613	0,845 (±0,028)	0,555 (±0,041)	0,728 (±0,029)	0,710 (±0,033)
	WCC	0,861	0,887 (±0,141)	0,751 (±0,186)	0,855 (±0,122)	0,842 (±0,146)
	RCC	0,745	0,825 (±0,059)	0,578 (±0,072)	0,746 (±0,049)	0,727 (±0,058)
	R-WCC	0,854	0,889 (±0,047)	0,689 (±0,098)	0,814 (±0,065)	0,807 (±0,069)
	GR	**0,907**	**0,923 (±0,055)**	**0,762 (±0,112)**	**0,863 (±0,075)**	**0,860 (±0,080)**
2d-4c-no3	BRUT	0,707	0,921 (±0,015)	0,702 (±0,056)	0,827 (±0,035)	0,812 (±0,041)
	WCC	0,955	0,979 (±0,001)	0,921 (±0,002)	0,959 (±0,001)	0,959 (±0,001)
	RCC	0,945	0,978 (±0,005)	0,916 (±0,019)	0,956 (±0,010)	0,956 (±0,010)
	R-WCC	0,955	0,980 (±0,002)	0,924 (±0,006)	0,961 (±0,003)	0,961 (±0,003)
	GR	**0,969**	**0,983 (±0,000)**	**0,933 (±0,001)**	**0,965 (±0,001)**	**0,965 (±0,001)**
10d-4c-no0	BRUT	0,508	0,866 (±0,023)	0,610 (±0,040)	0,772 (±0,025)	0,753 (±0,030)
	WCC	0,824	0,935 (±0,060)	0,833 (±0,145)	0,907 (±0,082)	0,898 (±0,090)
	RCC	0,761	0,891 (±0,027)	0,730 (±0,051)	0,847 (±0,030)	0,833 (±0,034)
	R-WCC	0,874	0,972 (±0,055)	0,929 (±0,142)	0,961 (±0,078)	0,957 (±0,086)
	GR	**0,896**	**1,000 (±0,000)**	**1,000 (±0,000)**	**1,000 (±0,000)**	**1,000 (±0,000)**
10d-4c-no1	BRUT	0,528	0,901 (±0,035)	0,705 (±0,028)	0,830 (±0,018)	0,814 (±0,021)
	WCC	0,857	0,978 (±0,026)	0,940 (±0,068)	0,966 (±0,038)	0,964 (±0,041)
	RCC	0,842	0,975 (±0,047)	0,928 (±0,127)	0,960 (±0,071)	0,958 (±0,076)
	R-WCC	0,868	0,998 (±0,000)	0,993 (±0,002)	0,997 (±0,001)	0,997 (±0,001)
	GR	**0,874**	**1,000 (±0,000)**	**1,000 (±0,000)**	**1,000 (±0,000)**	**1,000 (±0,000)**
iris	BRUT	0,632	0,811 (±0,027)	0,505 (±0,060)	0,677 (±0,047)	0,656 (±0,053)
	WCC	0,745	0,849 (±0,171)	0,716 (±0,131)	0,832 (±0,087)	0,825 (±0,109)
	RCC	0,723	0,878 (±0,055)	0,717 (±0,086)	0,831 (±0,059)	0,829 (±0,062)
	R-WCC	0,730	0,887 (±0,038)	0,730 (±0,056)	0,842 (±0,037)	0,840 (±0,040)
	GR	**0,793**	**0,912 (±0,048)**	**0,785 (±0,067)**	**0,880 (±0,040)**	**0,877 (±0,047)**
wine	BRUT	0,448	0,804 (±0,024)	0,494 (±0,074)	0,666 (±0,057)	0,644 (±0,065)
	WCC	0,744	0,870 (±0,079)	0,715 (±0,098)	0,833 (±0,063)	0,828 (±0,073)
	RCC	0,704	0,835 (±0,078)	0,661 (±0,100)	0,795 (±0,065)	0,788 (±0,073)
	R-WCC	0,688	0,869 (±0,031)	0,686 (±0,055)	0,810 (±0,039)	0,807 (±0,042)
	GR	**0,794**	**0,930 (±0,007)**	**0,810 (±0,017)**	**0,895 (±0,010)**	**0,895 (±0,010)**
segment	BRUT	0,499	0,814 (±0,045)	0,363 (±0,022)	0,554 (±0,014)	0,530 (±0,026)
	WCC	0,746	0,872 (±0,008)	0,394 (±0,025)	0,565 (±0,025)	0,564 (±0,025)
	RCC	0,581	0,829 (±0,037)	0,377 (±0,022)	0,564 (±0,012)	0,545 (±0,024)
	R-WCC	0,596	0,818 (±0,057)	0,371 (±0,030)	0,560 (±0,018)	0,539 (±0,034)
	GR	**0,759**	**0,876 (±0,001)**	**0,397 (±0,006)**	**0,568 (±0,006)**	**0,568 (±0,006)**
ionosphere	BRUT	0,540	0,577 (±0,007)	0,339 (±0,017)	0,524 (±0,016)	0,503 (±0,020)
	WCC	0,720	0,583 (±0,004)	**0,417 (±0,017)**	0,590 (±0,015)	**0,589 (±0,017)**
	RCC	0,669	0,585 (±0,006)	0,394 (±0,023)	0,571 (±0,019)	0,564 (±0,024)
	R-WCC	0,700	0,589 (±0,009)	0,402 (±0,018)	0,578 (±0,016)	0,573 (±0,020)
	GR	**0,753**	**0,610 (±0,019)**	0,415 (±0,012)	**0,594 (±0,010)**	0,586 (±0,012)
vehicle	BRUT	0,545	0,514 (±0,089)	0,198 (±0,015)	0,335 (±0,020)	0,330 (±0,021)
	WCC	**0,859**	**0,683 (±0,017)**	**0,261 (±0,005)**	**0,451 (±0,017)**	**0,414 (±0,007)**
	RCC	0,759	0,580 (±0,124)	0,241 (±0,018)	0,410 (±0,046)	0,389 (±0,024)
	R-WCC	0,713	0,634 (±0,049)	0,235 (±0,021)	0,390 (±0,037)	0,381 (±0,027)
	GR	0,851	0,564 (±0,051)	0,253 (±0,019)	0,429 (±0,035)	0,404 (±0,026)

En gras : le meilleur résultat pour un jeu de données.

Tab. 3.8: *Évaluation des différentes stratégies de résolution de conflit sur les jeux de données artifiels et de l'UCI.*

3.5 Bilan

Dans ce chapitre nous avons présenté la problématique de l'utilisation de plusieurs méthodes et résultats de clustering. L'objectif est de tirer parti des informations fournies par différents acteurs, ou experts, pour améliorer la recherche d'une solution à un problème. Nous avons étudié les différentes propositions existantes pour combiner plusieurs résultats de clustering et produire une solution consensuelle. De plus, nous avons également étudié la méthode SAMARAH développée initialement par Cédric Wemmert lors de sa thèse. Le fonctionnement de cette méthode ainsi que sa mise en œuvre pour effectuer du clustering collaboratif ont été analysés en détail. Une évaluation de cette méthode a été proposée, permettant de mieux comprendre son fonctionnement. Enfin, nous avons étudié en détail le processus de résolution de conflits au sein de cette méthode et nous avons proposé des solutions alternatives et innovantes pour la résolution de conflits en clustering collaboratif. Une méthode basée sur un algorithme génétique a notamment été proposée, celle-ci permettant d'obtenir de meilleurs résultats que la stratégie initialement développée en clustering collaboratif.

Contributions et valorisation

Les recherches effectuées dans ce chapitre ont permis de mieux comprendre le processus de clustering collaboratif. Elles ont engendrés un ensemble de développement qui ont conduit à la mise en place d'une méthode générique de développement de stratégie de collaboration. Les différentes étapes du clustering collaboratif ont été décomposées et présentées en détail. Il en ressort une généricité importante, permettant le développement de nouvelles stratégies de collaboration. Ces propositions ont été validées par une publication [Forestier et al., 2010d] dans la conférence internationale *IEEE International Conference on Intelligent Systems*. Des perspectives intéressantes sont actuellement à l'étude pour évaluer d'autres types de stratégie comme la sélection par tournoi, ou encore la définition de nouveaux opérateurs spécifiques à chaque stratégie. Il serait en effet possible de spécialiser l'application des opérateurs en fonction du type de stratégie mis en place. Une étude plus poussée sur les paramètres de la méthode (nombre de méthodes impliquées, diversité dans les résultats, etc.) est également en cours dans le but d'étudier plus précisément leur influence.

Seconde partie :
Connaissances et clustering

Chapitre 4

Intégration de connaissances en clustering

Sommaire

4.1 Introduction

Les connaissances peuvent être définies de manière succincte comme *les choses sues* par un individu. La gestion des connaissances, c'est-à-dire la caractérisation de leur nature, leur acquisition, leur formalisme et leur stockage sont étudiés dans de nombreux domaines tels que la philosophie, l'épistémologie, les sciences cognitives, etc.

Dans le domaine de la gestion des connaissances, il est courant de faire la distinction entre les notions de *données*, *d'information* et enfin de *connaissance*. On pourra donner comme définition :

- une donnée est en général mesurable ;
  exemple : *Il fait* 15° *dans cette pièce.*
- une information correspond à une donnée contextualisée ;
  exemple : *Il fait froid dans cette pièce.*
- une connaissance correspond à l'appropriation et l'interprétation des informations.
  exemple : *Pour avoir chaud, il suffit de monter le chauffage.*

La connaissance peut être vue comme la sélection, l'appropriation et l'interprétation des informations par un expert humain. Le *savoir* est alors défini comme la mise en perspective à long terme des connaissances acquises.

La clustering étant, par définition, une approche dite non supervisée (contrairement à la classification), on peut se poser la question de la pertinence de l'ajout de connaissances dans ce type

d'approche. Cependant, il faut rappeler que malgré la puissance et l'intérêt du clustering pour structurer des données de manière automatique, celui-ci peut montrer ses limites dans des cas où les données sont complexes. Le principal problème étant que la structure que l'on peut découvrir dans les données grâce au clustering peut être trompeuse par rapport aux classes réellement cherchées par l'expert. On aura alors une chance de proposer de nouvelles informations, inconnues jusqu'alors de l'expert, mais l'on risque également de produire des résultats de mauvaise qualité, ne reflétant pas les attentes.

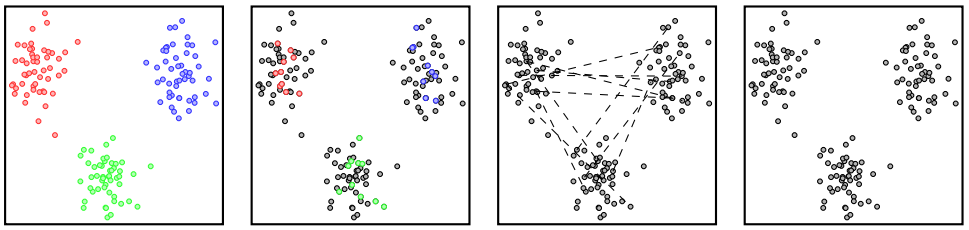
Prenons comme exemple un expert souhaitant créer des groupes au sein d'une population d'animaux. Un choix possible pour les décrire serait leur poids et leur taille. Cependant, si l'on prend comme exemple les chiens, leur poids peut varier d'une centaine de kilos à quelques centaines de grammes, et leur taille de deux mètres à quelques dizaines de centimètres. Même si l'on ajoute la couleur, la présence de queue ou d'autres critères, sans information additionnelle, les groupes créés automatiquement ont peu de chance de refléter les attentes de l'expert. Pour ce faire, il est nécessaire de disposer d'autres informations, ou des connaissances du domaine qui vont nous permettre de rendre plus efficace nos algorithmes de clustering et de les orienter vers de bons résultats. De plus, ces connaissances vont également nous permettre de faire un choix parmi plusieurs clusterings potentiels des données. En effet, si un choix est à faire, les connaissances peuvent nous guider vers une solution plus susceptible de refléter les attentes de l'expert.

Lorsque l'on différencie la classification supervisée et la classification non supervisée (clustering), l'élément clé réside dans la présence ou l'absence d'objets étiquetés permettant l'apprentissage d'une fonction de classification. Ces éléments étiquetés consistent en un ensemble d'objets dont on connait la classe ou l'étiquette. Ils sont généralement étiquetés manuellement par l'expert, ce qui est très coûteux en temps. Ils sont vus ici comme l'expression de la connaissance de l'expert. Dans la majorité des problèmes rencontrés, on dispose généralement d'un grand nombre d'objets non étiquetés et de pas ou très peu d'objets étiquetés. Dans les approches de fouille dites classiques, les méthodes supervisées ne s'intéressent qu'aux objets étiquetés et les méthodes non supervisées qu'aux objets non étiquetés.

Depuis quelques années, de nouvelles méthodes de classification ont fait leur apparition. Les termes les plus communément utilisés pour caractériser ces approches qui se trouvent entre les approches non supervisées et les approches supervisées sont *semi-supervisées*, *partiellement supervisées* ou encore *hybrides*. Sous ces termes se regroupent un ensemble de méthodes et d'approches qui utilisent à la fois des données non étiquetées et des données étiquetées ainsi que d'autres types de connaissances. Ces nouvelles méthodes apportent une solution aux problèmes pour lesquels l'utilisation de méthodes totalement automatiques n'est pas satisfaisant. Il convient alors de différencier deux types d'approches : la classification semi-supervisée et le clustering semi-supervisé. En classification semi-supervisée les objets étiquetés ainsi que les objets non étiquetés vont être utilisés pour construire une fonction de classification. Ainsi, l'objectif est d'utiliser les objets non étiquetés pour mieux saisir la configuration de l'espace des données. La classification est une opération inductive, dont l'objectif est de créer un classifieur qui généralise le modèle des données disponibles, et qui peut être utilisé par la suite pour traiter d'autres données. En clustering semi-supervisé, les objets étiquetés sont utilisés pour fournir des informations à l'algorithme de clustering qui va être guidé par ces connaissances dans sa recherche de clusters pertinents. Le clustering semi-supervisé est donc une opération transductive, car son objectif est de définir des clusters pour expliquer les données traitées, et éventuellement d'étiqueter les objets non étiquetés au départ.

Dans ce chapitre nous nous intéresserons à ce deuxième type d'approche (clustering semi-supervisé). Le lecteur intéressé par la classification semi-supervisée pourra se référer au rapport de recherche proposé par Zhu [2008] qui contient une introduction et un état de l'art mis à jour régulièrement [1]. Nous étudierons par la suite les différents types de connaissances disponibles et comment les algorithmes de clustering peuvent en tirer parti. Nous commencerons par un état de l'art des approches proposées dans la littérature ainsi que les différents formalismes de connaissances proposés. Nous présenterons et validerons ensuite nos propositions pour l'intégration de connaissances en clustering collaboratif.

[1] `http://pages.cs.wisc.edu/~jerryzhu/`



(a) Supervisé. (b) Partiellement étiqueté. (c) Partiellement contraint. (d) Non supervisé.

Fig. 4.1: *Exemple des différents types de connaissances selon Jain [2009].*

4.2 Intégration de connaissances en clustering

Les résultats parfois décevants des algorithmes de clustering face aux données de plus en plus complexes ont poussé la communauté scientifique à proposer de nouvelles approches permettant d'obtenir de meilleurs résultats sur ce type de données. Comme présenté dans le Chapitre 2, le processus de clustering est par définition une approche non supervisée, c'est-à-dire qu'il se base uniquement sur les données et n'utilise pas de connaissances. Cependant, sans aucune supervision, les algorithmes de clustering peuvent aboutir à des solutions non pertinentes. Les recherches se sont donc concentrées sur des approches permettant de guider le processus par des connaissances du domaine ou fournies par l'expert. L'objectif est de permettre à l'expert humain d'incorporer des connaissances du domaine dans le processus de fouille de données, et de le guider ainsi vers de meilleurs résultats. Plusieurs études [Anand et al., 1995; Kopanas et al., 2002] ont montré le rôle important joué par ces connaissances du domaine ainsi que par l'expert dans le processus de fouille de données. Ces études expliquent que le processus d'extraction de connaissances dans les données ne peut pas être totalement automatique et il est nécessaire d'étudier les mécanismes permettant les interactions entre d'une part les traitements automatiques et d'autre part la supervision de l'expert qui guide ces traitements. L'idéal est de pouvoir représenter la connaissance de l'expert en vue d'automatiser l'utilisation de celle-ci dans le processus de fouille. Cependant, en fonction des domaines, la représentation et le type des connaissances peuvent être très hétérogènes.

Deux principales représentations des connaissances ont été proposées jusqu'alors dans le cadre du clustering. La première concerne l'utilisation de contraintes entre objets et la seconde l'utilisation d'objets étiquetés. Ces deux types de connaissances sont présentés dans les sections suivantes et sont illustrés sur la figure 4.1 qui les situent par rapport aux approches supervisées et non supervisées.

4.2.1 Représentation des connaissances

4.2.1.1 Contraintes entre les objets

En clustering contraint, les connaissances sont exprimées sous la forme de relations *must-link* et *cannot-link* et sont utilisées pour guider le processus de clustering. Un lien must-link indique que deux objets devraient être dans le même cluster à la fin du processus alors qu'un lien cannot-link signifie le contraire. Ce type de connaissance est parfois plus simple à obtenir qu'un classique ensemble d'objets étiquetés. En effet, il n'est pas nécessaire de connaitre le nombre de classes, leurs étiquettes, leurs caractéristiques, etc. Seule l'information entre couple d'objets est modélisée.

Formellement, l'ensemble des contraintes est noté C avec $(C_=)$ l'ensemble des must-link et $(C_\neq)$ l'ensemble des cannot-link : $C = (C_=) \cup (C_\neq)$. Si deux objets x_i et x_j doivent être dans le même cluster, ils seront liés par un lien must-link exprimé par $c_=(i,j)$. À l'inverse, si deux objets ne doivent pas être placés dans le même cluster, ils seront liés par une contrainte cannot-link $c_\neq(i,j)$. On notera qu'il existe une inférence transitive concernant les must-link. Par exemple si $c_=(i,j)$ et $c_=(j,k)$ alors $c_=(i,k)$. Ce n'est pas le cas concernant les cannot-link comme $c_\neq(i,j)$ et $c_\neq(j,k)$ n'implique pas $c_\neq(i,k)$. Cependant on notera que si $c_=(i,j)$ et $c_\neq(i,k)$ alors $c_\neq(j,k)$.

Ces propriétés permettent d'inférer de nouvelles contraintes à partir d'un ensemble de contraintes disponibles. Il est courant de représenter l'ensemble des must-link par un graphe et d'en étudier les composantes connexes. Il faut être vigilant car plus on augmente le nombre de contraintes, plus il peut être difficile de trouver une solution ne violant aucune de celles-ci. De plus, certaines contraintes peuvent être contradictoires et peuvent alors rapidement rendre inefficace le système tentant de les utiliser.

Wagstaff et al. [2001] ont présenté une version contrainte de l'algorithme KMEANS (COP-KMEANS) qui utilise ce type de contrainte pour biaiser l'affectation des objets aux clusters. À chaque étape, l'algorithme essaie de satisfaire les contraintes spécifiées par l'expert. Ces travaux contiennent également une version contrainte de l'algorithme COBWEB (COP-COBWEB). Ces algorithmes ne renvoient un résultat que s'ils trouvent une solution qui satisfait toutes les contraintes.

Ces algorithmes qui tentent de faire respecter toutes les contraintes cherchent généralement une solution optimale. Ceci n'est pas toujours la meilleure stratégie car il n'y a aucune garantie que l'algorithme pourra converger et trouver cette solution. Davidson et Ravi [2005] ont montré que la satisfaction de toutes les contraintes must-link et cannot-link par un algorithme de type KMEANS est un problème NP-complet, et que les contraintes cannot-link peuvent empêcher l'algorithme de converger. Le problème de convergence peut être évité en étudiant le respect des contraintes au niveau local et non au niveau d'une solution globale. Davidson et Ravi [2005] ont proposé un algorithme de clustering hiérarchique contraint et ont montré que le problème de satisfaction des contraintes devient alors un problème P-complet si l'on considère un respect local des contraintes.

Ces contraintes peuvent également servir à apprendre une fonction de distance biaisée par la connaissance des liens entre les objets [Bilenko et al., 2004]. La distance entre deux objets est réduite pour un lien must-link et augmentée pour un cannot-link. Soit X les données, l'ensemble des must-link $C_=$ et l'ensemble des cannot-link $C_{\neq}$, le but est de trouver une distance métrique D entre les objets de X qui minimise :

$$\sum_{c_=(x,y)} D(x,y) \tag{4.1}$$

et maximise :

$$\sum_{c_{\neq}(x,y)} D(x,y) \tag{4.2}$$

Dans [Basu et al., 2004b], les auteurs expliquent qu'il existe deux manières d'utiliser les contraintes. La première est de modifier un algorithme de clustering pour qu'il respecte les contraintes imposées. La seconde est de modifier la fonction de distance entre les objets. Il propose d'utiliser ces deux approches conjointement via un modèle probabiliste basé sur les champs de Markov aléatoires. Un algorithme basé sur EM est utilisé pour trouver un minimum local à la fonction objective combinant respect des contraintes et modification de la métrique.

Klein et al. [2002] ont utilisé les contraintes entre paire d'objets ainsi que la position et le voisinage de ces objets pour créer de nouvelles contraintes. Les contraintes sont propagées aux objets avoisinant les objets impliqués dans des contraintes. Par exemple si les objets (x, y) sont liés par un must-link ($c_=(x, y)$) et que l'objet z est *proche* de x alors il existe sûrement un lien must-link entre z et y ($c_=(z, y)$). Cette approche permet de générer de nombreuses contraintes à partir d'un nombre limité de contraintes initiales. Cette stratégie a amélioré les résultats des précédentes versions contraintes de l'algorithme KMEANS et requiert généralement moins de contraintes initiales pour obtenir les mêmes performances. Il est cependant nécessaire de définir la notion de voisinage pour considérer qu'un objet est assez proche d'un autre pour bénéficier de ses contraintes.

Dans [Cohn et al., 2003], la divergence asymétrique de Kullback–Leibler a été utilisée pour le clustering de documents avec interaction de l'expert. Ainsi, la fonction de distance apprise (c'est-à-dire biaisée par les contraintes) reflète les différentes perspectives des experts sur les documents. Dans [Bilenko et al., 2004], les auteurs proposent un algorithme appelé MPCK-KMEANS, qui procède

à l'apprentissage d'une distance sur les objets affectés par des contraintes à chaque itération de l'algorithme de clustering. Ainsi, l'algorithme apprend différentes fonctions de distance pour chacun des clusters.

L'intégration de connaissances sous la forme de contraintes a également été étudiée dans le clustering basé sur la densité (voir Chapitre 2). Ruiz et al. [2009] ont proposé un système basé sur l'algorithme DBSCAN qui tire parti des must-link et des cannot-link. L'algorithme proposé utilise DBSCAN pour créer des clusters temporaires qui sont ensuite coupés et fusionnés en fonction des contraintes disponibles. Les expériences présentées montrent que la version contrainte de DBSCAN donne de meilleurs résultats que la version sans contrainte.

Les travaux récents en clustering contraint se concentrent sur l'évaluation de l'utilité (c'est-à-dire l'intérêt potentiel) d'un ensemble de contraintes [Davidson et al., 2006; Wagstaff, 2007]. Le but est de pouvoir évaluer la qualité d'un ensemble de contraintes, c'est-à-dire de vérifier, par exemple, si les contraintes sont cohérentes par rapport à l'espace des données, s'il y a des contraintes contradictoires, etc.

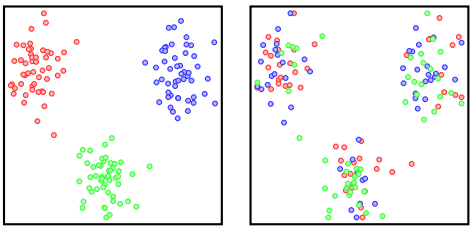
4.2.1.2 Ensemble d'objets étiquetés

Une deuxième approche utilisée pour l'intégration de connaissances en clustering fait appel à un ensemble d'objets étiquetés. L'ensemble des données disponibles $X = \{x_1, \ldots, x_n\}$ est généralement séparé en deux sous-ensembles contenant d'une part les objets dont on connait l'étiquette, et d'autre part les objets dont l'étiquette est inconnue. Formellement, on définit $\mathcal{Y} = \{y_1, \ldots y_t\}$ comme l'ensemble des étiquettes possibles (c'est-à-dire l'ensemble des labels des classes), $X_l = \{(x_i, y_i)|i = 1, \ldots, m\}$ l'ensemble des données dont l'étiquette est connue et $X_u = \{x_i|i = m+1, \ldots, l\}$ celui des objets dont l'étiquette est inconnue. Ce type d'approche partage des similarités avec les approches supervisées. En effet dans le cas supervisé, l'ensemble des objets étiquetés X_l est utilisé pour apprendre une fonction de classification qui est ensuite appliquée pour prédire la classe des objets de X_u. Ici, l'approche est sensiblement différente car l'objectif est d'étudier la structure de X_u en appliquant un algorithme de clustering, mais en utilisant les informations disponibles dans X_l pour améliorer le processus. De plus, les contraintes applicables sur l'ensemble X_l en classification supervisée sont généralement moins importantes en clustering semi-supervisé. Par exemple, il n'est pas toujours nécessaire d'avoir des exemples pour chaque classe, le nombre d'objets étiquetés peut être très faible ou encore le nombre d'objets étiquetés peut être très différent suivant les classes.

Différentes approches existent concernant l'utilisation de ce type de connaissances. Cependant, la majorité d'entre-elles font un certain nombre d'hypothèses sur ces objets étiquetés. Les deux plus importantes sont :

- les objets *proches* dans l'espace des données ont probablement la même étiquette ;
- deux objets que l'on peut connecter en passant par des régions de forte densité ont probablement la même étiquette.

Ces deux hypothèses sont assez intuitives. Elles stipulent que l'information des classes, ici utilisée comme connaissance, suit globalement la distribution des données. En effet, si la distribution des étiquettes n'a aucune corrélation avec la distribution des clusters, alors la connaissance de ces étiquettes n'apportera aucune information au clustering (voir figure 4.2). Pire, elle risque de fausser et de guider notre algorithme vers une solution non voulue. Bien sûr, il peut être intéressant dans certains cas que l'information des classes ne suive pas totalement la distribution des données. Par exemple si une classe est composée de plusieurs clusters dans l'espace des données, les deux règles précédentes sont alors violées. Cependant, cette connaissance peut être utilisée pour identifier des informations que les données seules ne pouvaient pas fournir. Comme, par exemple, que deux clusters mal séparés dans l'espace des données appartiennent à des classes différentes (voir figure 4.3). Les connaissances peuvent également servir à lever des ambiguïtés quand deux clusterings des données semblent identiques et que rien par rapport aux données ne nous permet de faire un choix. Dans ce cas, les connaissances peuvent servir à sélectionner le résultat le plus en accord avec les connaissances de l'expert.



(a) (b)

Fig. 4.2: *Deux cas où la distribution des classes suit la distribution des données (a) et où elle ne la suit pas (b). Dans le cas (b) la connaissance des classes n'est pas une information pertinente.*

Utilisation des connaissances pour l'initialisation

Dans [Basu et al., 2002], l'ensemble des objets étiquetés est utilisé pour initialiser les centroïdes dans l'algorithme KMEANS. Deux algorithmes ont été proposés, Seeded-KMEANS et Constrained-KMEANS. Dans Seeded-KMEANS, les objets étiquetés sont utilisés pour initialiser les clusters et les clusters sont mis à jour durant le processus de clustering comme dans la version classique de KMEANS. Les objets étiquetés ne sont utilisés que lors de l'étape initiale d'initialisation des centroïdes. Dans Constrained-KMEANS, les objets étiquetés sont utilisés pour l'initialisation et sont affectés à un cluster et ne peuvent plus en changer par la suite. Seuls les objets non étiquetés peuvent changer de cluster durant l'étape d'affectation aux clusters lors des itérations de l'algorithme. Le choix entre ces deux approches est généralement fait en fonction de la connaissance sur le bruit dans les données. En effet, si les données sont bruitées il peut être dangereux de ne pas autoriser de changement de clusters.

Ce type d'approche par initialisation a l'avantage de pouvoir laisser un espace de liberté à l'expert pour la découverte de clusters pour lesquels aucune connaissance n'est disponible. Par exemple, si l'expert dispose d'objets étiquetés pour deux classes, il peut utiliser ces objets pour initialiser deux centroïdes, mais peut également rajouter des centroïdes initialisés aléatoirement dans l'espoir de découvrir de nouveaux clusters. Cette approche contraint cependant l'expert à paramétrer avec précision le nombre de clusters attendus.

Dans [Wang et al., 2007], les auteurs ont introduit deux stratégies pour une initialisation de ces clusters sans utilisation de connaissance. La première stratégie intitulée Farthest-Seeded-KMEANS, consiste à initialiser les clusters sans connaissance en cherchant les objets les plus éloignés des centroïdes déjà créés. À l'initialisation de l'algorithme, un centroïde est créé pour chaque ensemble d'objets appartenant à la même classe dans l'ensemble des objets étiquetés. Puis, pour chaque centroïde que l'expert souhaite ajouter, l'algorithme va chercher l'objet des données le plus éloigné de tous les centroïdes déjà existants. Il est important de noter que cette stratégie est très sensible aux objets atypiques [2]. En effet, ceux-ci ont une grande chance d'être sélectionnés pour l'initialisation des clusters sans connaissance. La deuxième stratégie proposée est intitulée Splitting-Seeded-KMEANS. Dans cette approche, autant de clusters que de classes sont créés et les clusters sont initialisés avec les connaissances disponibles. Cette étape revient à appliquer l'algorithme Seeded-KMEANS sur les données. Puis, le cluster le plus dispersé est coupé en deux (les auteurs utilisent la distance moyenne des objets d'un cluster au centroïde pour effectuer le choix). Cette opération de scission est appliquée jusqu'à obtenir le nombre de clusters demandé par l'expert. Cette stratégie suppose que les clusters les plus dispersés sont en fait composés de plusieurs *sous-clusters* qu'il est intéressant d'identifier. Les auteurs ont montré sur quelques expérimentations que ces stratégies donnent de meilleurs résultats que l'initialisation aléatoire.

Bohm et Plant [2008] ont également proposé un algorithme de clustering hiérarchique basé sur la densité qui utilise des données étiquetées. À chaque étape, l'algorithme affecte l'étiquette la plus consistante à chaque cluster. Cette information est ensuite utilisée pour guider la construction de la hiérarchie.

[2] Objets très différents par rapport aux autres objets composant le jeu de données.

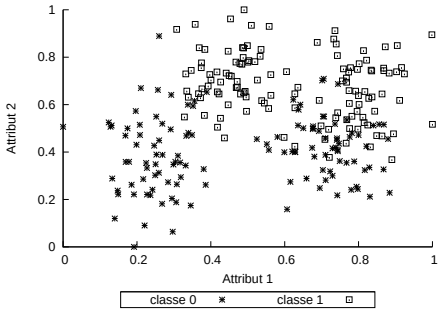


Fig. 4.3: *Exemple de jeu de données avec deux classes se recouvrant composées chacune de deux clusters.*

Pour introduire des connaissances dans un algorithme de clustering quand les données étiquetées ne partagent qu'un sous-ensemble des attributs avec les objets non étiquetés, Gao et al. [2006] ont formulé une nouvelle approche comme un problème d'optimisation. Les auteurs ont introduit deux algorithmes d'apprentissage qui sont basés sur des méthodes de clustering dures et floues. Une étude empirique montre que les algorithmes proposés améliorent la qualité des résultats de clustering malgré un nombre limité d'objets étiquetés.

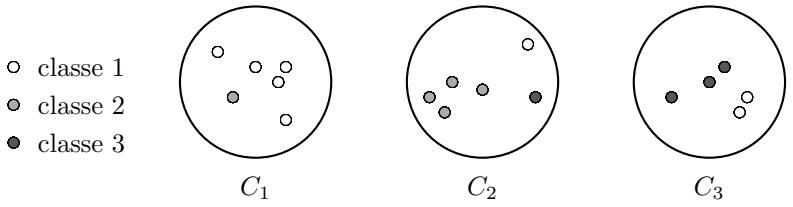
Utilisation des connaissances pour l'évaluation de la pureté des clusters

Une autre approche pour tirer parti de ces objets étiquetés est d'évaluer la pureté des clusters. Rappelons qu'une des hypothèses concernant l'utilisation d'objets étiquetés comme connaissance est que deux objets de la même étiquette sont probablement proches dans l'espace des données. Ceci peut être traduit par le fait que les objets de la même étiquette appartiennent au même cluster ou qu'un cluster ne contient que des objets ayant la même étiquette. Cette propriété peut être utilisée dans un algorithme de clustering pour guider le processus de clustering pour créer des clusters ayant une forte pureté, c'est-à-dire contenant des objets d'une unique étiquette.

Il existe de très nombreuses manières d'étudier la pureté des clusters d'un résultat de clustering, nous les présentons en détail dans la suite de ce chapitre (voir section 4.2.3). À titre d'exemple, la figure 4.4 illustre une façon simple de calculer la pureté. Ce calcul compte, pour chaque cluster, le nombre d'objets de l'étiquette la plus représentée dans le cluster. Cette stratégie considère que tous les objets du cluster doivent appartenir à l'étiquette majoritaire (c'est-à-dire la plus représentée au sein du cluster). Les objets des étiquettes minoritaires (c'est-à-dire n'étant pas de l'étiquette majoritaire) sont considérés comme des erreurs. Ce calcul est effectué pour chaque cluster et est ensuite moyenné pour donner une évaluation à l'ensemble du clustering.

Ce type d'évaluation peut être intégré dans des algorithmes de clustering comme un moyen d'utiliser des connaissances. Par exemple, Demiriz et al. [1999] ont utilisé ce type d'approche pour créer un algorithme de clustering semi-supervisé. Cet algorithme repose sur une version de l'algorithme KMEANS utilisant un algorithme génétique pour calculer les centroïdes. La fonction objective de cet algorithme est une moyenne géométrique entre la fonction objective classique de KMEANS (la dispersion des clusters) et un critère de pureté calculé à l'aide de la connaissance de l'étiquette d'un certain nombre d'objets. Les expériences menées montrent que l'algorithme utilisant les connaissances fournit de meilleurs résultats que l'algorithme utilisant uniquement les données non étiquetées.

Une approche faisant également appel à l'évaluation de la pureté des clusters est proposée par Eick et al. [2004]. Cette approche appelée *clustering supervisé*, consiste à utiliser l'étiquette des objets, afin de créer des clusters purs en terme d'étiquette. Dans ce cadre, on considère que l'on connait l'étiquette de tous les objets à classer. Le but est de former les groupes les plus homogènes possible en terme de classe. Deux principaux algorithmes sont proposés. Le premier, SRIDHCR (*Single Representative Insertion/Deletion Steepest Descent Hill Climbing with Randomized Restart*),



$$\mathbf{\Pi}_{\text{simple}} = (5 + 4 + 3)/17 \approx 0.71$$

Fig. 4.4: *Exemple de calcul de pureté simple.*

consiste à chercher un ensemble de médoides qui maximise la pureté des clusters. L'algorithme démarre en sélectionnant aléatoirement un certain nombre d'objets comme médoides initiaux. Puis, dans un processus itératif, des médoides sont retirés et d'autres sont ajoutés à l'ensemble. À chaque itération, la solution maximisant la pureté des clusters en terme d'étiquette est conservée. Comme l'algorithme est très dépendant des médoides sélectionnés initialement, celui-ci est relancé plusieurs fois avec différents ensembles initiaux. La meilleure solution parmi les différentes exécutions est conservée. Le deuxième algorithme SCEC (*Supervised Clustering using Evolutionary Computing*), utilise un algorithme génétique pour trouver le meilleur ensemble de médoides. Une solution est codée comme un ensemble de taille variable de médoides. Une population de ces solutions est générée aléatoirement au début de l'algorithme, puis évolue au cours des générations. L'avantage de l'algorithme génétique (SCEC) par rapport à la solution itérative (SRIDHCR) est que l'espace des solutions est mieux parcouru et l'algorithme a donc plus de chance de trouver une meilleure solution, bien qu'il ne soit jamais garanti d'atteindre le maximum global.

Il est également à noter que, d'une façon générale, l'ensemble d'objets étiquetés peut être utilisé pour générer un ensemble de contraintes (*must-link* et *cannot-link*). Il suffit de créer des must-link entre les objets partageant la même étiquette et des cannot-link entre les objets d'étiquettes différentes. On peut alors envisager d'utiliser tous les algorithmes vus dans la section précédente. Ainsi, les contraintes sont généralement vues comme une connaissance plus faible que des objets étiquetés.

4.2.1.3 Autres types de connaissance

Dans les deux sections précédentes nous avons vu l'utilisation de connaissance en clustering sous forme de contraintes (section 4.2.1.1) et sous forme d'objets étiquetés (section 4.2.1.2). Cependant, d'autres types de connaissance existent. Par exemple, on peut considérer que le nombre de clusters qui est souvent un paramètre de l'algorithme de clustering est une connaissance du domaine. En effet, il est nécessaire de disposer d'une expertise importante sur ses données pour pouvoir fournir le nombre de clusters adéquat. Un autre type de connaissance peut être la taille des clusters, c'est-à-dire une taille minimale ou maximale d'objets par clusters. Bradley et al. [2000] ont défini la τ_n-contrainte qui spécifie une taille n maximale pour les clusters. La ϵ-contrainte définit que deux objets à l'intérieur d'un même cluster doivent être situés plus près qu'un seuil de distance ϵ. La δ-contrainte définit quant à elle que deux objets de deux clusters différents doivent être au moins séparés d'une distance égale à δ [Davidson et Ravi, 2005]. À noter que ces approches ne précisent pas comment paramétrer ces seuils qui sont laissées au choix de l'expert.

Kumar et Kummamuru [2008] ont également introduit un autre type de connaissance dans un algorithme de clustering basé sur la comparaison relative entre les objets. Par exemple *"l'objet x est plus proche de l'objet y que de l'objet z"*. Une étude expérimentale sur des données à forte dimension a montré que l'algorithme proposé obtenait de meilleures performances et était plus robuste qu'un algorithme similaire utilisant des contraintes entre les objets (*must-link* et *cannot-link*).

Un autre type de connaissance peut être défini sous la forme d'un résultat de clustering déjà existant. En effet, il existe souvent plusieurs clusterings possibles d'un même jeu de données.

L'expert peut avoir obtenu un résultat de clustering, en être satisfait, mais vouloir continuer le processus pour identifier d'autres regroupements parmi ses données. Le premier clustering sert alors de contrainte pour éviter de retrouver cette solution. Le but est de trouver une autre solution de bonne qualité mais différente de la solution déjà trouvée lors de la première exécution. Bae et Bailey [2006] ont proposé un système appelé COALA pour trouver un clustering alternatif à partir d'un clustering existant. Le système est basé sur la recherche d'une solution de bonne qualité mais la plus dissimilaire possible d'un clustering donné. Qi et Davidson [2009] ont continué ce travail de recherche de clustering alternatif en posant le problème comme un problème d'optimisation contrainte ce qui permet son analyse en détail. L'algorithme proposé est flexible car il permet à l'expert de donner des retours positifs et négatifs sur les clusters de la solution existante. L'expert pourra indiquer quels clusters il souhaite conserver et s'il souhaite privilégier une solution très différente ou au contraire légèrement différente mais en préservant la qualité.

Une autre formalisation des connaissances a été proposée par Pedrycz [2004] qui les définit de trois façons différentes comme l'incertitude de l'appartenance aux classes, l'information basée sur la proximité de paires d'objets et enfin l'étiquetage d'une partie des données. L'auteur discute des différentes manières d'intégrer ces connaissances dans l'algorithme FUZZY-C-MEANS. Bouchachia et Pedrycz [2006] ont étendu ces travaux et présenté un algorithme de clustering flou qui prend en compte des connaissances du domaine à travers des objets étiquetés. Un des avantages de la méthode est de prendre en compte les classes séparées en plusieurs clusters. Pendant l'étape de collaboration, la méthode identifie les classes correspondantes à plusieurs clusters et ajoute ou retire des clusters à partir de cette information.

Une autre alternative pour représenter et utiliser des connaissances est de faire appel à une ontologie. Une ontologie [Gruber, 1995] est un réseau sémantique qui regroupe un ensemble de concepts tentant de décrire un domaine. Ces concepts sont liés les uns aux autres par des relations taxonomiques (hiérarchisation des concepts) d'une part, et sémantiques d'autre part. Les ontologies fournissent un cadre formel pour la représentation de connaissances. Ce type de représentation a également été étudié pour intégrer des connaissances dans les algorithmes de clustering. Adryan et Schuh [2004] ont utilisé *Gene Ontology* (GO) qui est une ontologie contenant des annotations sur les fonctions des gènes sous forme hiérarchique. La structure de GO représente les liens entre les termes biologiques associés aux gènes. Les auteurs proposent une application qui incorpore la structure de GO comme une plateforme pour sélectionner des sous-ensembles de données d'expression de gène qui sont ensuite utilisés dans un processus de clustering. Breaux et Reed [2005] ont présenté un système utilisant un algorithme de clustering hiérarchique modifié pour tenir compte de la hiérarchie d'une ontologie. L'algorithme est utilisé pour le clustering de documents car l'ontologie utilisée fournit des relations entre les mots qui peuvent être utilisées comme une distance dans l'algorithme de clustering. Une autre proposition [Hotho et al., 2001] pour le clustering de documents utilise les connaissances d'une ontologie lors d'une pré-étape au clustering ainsi que pour la sélection des résultats. Différents jeux de données sont produits en utilisant différents attributs sélectionnés grâce aux informations issues des concepts de l'ontologie. Plusieurs résultats de clustering sont ensuite calculés à l'aide de l'algorithme KMEANS. Les résultats sont ensuite distingués et expliqués grâce aux différents concepts correspondants dans l'ontologie. Une ontologie peut également servir à fournir des connaissances pour personaliser la recherche d'information dans des bases de données [Brut et al., 2008].

Les ontologies sont un moyen puissant de structurer et d'organiser des connaissances sur un domaine. Cependant, les connaissances ainsi représentées sont très spécifiques à un domaine et bien souvent à une application donnée. Les informations que l'on peut retirer des ontologies sont très hétérogènes et peuvent affecter les algorithmes différemment selon les domaines d'application. Nous verrons dans le Chapitre 4 de cette thèse nos propositions pour la structuration de connaissances dans le cadre de l'interprétation d'images de télédétection.

4.2.2 Acquisition des connaissances

L'acquisition des connaissances correspond à l'étape où l'on cherche à obtenir des informations sur le domaine ou sur le problème et à les modéliser et les stocker pour pouvoir les réutiliser par

la suite. Deux principales approches existent, les approches passives et les approches actives. Dans les approches passives les connaissances sont données a priori avant l'exécution de l'algorithme. Celles-ci ne sont pas remises en cause et sont utilisées comme telles dans l'algorithme. Par exemple un expert va étiqueter un ensemble d'objets avant que ceux-ci soient utilisés comme connaissances dans un algorithme de clustering. A contrario, les approches actives font intervenir l'expert pendant le processus de clustering, c'est-à-dire que celui-ci sera sollicité pendant l'étape de clustering pour fournir des informations.

4.2.2.1 Approches passives

Les approches passives sont les approches les plus courantes en fouille de données. Bien qu'il soit admis qu'un processus de fouille consiste en des allez-retours entre traitement automatique et analyse par l'expert [Han et al., 1999], ces interventions ont souvent lieu entre les exécutions des algorithmes. L'enjeu est ici de réussir à demander à l'expert les connaissances a priori dont il dispose sur le problème ou le domaine (par exemple le nombre de clusters, le nombre de classes, l'étiquette de certains objets, la définition de contraintes, la sélection de sources de données, etc.). Des plates-formes de visualisation de données peuvent être mises en place à ce niveau pour faciliter l'acquisition des connaissances auprès de l'utilisateur. Martin et al. [2010] ont par exemple proposé un système pour l'intégration interactive de contraintes pour la réduction de dimensions et la visualisation. Ce type d'approche permet, à l'aide d'échanges avec l'expert, de produire une masse de connaissances qui sera ensuite utilisée lors du processus de fouille.

4.2.2.2 Approches actives

Les approches actives, ou approches par apprentissage actif (*active learning*), font appel à l'expert pendant l'exécution de l'algorithme de clustering. Dans le cadre du clustering, ces approches ont essentiellement été utilisées en clustering contraint. Elles consistent à interroger l'expert sur un couple d'objets pour savoir si ceux-ci doivent être regroupés (must-link) ou au contraire doivent être séparés (cannot-link). L'intérêt de demander à l'expert cette information au cours du clustering, est de pouvoir le solliciter sur les objets pour lesquels il est difficile de se prononcer par rapport aux seules données. Typiquement, les objets à la frontière des classes seront de bons candidats à soumettre au jugement de l'expert. De cette manière, au lieu de demander à l'expert des informations sur des couples d'objets sélectionnés aléatoirement, les algorithmes se concentrent sur les objets difficiles à classer et qui en général produisent des erreurs de classification. Tout l'enjeu est alors de réussir à sélectionner les bons couples d'objets à proposer à l'expert. De plus, ce processus étant coûteux en temps, il est nécessaire de limiter au maximum le nombre de fois où l'expert est sollicité. Il est également nécessaire de réussir à tirer parti au maximum de cette nouvelle information, car si l'expert ne voit pas rapidement l'amélioration de la solution grâce à son aide, il perdra rapidement confiance dans le système. Enfin, il est nécessaire de prendre en compte la possibilité d'une erreur de la part de l'expert qui peut très rapidement fausser le résultat. Cependant, ce genre d'approche a montré de très bons résultats en clustering contraint. Plusieurs travaux [Basu et al., 2004a; Huang et Lam, 2009] ont comparé des systèmes utilisant des contraintes générées passivement puis activement et ont montré la pertinence de ces dernières. Il est cependant nécessaire de préciser que ces approches sont très coûteuses en temps et en mobilisation car il est nécessaire d'avoir un expert disponible pour chaque exécution de l'algorithme. De plus, il est difficile de définir des scénarios d'évaluation car l'ensemble des contraintes fournies par l'expert peut varier d'une exécution à l'autre ou bien d'un expert à l'autre.

Huang et Lam [2009] ont par exemple présenté un système d'apprentissage actif pour le clustering semi-supervisé de documents textuels. Cette approche utilise une méthode d'évaluation de la qualité des contraintes pour sélectionner de manière astucieuse les contraintes suivantes à soumettre à l'expert. Pour minimiser le nombre de contraintes nécessaires à l'amélioration significative des résultats, Grira et al. [2008] ont défini un mécanisme de sélection des contraintes candidates. Un système flou de sélection active est proposé et évalué sur une base de données d'images pour illustrer le fait que le clustering des objets peut être amélioré de manière significative avec peu de

contraintes. Dans le même esprit, Basu et al. [2004a] ont présenté un système de clustering contraint ainsi qu'une méthode de sélection active des contraintes dites informatives, permettant d'améliorer les performances du clustering. Des résultats empiriques et théoriques confirment que ce système de sélection active des contraintes améliore significativement les performances de l'algorithme même avec relativement peu de contraintes.

Ces systèmes sont cependant dépendants du niveau de sémantique porté par les objets à traiter. En effet, lorsqu'on présente à l'expert un couple d'objets et qu'il est sollicité pour y appliquer une contrainte du type must-link ou cannot-link, il est nécessaire que l'objet soit facilement interprétable visuellement par l'expert. Le clustering de base de données d'images est un très bon exemple, car l'expert peut souvent en un coup d'œil décider si deux images doivent être regroupées ensemble (par exemple deux images de *tigre*, ou deux images de *paysage*). Cependant, dans le cas d'objets complexes n'ayant par forcément de représentation graphique et possédant un grand nombre d'attributs, il n'est pas réaliste de demander à l'expert de se prononcer. En effet, l'expert n'a pas forcément la connaissance des raisons qui séparent réellement les différentes classes qu'il recherche. Si les objets manipulés sont des vecteurs de valeurs réelles composés de plusieurs dizaines de mesures, l'expert pourra difficilement se prononcer sur leur regroupement. Ce problème de sémantique des objets concernant les contraintes nous a poussé à explorer plus en détail l'utilisation d'objets étiquetés. En effet, cette représentation apparaît comme étant plus générique et de ce fait, plus facilement applicable à un ensemble de problèmes. Nous allons étudier dans la section suivante comment ces objets étiquetés peuvent être utilisés pour évaluer la pureté de clusters.

4.2.3 Évaluation de la pureté d'un clustering

L'évaluation de la pureté d'un clustering consiste à quantifier si le résultat du clustering est cohérent par rapport à la connaissance disponible sur les données. Nous considérons ici que la connaissance est un ensemble d'objets étiquetés. Définissons tout d'abord un ensemble de notations qui serviront à formaliser les différents critères d'évaluation de pureté présentés.

Notations :

- Soit m le nombre d'objets étiquetés ;
- Soit $\mathcal{C} = \{C_1, \ldots, C_K\}$ les clusters trouvés par l'algorithme de clustering ;
- Soit $\mathcal{Y} = \{Y_1, \ldots, Y_L\}$ les objets étiquetés pour chacune des L étiquettes ;
- Soit $n_k = |C_k|$ le nombre d'objets du cluster C_k ;
- Soit $n_j^i = |C_i \cap Y_j|$ le nombre d'objets de la classe Y_j dans le cluster C_i .

4.2.3.1 Critères d'évaluation de la pureté

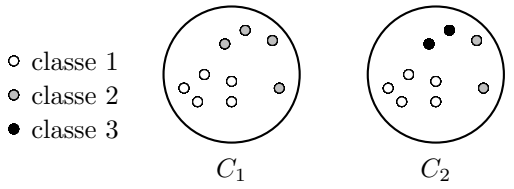
Calcul de pureté

La façon la plus simple de calculer la pureté est de chercher la classe majoritaire dans chacun des clusters et de sommer le nombre d'objets de cette classe pour chacun des clusters [Manning et al., 2008]. La pureté d'un clustering se définit alors comme :

$$\mathbf{\Pi}_{\text{simple}}(\mathcal{C}, \mathcal{Y}) = \frac{1}{m} \sum_{i}^{K} \arg\max_{C_j \in \mathcal{C}} (n_j^i) \tag{4.3}$$

Ce calcul de la pureté revient à estimer le pourcentage d'objets appartenant à la classe majoritaire de leur cluster pour l'ensemble du clustering. Sa valeur est bornée dans $[0; 1]$, 1 indiquant que les clusters sont tous purs, c'est-à-dire qu'ils ne contiennent que des objets d'une unique classe. Un exemple de calcul de pureté est donné sur la figure 4.5 ($\mathbf{\Pi}_{\text{simple}}$).

Une autre façon de calculer la pureté des clusters est proposée dans le domaine du traitement de données audio où l'objectif est d'étudier si des sons sont produits par la même source. Solomonoff et al. [1998] le formulent comme la probabilité, étant donné un cluster, que deux objets tirés au



	$\mathbf{\Pi}_{\text{simple}}$	$\mathbf{\Pi}_{\text{prob}}$
C_1	≈ 0.55	≈ 0.506
C_2	≈ 0.55	≈ 0.407
Moy	≈ 0.55	≈ 0.456

Fig. 4.5: *Exemple de calcul de pureté pour les critères* $\mathbf{\Pi}_{simple}$ *et* $\mathbf{\Pi}_{prob}$.

hasard sans remise soient de la même classe. La probabilité que le premier objet tiré du cluster C_i soit de la classe Y_j est de n_j^i/n_i. La probabilité que le second soit également de la classe Y_j est de $(n_j^i/n_i)^2$. Si ces deux objets proviennent de la même classe, alors ils proviennent soit tous les deux de la classe 1, soit tous les deux de la classe 2, etc. Ces événements étant exclusifs, l'appartenance aux classes étant dure, les probabilités peuvent être sommées pour évaluer la pureté d'un cluster.

$$\pi_{\text{prob}}(C_i) = \sum_{j=1}^{L} \left(\frac{n_j^i}{n_i}\right)^2 \tag{4.4}$$

Ce qui donne pour un clustering :

$$\mathbf{\Pi}_{\text{prob}}(\mathcal{C}, \mathcal{Y}) = \frac{1}{m} \sum_{i=1}^{K} (n_i \ \pi_{\text{prob}}(C_i)) \tag{4.5}$$

Cette mesure a l'avantage, par rapport à la pureté simple (équation (4.3)), de prendre en compte la distribution des classes minoritaires d'un cluster (c'est-à-dire les classes différentes de la classe majoritaire), et favorise donc les clusters présentant un nombre limité de classes différentes. Sa valeur est également borné dans $[0; 1]$, 1 indiquant que les clusters sont tous purs. Un exemple de la différence entre la pureté simple (équation (4.3)) et la pureté prenant en compte les classes minoritaires (équation (4.4)) est donné avec la figure 4.5. Ce critère privilégie donc également la pureté en terme de classe minoritaire, c'est-à-dire qu'à nombre d'objets de classe majoritaire identique, un cluster avec moins d'objets d'étiquette différente sera considéré comme plus pur.

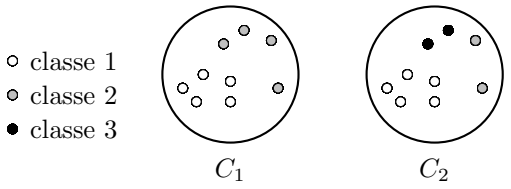
Ces deux mesures de pureté ont cependant un inconvénient majeur, qui est de surévaluer la qualité d'un clustering avec un nombre important de clusters. En effet, la pureté est maximisée dans le cas extrême où l'on observe un nombre de clusters égal au nombre d'objets. De fait, si cette mesure de pureté est utilisée dans un algorithme où le nombre de clusters peut évoluer, l'algorithme tendra à créer un nombre plus important de clusters pour s'assurer de leur pureté. Différentes propositions ont été faites pour résoudre ce problème. Par exemple, Ajmera et al. [2002] ont proposé de calculer la pureté des clusters en terme de classes et réciproquement la pureté des classes en terme de clusters, c'est-à-dire de considérer, pour chaque classe, sa répartition dans les différents clusters. Ces deux valeurs sont ensuite combinées pour donner une évaluation globale du clustering. Considérer également la pureté des classes permet de pénaliser les solutions proposant un nombre trop important de clusters. La pureté des classes se calcule de manière similaire à la pureté des clusters, mais en observant la distribution des clusters des objets dans chaque classe :

$$\pi_{\widetilde{\text{prob}}}(Y_i) = \sum_{i=1}^{K} \left(\frac{n_j^i}{|Y_i|}\right)^2 \tag{4.6}$$

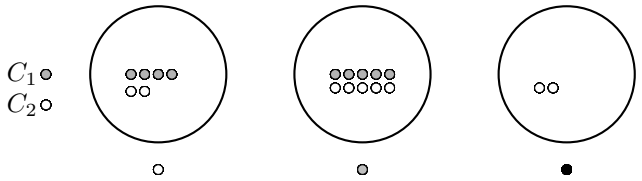
ce qui donne pour un clustering :

$$\mathbf{\Pi}_{\widetilde{\text{prob}}}(\mathcal{C}, \mathcal{Y}) = \frac{1}{m} \sum_{j=1}^{L} (|Y_j| \ \pi_{\widetilde{\text{prob}}}(Y_j)) \tag{4.7}$$

La pureté des clusters et la pureté des classes sont ensuite combinées :



(a) Calcul de la pureté des clusters.



(b) Calcul de la pureté des classes.

Fig. 4.6: *Résultat de clustering (a) permettant le calcul de la pureté des clusters, interprétation au niveaux des classes (b) permettant le calcul de la pureté des classes.*

$$\mathbf{\Pi}_{\text{overall}}(\mathcal{C},\mathcal{Y}) = \sqrt{\mathbf{\Pi}_{\text{prob}}(\mathcal{C},\mathcal{Y}) \times \mathbf{\Pi}_{\widetilde{\text{prob}}}(\mathcal{C},\mathcal{Y})} \tag{4.8}$$

Une autre approche consiste à considérer en plus la qualité du clustering à partir des données. Demiriz et al. [1999] utilisent un algorithme génétique pour optimiser la pureté des clusters en utilisant un critère similaire à celui de l'équation (4.4). Pour éviter que l'algorithme ne génère une solution avec un nombre trop important de clusters, la fonction objective est une moyenne arithmétique de la pureté des clusters et de la qualité du clustering. La qualité du clustering est évaluée grâce à l'indice de *Davies et Bouldin* [Davies et Bouldin, 1979] qui favorise les clusters compacts bien séparés dans l'espace des données. La combinaison de ces deux critères permet d'éviter des solutions trop extrêmes.

Enfin, Eick et al. [2004] ont également proposé d'introduire une notion de pénalité par rapport au nombre de clusters (équation (4.9)) pénalisant les solutions ayant un nombre de clusters trop important par rapport au nombre de classes.

$$\text{penalty}(K) = \begin{cases} \sqrt{\frac{K-L}{N}} & \text{si } K \geq L \\ 0 & \text{sinon} \end{cases} \tag{4.9}$$

avec K le nombre de clusters, L le nombre de classes et N le nombre d'objets. Cette pénalité est retranchée de l'indice de pureté choisi, pondérée par un paramètre β donné par l'expert, comme suit :

$$\mathbf{\Pi}_{\text{penalty}}(\mathcal{C},\mathcal{Y}) = \mathbf{\Pi}_{\text{simple}}(\mathcal{C},\mathcal{Y}) - \beta\ \text{penalty}(K) \tag{4.10}$$

Le paramètre β est choisi en fonction de l'importance qui est accordée à la pénalité. Une autre solution est d'utiliser le cadre de la théorie de l'information et d'évaluer l'information mutuelle normalisée [Strehl et Ghosh, 2002] entre les connaissances et le clustering :

$$\mathbf{\Pi}_{\text{nmi}}(\mathcal{C},\mathcal{Y}) = \frac{I(\mathcal{C},\mathcal{Y})}{[H(\mathcal{C}) + H(\mathcal{Y})]/2} \tag{4.11}$$

où I est l'information mutuelle définie par :

$$I(\mathcal{C},\mathcal{Y}) = \sum_i \sum_j \frac{n_j^i}{N} \log \frac{n_j^i/N}{|C_i|/N \times |Y_j|/N} \tag{4.12}$$

$$= \sum_i \sum_j \frac{n_j^i}{N} \log \frac{n_j^i}{|C_i| \times |Y_j|} \tag{4.13}$$

et H est l'entropie définie par :

$$H(\mathcal{Y}) = -\sum_k \frac{|Y_k|}{N} \log \frac{|Y_k|}{N} \tag{4.14}$$

L'information mutuelle I (équation (4.12)) mesure la quantité d'information que l'on obtient sur les classes étant donné un clustering. Le dénominateur dans l'équation (4.11) permet de normaliser le critère qui prend alors ses valeurs dans $[0;1]$, 1 indiquant des clusters purs. La valeur maximale de ce critère étant obtenue dans le cas où le nombre de clusters est égal au nombre de classes. Le critère ne possède donc pas les inconvénients des indices de pureté présentés précédemment.

Comparaison de partitions

Un autre indice couramment utilisé pour quantifier la concordance entre un clustering et une connaissance d'appartenance aux classes est l'indice de *Rand* [Rand, 1971] qui permet de comparer des partitions. Cet indice consiste à comparer des couples d'objets et vérifier s'ils sont classés de manière similaire dans deux partitions. Ici, il permet de vérifier si les couples d'objets de la même classe d'après les connaissances disponibles, ont été placés dans un même cluster. On dit qu'un couple d'objets est un vrai positif (VP) si les deux objets sont de la même classe et sont placés dans le même cluster, et un vrai négatif (VN) si les deux objets sont de classes différentes et sont placés dans deux clusters différents. Un faux positif (FP) correspond à deux objets de classes différentes placés dans le même cluster. Un faux négatif (FN) correspond à deux objets de la même classe placés dans deux clusters différents. L'indice peut être défini de la manière suivante :

$$\mathbf{\Pi}_{\text{rand}}(\mathcal{C},\mathcal{Y}) = \frac{VP + VN}{VP + FP + FN + VN} \tag{4.15}$$

$(VP + FP + FN + VN)$ représentant tous les couples possibles d'objets et $(VP + VN)$ les couples d'objets correctement classés. L'indice de *Rand* donne cependant un poids égal aux faux positifs et aux faux négatifs.

La *F-Mesure* [van Rijsbergen, 1979] quant à elle, permet de pondérer ces deux valeurs en tenant compte de la précision (P) et du rappel (R) :

$$P = \frac{VP}{VP + FP} \quad R = \frac{VP}{VP + FN}$$

$$\mathbf{\Pi}_{\text{fmesure}}(\mathcal{C},\mathcal{Y}) = \frac{(\alpha^2 + 1)P \times R}{\alpha^2 P + R} \tag{4.16}$$

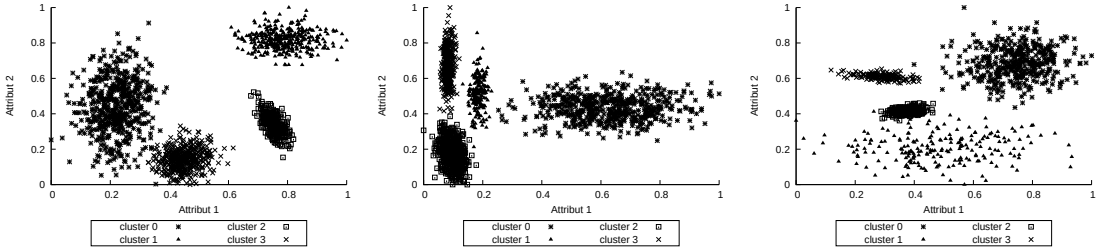
Le paramètre α peut être utilisé pour pénaliser plus fortement les faux négatifs que les faux positifs en sélectionnant une valeur $\alpha > 1$. Si $\alpha = 1$, la précision et le rappel ont alors la même importance.

L'avantage de ces deux indices ($\mathbf{\Pi}_{\text{rand}}$ et $\mathbf{\Pi}_{\text{fmesure}}$) est qu'ils intègrent implicitement le nombre de clusters, en défavorisant naturellement les solutions avec un nombre de clusters trop important. En effet, plus le nombre de clusters va augmenter, plus le regroupement des couples d'objets tendra à diverger de la connaissance disponible.

Le tableau 4.1 résume la liste des critères de pureté présentés dans cette section.

mesure	valeurs	sensibilité # clusters	sensibilité # clusters ≠ # classes	référence
$\mathbf{\Pi}_{\text{simple}}$	$[0;1]$	oui	oui	[Manning et al., 2008]
$\mathbf{\Pi}_{\text{prob}}$	$[0;1]$	oui	oui	[Solomonoff et al., 1998]
$\mathbf{\Pi}_{\text{overall}}$	$[0;1]$	non	non	[Ajmera et al., 2002]
$\mathbf{\Pi}_{\text{penalty}}$	$[0;1]$	non	non	[Eick et al., 2004]
$\mathbf{\Pi}_{\text{nmi}}$	$[0;1]$	non	non	[Strehl et Ghosh, 2002]
$\mathbf{\Pi}_{\text{rand}}$	$[0;1]$	non	non	[Rand, 1971]
$\mathbf{\Pi}_{\text{fmesure}}$	$[0;1]$	non	non	[van Rijsbergen, 1979]

Tab. 4.1: *Liste des critères de pureté et leurs propriétés.*



(a) 2d-4c-no0. (b) 2d-4c-no1. (c) 2d-4c-no2.

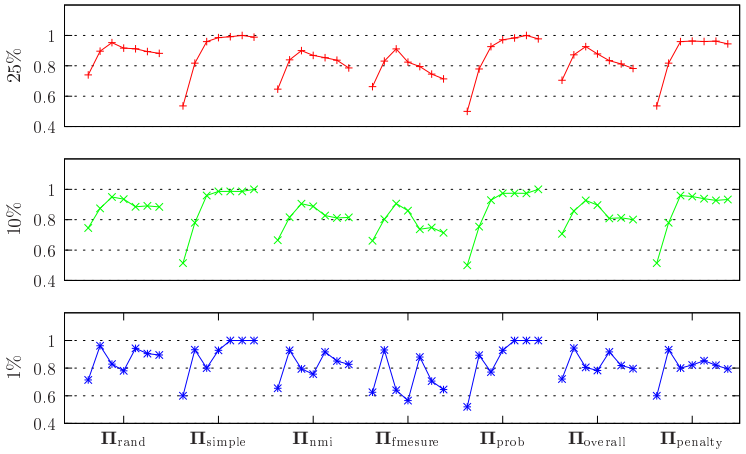
Fig. 4.7: *Les trois jeux de données utilisés pour évaluer le comportement des critères de pureté.*

4.2.3.2 Comparaison des critères

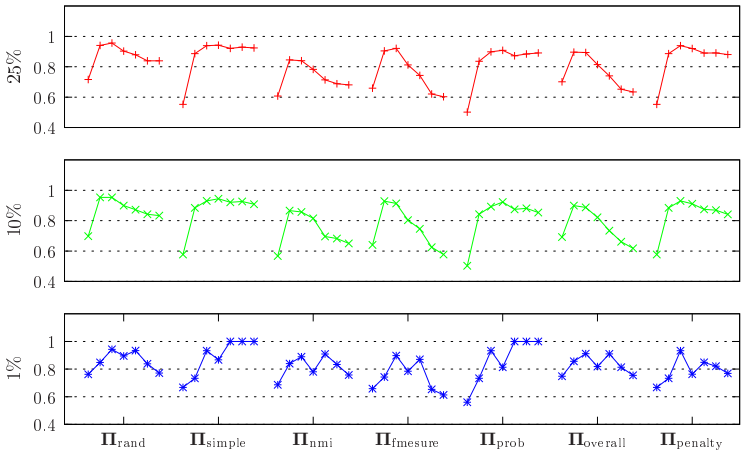
Dans cette section, nous allons présenter une étude réalisée dans le cadre de cette thèse sur les critères présentés dans la section précédente. L'objectif est d'étudier le comportement de ces critères dans des cas simples d'utilisation pour mettre en avant les limites et les avantages des différentes mesures. La figure 4.7 présente trois jeux de données [3] composés chacun de quatre clusters dans un espace à deux dimensions. L'algorithme KMEANS a été appliqué sur ces jeux de données avec un nombre de clusters variant de 2 à 8 (8 étant deux fois le nombre de clusters attendu). Pour chaque clustering, les mesures présentées dans les sections précédentes ont été calculées. Trois configurations différentes ont été évaluées, la première avec 1% des données étiquetées, la seconde avec 10% des données étiquetées et enfin la dernière avec 25% des données étiquetées. Le but est d'étudier comment les mesures réagissent en fonction du volume de connaissance disponible, c'est-à-dire en fonction du nombre d'objets étiquetés disponibles. Chaque expérience a été effectuée 100 fois avec des initialisations aléatoires de l'algorithme, puis les valeurs de chacun des indices ont été moyennées. La figure 4.8 représente les résultats pour les jeux de données des figures 4.7(a - c).

Quand le nombre d'objets étiquetés disponibles est faible (1%), la majorité des critères ont un comportement très aléatoire. En effet, il n'est pas du tout garanti d'avoir des objets pour chacune des classes du jeu de données. C'est pourquoi ces critères nous semblent difficilement utilisables quand vraiment très peu de connaissances sont disponibles. Quand le nombre d'objets étiquetés augmente (10%), il est plus probable d'avoir des objets étiquetés pour chaque classe. Par conséquent, certaines courbes deviennent moins chaotiques en trouvant leur maximum au nombre attendu de clusters. Le phénomène présenté précédemment sur le fait que les mesures de pureté surévaluent la qualité du clustering quand le nombre d'objets augmente peut être observé. En effet, la pureté simple ($\mathbf{\Pi}_{\text{simple}}$) ainsi que la pureté par cluster ($\mathbf{\Pi}_{\text{prob}}$) ne font qu'augmenter avec le nombre de clusters. Les autres indices ($\mathbf{\Pi}_{\text{rand}}, \mathbf{\Pi}_{\text{nmi}}, \mathbf{\Pi}_{\text{fmesure}}, \mathbf{\Pi}_{\text{overall}}, \mathbf{\Pi}_{\text{penalty}}$) ont tendance à diminuer avec l'augmentation du nombre de clusters, les plus caractérisques étant $\mathbf{\Pi}_{\text{fmesure}}$, $\mathbf{\Pi}_{\text{overall}}$ et $\mathbf{\Pi}_{\text{nmi}}$. Les critères $\mathbf{\Pi}_{\text{rand}}$ et $\mathbf{\Pi}_{\text{penalty}}$ diminuent de façon moins évidente. Il est intéressant de noter qu'il n'y a pas de différence importante entre les résultats obtenus avec 10% d'objets étiquetés et

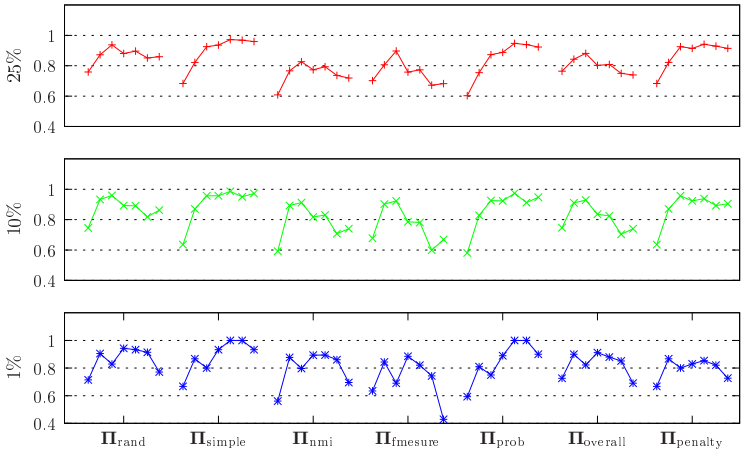
[3] `http://dbkgroup.org/handl/generators/`



(a) Évolution pour le jeu de donnée 2d-4c-no0 (figure 4.7 (a)).



(b) Évolution pour le jeu de donnée 2d-4c-no1 (figure 4.7 (b)).



(c) Évolution pour le jeu de donnée 2d-4c-no2 (figure 4.7 (c)).

Fig. 4.8: *Évolution des critères en fonction du nombre de clusters.*

25% d'objets étiquetés. Ceci peut être expliqué par la relative simplicité du jeu de données. En effet, à partir de 25% on peut estimer avoir suffisamment de connaissances pour connaître la distribution des données.

4.2.3.3 Problématique de la correspondance entre classes et clusters

La problématique de la correspondance entre classes et clusters apparaît quand le nombre de clusters présents dans les données peut être différent du nombre de classes recherché par l'expert. En effet, il est possible, et même courant, que dans certains problèmes, les classes soient en fait composées de plusieurs clusters dans les données. Il est possible d'argumenter sur ce sujet en avançant que ce type de problème arrive quand les classes du problème sont mal définies. Cependant, il est tout à fait possible que des classes soient formées d'entités plus hétérogènes au sein des données sous la forme de différents clusters. La classe peut représenter un concept haut niveau pour l'expert et avoir une granularité plus fine au niveau de l'espace de données. L'expert peut par exemple être intéressé par une classe *vin d'Alsace* qui finalement dans les données sera présente sous la forme de plusieurs clusters de *vin d'Alsace* de cépages différents (Riesling, Gewurztraminer, etc.).

Considérer la pureté individuelle des clusters permet de s'affranchir de ce problème, car il est alors possible d'identifier plusieurs clusters purs de *vin d'Alsace*. Cependant, si l'on résonne au niveau global, on va constater que tous les objets étiquetés comme *vin d'Alsace* ne sont pas forcément dans le même cluster et l'interpréter comme une erreur.

Il faut donc être vigilant quant à l'utilisation des critères de pureté. Il faut réussir à trouver un bon compromis entre une étude locale de la pureté des clusters, c'est-à-dire leur pureté individuelle, et d'autre part une étude globale, c'est-à-dire la cohérence de l'ensemble des clusters. Ne considérer que la pureté des clusters individuellement n'est pas pertinent car si le nombre de clusters est surévalué la pureté le sera également. Enfin, une évaluation globale doit être manipulée avec précaution car les classes peuvent être composées de plusieurs clusters et l'évaluation sous-estimera alors la qualité.

4.3 Intégration de connaissances en clustering collaboratif

4.3.1 Problématique

Comme nous l'avons étudié dans le Chapitre 2, le clustering collaboratif permet de faire collaborer plusieurs méthodes de clustering. La collaboration peut être définie comme un processus, où deux acteurs ou plus, travaillent ensemble pour arriver à un but commun en partageant des connaissances. La première étape du clustering collaboratif consiste à effectuer plusieurs clusterings différents des données. Puis, ces différents résultats sont modifiés pendant une étape de raffinement. Lors de cette étape, chaque résultat est remis en cause à partir des informations contenues dans les autres résultats.

Dans cette section, nous allons présenter nos propositions pour l'intégration de connaissances au sein de la méthode de clustering collaboratif. Nous nous concentrons ici sur les connaissances issues d'objets étiquetés. Nous allons voir qu'il existe différents niveaux d'intégration des connaissances dans la méthode. Chacun de ces niveaux nécessite une attention spécifique. La figure 4.9 présente les trois principaux niveaux d'intégration des connaissances.

- Le premier niveau (1) consiste à utiliser des connaissances a posteriori, c'est-à-dire à la fin du processus. En effet, nous verrons que les informations disponibles vont nous permettre d'améliorer le résultat final. Le processus mis en œuvre est un mécanisme d'étiquetage qui consiste à affecter une étiquette aux clusters et non plus aux objets. Nous verrons que ce niveau est le plus simple à mettre en œuvre car il ne nécessite pas de modifier la méthode directement, mais uniquement d'ajouter une étape en fin du processus.

- Le second niveau (2) consiste à utiliser les connaissances directement dans la méthode de clustering collaboratif afin de guider le processus de collaboration. Rappelons que le système doit

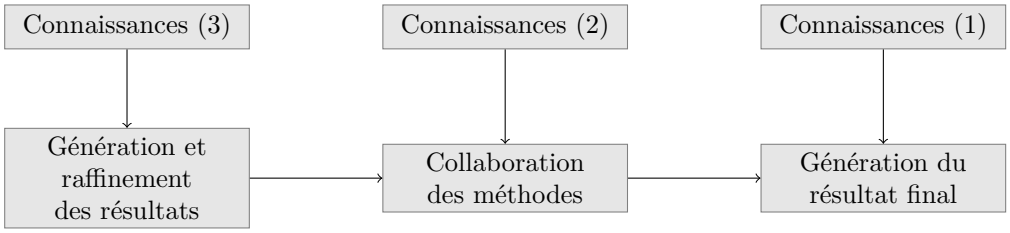


Fig. 4.9: *Les différents niveaux d'intégration des connaissances.*

prendre de nombreuses décisions quant aux modifications à apporter aux résultats pour les rendre plus similaires et de bonne qualité. Les connaissances sont utilisées à ce niveau pour évaluer la qualité des résultats et guider le choix des modifications à effectuer.

- Le dernier niveau (3) consiste à modifier les méthodes qui vont prendre part à la collaboration pour qu'elles tiennent compte des connaissances directement au niveau de leur clustering des données. Ce niveau est le plus coûteux à mettre en œuvre car il nécessite de modifier en profondeur chacune des méthodes que l'on souhaite voir collaborer. En effet, il est nécessaire d'adapter les opérateurs spécifiques à chaque méthode (scission, fusion, reclustering) pour qu'ils utilisent les connaissances disponibles.

4.3.2 Intégration des connaissances a posteriori

L'intégration de la connaissance a posteriori consiste à utiliser la connaissance des objets étiquetés pour étiqueter les clusters d'un résultat de clustering et à améliorer le résultat produit. La plupart des algorithmes de clustering semi-supervisé opèrent sous deux contraintes fortes :

1. À chaque étiquette (ou classe) ne correspond qu'un seul cluster ;
2. On dispose d'au moins un exemple de chaque classe.

Or, ces deux hypothèses sont souvent mises à défaut dans les problèmes réels. En effet, comme expliqué dans la section 4.2.3.3, il est souvent possible de rencontrer des classes correspondant en fait à plusieurs clusters. De plus, le choix des objets étiquetés est souvent effectué de manière aléatoire ce qui explique pourquoi il n'est pas garanti d'avoir des exemples pour chacune des classes réelles. Nous nous proposons ici de relâcher ces deux contraintes. Pour cela, nous utilisons les étiquettes des objets pour affecter une étiquette à chacun des clusters quand cela est possible. Pour chaque cluster, la distribution des étiquettes présentes dans celui-ci est étudiée. Un vote à la majorité est ensuite effectué pour connaître l'étiquette la plus probable pour le cluster. Une approche sensiblement similaire a déjà été explorée par Jia et Richards [2005] pour réduire la taille d'un jeu d'apprentissage en classification supervisée. Nous allons nous en inspirer ici pour proposer un mécanisme d'étiquetage des clusters.

La première étape consiste à appliquer un clustering collaboratif sur les données. À l'issue du processus collaboratif, nous allons travailler sur le résultat unifié, c'est-à-dire le résultat final issu de la fusion des différents résultats impliqués dans la collaboration (voir section 3.3.2.3). Il aurait cependant été possible d'appliquer la même approche à chaque résultat raffiné.

Soit $\mathcal{C}^{(*)} = \{C_1^{(*)}, \ldots, C_k^{(*)}\}$ le résultat unifié obtenu à la fin du processus collaboratif. L'objectif est d'utiliser les connaissances pour affecter une étiquette de $\mathcal{Y} = \{Y_1, \ldots, Y_L\}$ à chacun des clusters de $\mathcal{C}^{(*)}$. Un vote est effectué au sein de chaque cluster pour identifier l'étiquette la plus présence. Pour cela, l'intersection entre les objets appartenant au cluster et les objets étiquetés est calculée. L'intersection maximale est conservée pour affecter une étiquette au cluster. La règle de vote suivante est appliquée :

$$\texttt{classe-majoritaire}(C_k^{(*)}) = \arg \max_{\mathcal{C}_i^{(*)} \in \mathcal{C}^{(*)}} |\mathcal{C}_i^{(*)} \cap Y_l| \tag{4.17}$$

	Étiquetage de clusters	S-KMEANS	C-KMEANS
Jeu de données 1	96.75(± 1.43)	97.75(±0.00)	**97.96**(±**0.22**)
Jeu de données 2	**98.36**(± **0.48**)	61.81(±7.21)	62.75(±7.50)
Jeu de données 3	**91.18**(±**10.81**)	55.56(±0.69)	55.57(±0.75)
Jeu de données 4	**97.98**(±**3.87**)	70.3(±5.09)	67.12(±3.64)

Tab. 4.2: *Résultats sur les quatres jeux de données.*

Si aucun objet étiqueté n'est disponible dans un cluster, c'est-à-dire $\forall k, \mathcal{C}_i^{(*)} \cap Y_k = \emptyset$, une nouvelle étiquette est donnée au cluster. Les deux principaux avantages de cette approche sont qu'elle permet :

1. Qu'une classe ne corresponde pas nécessairement qu'à un seul cluster car plusieurs clusters peuvent obtenir la même étiquette à l'issue de l'étape de vote ;
2. Que certains clusters ne soient pas étiquetés.

Le premier avantage permet de découvrir des classes composées de plusieurs clusters dans l'espace des données. Ceci est de plus en plus fréquent avec l'augmentation de la masse de données et de la granularité des classes. Le second avantage permet quant à lui de laisser un degré de liberté à la méthode pour découvrir de nouvelles classes. Une étape manuelle est nécessaire où les clusters non étiquetés sont présentés à l'expert.

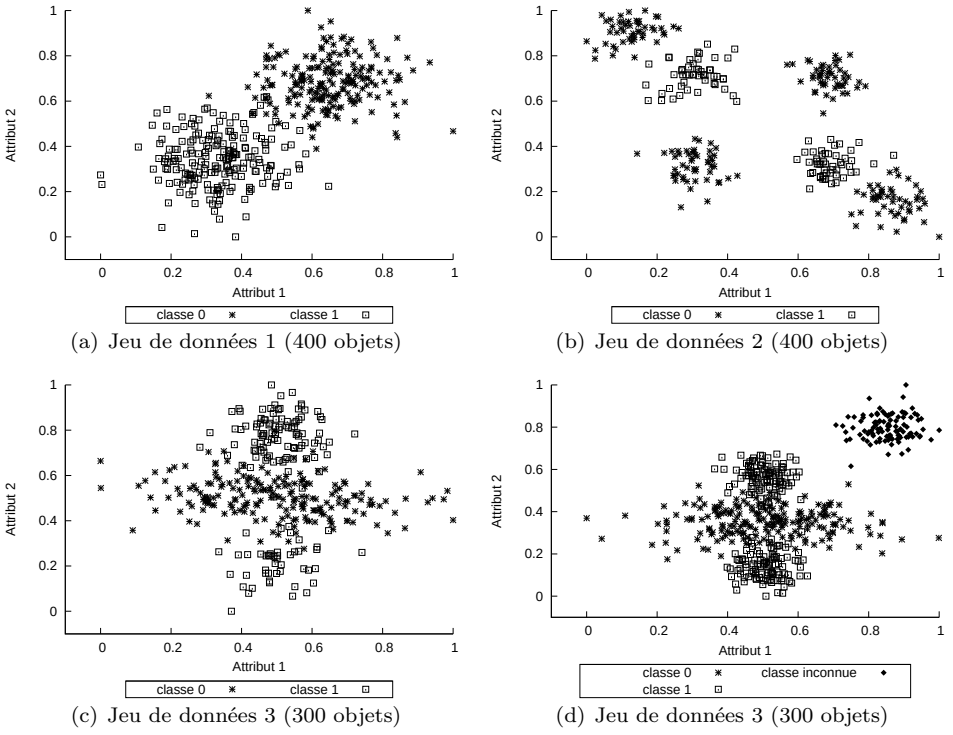
Pour illustrer le fonctionnement de cette approche nous avons effectué des expériences sur des jeux de données artificiels. Les figures 4.10 (a - d) présentent les quatre jeux de données utilisés dans ces expériences. Le jeu de données 1 (4.10 (a)) est un jeu de données simple composé de deux classes représentées sous la forme de deux clusters. Le jeu de données 2 (4.10 (b)) est plus complexe car une des classes est composée de deux clusters. Le jeu de données 3 (4.10 (c)) est encore plus complexe car une classe est décomposée en deux clusters, et une autre en quatre clusters. Enfin, le jeu de données 4 (4.10 (d)) est similaire au jeu de données (4.10 (b)) mais présente une troisième classe sous la forme d'un unique cluster. Cependant, on considère que ce cluster est inconnu de l'expert. L'enjeu sera par conséquent d'utiliser les éventuelles connaissances disponibles sur les deux classes pour les identifier, mais également pour découvrir ce dernier cluster. Pour donner un point de comparaison nous avons également utilisé les algorithmes semi-supervisés Seeded-KMEANS (S-KMEANS) et Constrained-KMEANS (C-KMEANS). Ces algorithmes utilisent les connaissances pour initialiser les clusters avant l'application de KMEANS.

Dans toutes ces expériences, 5% des objets des jeux de données ont été utilisés comme connaissance. Les algorithmes ont été appliqués 100 fois et les résultats ont été moyennés. Le tableau 4.2 présente les résultats sous la forme de la moyenne et de l'écart-type de la précision de classification calculée sur les résultats pour les quatre jeux de données. Comme on peut le constater dans ce tableau, quand le jeu de données est simple (jeu de données 1) les résultats sont sensiblement équivalents. Cependant, quand le jeu de données devient plus complexe (jeux de données 2, 3 et 4) les résultats sont clairement à l'avantage de notre approche qui permet de prendre en compte les classes composées de plusieurs clusters.

La figure 4.11 montre l'évolution de la précision de classification en fonction du pourcentage d'objets étiquetés disponibles sur le jeu de données 2. On peut noter sur cette courbe que la méthode proposée fait preuve d'une grande stabilité, la précision étant sensiblement la même entre 5% à 50% de données étiquetées disponibles.

4.3.3 Guider le processus par les connaissances

Dans cette section nous allons voir comment utiliser les connaissances directement dans la méthode collaborative. Nous rappelons ici que la méthode de clustering collaboratif SAMARAH présentée dans le Chapitre 2 consiste à utiliser plusieurs méthodes de clustering puis à comparer et modifier les résultats obtenus. Dans cette méthode, un certain nombre de critères sont utilisés pour comparer les résultats et évaluer leur qualité.



(a) Jeu de données 1 (400 objets)
(b) Jeu de données 2 (400 objets)
(c) Jeu de données 3 (300 objets)
(d) Jeu de données 3 (300 objets)

Fig. 4.10: *Les quatre jeux de données utilisés dans les expériences.*

Ces critères de qualité sont notamment utilisés lors de la phase de résolution de conflits entre deux résultats. Le coefficient local d'évaluation (équation (3.13), page 53) consiste à comparer la similarité et la qualité d'un couple de résultats de clustering. Nous proposons de modifier ce coefficient afin que l'évaluation de la qualité prenne en compte les connaissances.

Dans la version initiale de SAMARAH, le coefficient local d'évaluation $\gamma^{(i,j)}$ évalue la similarité et la qualité de deux résultats de clustering $\mathcal{C}^{(i)}$ et $\mathcal{C}^{(j)}$. Ce coefficient inclut un critère de qualité $\delta^{(i)}$ qui évalue la qualité du résultat $\mathcal{C}^{(i)}$. Dans sa version actuelle, celui-ci ne prend en compte que la qualité interne des clusterings à travers un unique critère. Nous proposons de remplacer ce coefficient par un coefficient générique $Q(\mathcal{C}^{(i)})$ permettant d'intégrer différents types de connaissances dans la méthode. Celui-ci peut prendre en compte deux aspects du résultat, sa qualité interne et sa qualité externe. L'évaluation interne consiste à évaluer la qualité du clustering à travers une mesure de qualité non-supervisée (voir section 2.3). L'évaluation externe consiste à évaluer la qualité du résultat par rapport à des informations externes fournies par l'expert (par exemple le nombre de clusters, des objets étiquetés, des contraintes, etc.). Nous allons voir dans la suite comme le coefficient $Q(.)$ a été défini pour prendre en compte les différentes connaissances.

4.3.3.1 Contraintes sur la qualité

Pour pouvoir prendre en compte différents types de connaissances du domaine, et pour pouvoir les combiner, nous avons défini un nouveau critère d'évaluation de la qualité qui prend en compte les connaissances sous forme de contraintes :

$$Q(\mathcal{C}^{(i)}) = \sum_{c=1}^{N_c} q_c(\mathcal{C}^{(i)}) \times p_c \tag{4.18}$$

où N_c est le nombre de contraintes à respecter correspondant aux connaissances du domaine disponibles, q_c est le critère utilisé pour évaluer le résultat en fonction de la c-ième contrainte

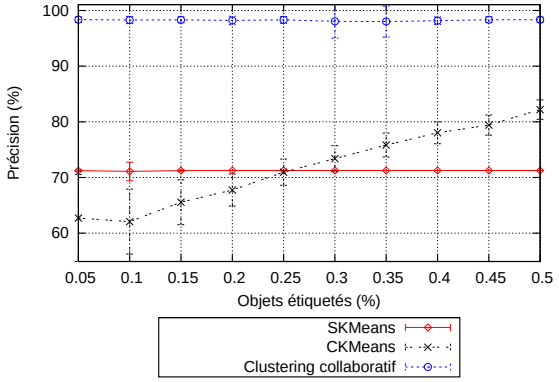


Fig. 4.11: *L'évolution de la précision de classification pour le jeu de données 2 (figure 4.10 (b)).*

($q_c(.) \in [0,1]$) et p_c est l'importance relative donnée par l'expert à la c-ième contrainte ($p_1 + p_2 + \cdots + p_{N_c} = 1$)). Par défaut, chaque contrainte a un poids égal à $1/N_c$.

Ainsi, il est possible d'intégrer un type de connaissances en clustering collaboratif s'il est possible de l'exprimer sous forme de contraintes et de définir une fonction prenant ses valeurs dans $[0;1]$ pour l'évaluation du respect de celles-ci. Nous allons décrire dans la prochaine section, des exemples de types de connaissances qui peuvent être intégrés de cette manière.

Contraintes basées sur la qualité interne du clustering

L'utilisation de contraintes basées sur la qualité interne est la manière la plus simple d'évaluer la qualité d'un clustering. En effet, peu de connaissances sont nécessaires, car l'évaluation de la qualité du regroupement est basée sur les données elles-même (voir section 2.3). Le choix de ce critère est en général dépendant de la méthode de clustering utilisée. En effet, le critère est souvent lié à la fonction objective de la méthode de regroupement utilisée, cette fonction objective étant souvent de fait à même de juger de la qualité du clustering et de son éventuelle amélioration.

Une autre connaissance utilisée généralement de pair avec l'utilisation de critères internes est l'utilisation d'une estimation du nombre de clusters fournie par l'expert. En effet, l'expert est en général capable d'exprimer ses besoins sous une forme plus ou moins précise : *"j'attends entre 5 et 10 clusters"*. Pour utiliser cette information il est possible de pondérer le critère de qualité interne par une évaluation de ce nombre de clusters. Avec n_{sup} la borne supérieure et n_{inf} la borne inférieure ($n_{\text{inf}} \leq n_{\text{sup}}$), on obtient :

$$p(C^{(i)}) = \frac{n_{\text{sup}} - n_{\text{inf}}}{|K^{(i)} - n_{\text{inf}}| + |n_{\text{sup}} - K^{(i)}|} \tag{4.19}$$

où $[n_{\text{inf}}, n_{\text{sup}}]$ est donné par l'expert. La figure 4.12 présente une courbe représentant les valeurs de l'évaluation de ce critère pour un intervalle compris entre 5 et 10 en fonction du nombre de clusters allant de 1 à 14. Comme on peut l'observer sur cette courbe, quand la valeur trouvée se situe dans l'intervalle proposé par l'expert, la valeur est 1 et plus la valeur s'éloigne de l'intervalle plus la qualité diminue. Cette valeur peut être utilisée comme une pondération dans un critère de qualité interne q_c :

$$q'_c(\mathcal{C}^{(i)}) = q_c(\mathcal{C}^{(i)}) \times p(\mathcal{C}^{(i)}) \tag{4.20}$$

Cette approche permet de prendre en compte la qualité du résultat interne, mais celle-ci est pondérée par une information fournie par l'expert sur le nombre de clusters attendu.

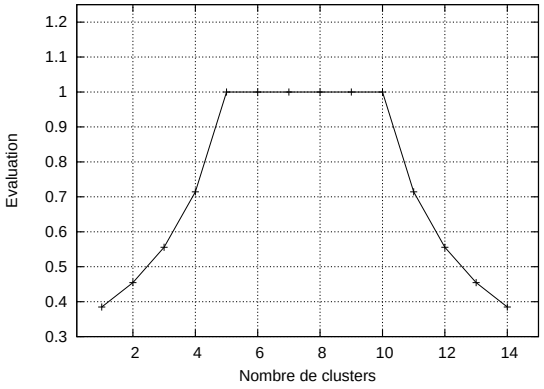


Fig. 4.12: *Évolution de l'évaluation du nombre de clusters.*

Contraintes basées sur des liens entre objets

Si des contraintes entres les objets sont disponibles (voir section 4.2.1.1), il est possible d'évaluer l'accord des résultats avec ces contraintes et de favoriser les résultats respectant au mieux les contraintes sur les objets.

$$q_{link}(\mathcal{C}^{(i)}) = \frac{1}{n_r} \sum_{j=1}^{n_r} v(\mathcal{C}^{(i)}, c_j) \tag{4.21}$$

où n_r est le nombre de contraintes entre les objets, c_j est un lien must-link ou cannot-link et $v(\mathcal{C}^{(i)}, c_j) = 1$ si $\mathcal{C}^{(i)}$ respecte la contrainte c_j, 0 sinon.

Contraintes basées sur la qualité externe du clustering

Ce type de contrainte consiste à utiliser des connaissances externes fournies par l'expert pour évaluer le résultat. Nous considérons ici l'utilisation des objets étiquetés introduite dans la section 4.2.1.2. Ces objets étiquetés peuvent être utilisés pour évaluer l'accord entre ces connaissances et les résultats à évaluer. Les mesures de pureté peuvent notamment être utilisées pour évaluer la pureté des clusters en fonction des objets étiquetés. Utiliser les connaissances à ce niveau du processus collaboratif nous permet de nous assurer que la méthode collaborative va avantager les résultats en accord avec les connaissances fournies par l'expert. En effet, les critères d'évaluation de la qualité dans la méthode collaborative servent à guider le processus et à faire des choix sur les modifications pertinentes à appliquer. La méthode sera par conséquent influencée pour que les décisions prises soient en accord avec les connaissances.

4.3.3.2 Prise en compte de la pureté en clustering collaboratif

Dans cette section, nous proposons une évaluation de la pertinence des différents critères de pureté en clustering collaboratif afin d'étudier leur utilité dans le processus de collaboration. Pour évaluer la qualité de ces critères de pureté, nous avons développé deux indices d'évaluation permettant de juger de leur pertinence pour évaluer un couple de résultats. Ces deux indices font appel à une fonction qui évalue la qualité d'un couple de résultats selon un critère de pureté donné en calculant la moyenne des deux évaluations des deux résultats. Cette valeur moyenne est utilisée en clustering collaboratif pour évaluer la qualité d'un couple de résultats. Soit $\mathcal{C}^{(K)}$ un résultat avec K clusters, $\mathcal{C}^{(M)}$ un résultat avec M clusters et $\mathbf{\Pi}$ une fonction de pureté, la moyenne des deux évaluations est définie telle que :

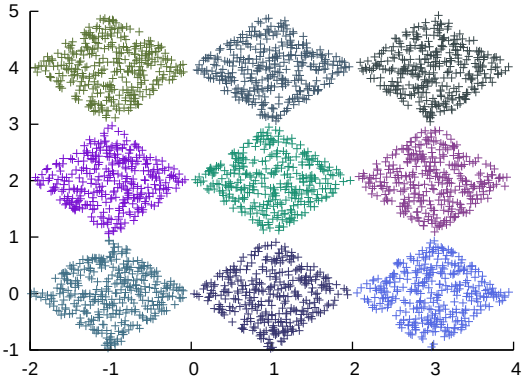


Fig. 4.13: *Jeu de données 9-Diamonds.*

$$\mathfrak{f}(\mathcal{C}^{(K)}, \mathcal{C}^{(M)}) = \frac{1}{2}(\mathbf{\Pi}(\mathcal{C}^{(K)}, \mathcal{Y}) + \mathbf{\Pi}(\mathcal{C}^{(M)}, \mathcal{Y})) \tag{4.22}$$

Le premier indice défini à partir de cette fonction évalue la différence entre la valeur maximale de la fonction et la valeur de la fonction quand les méthodes fournissent des résultats ayant tous les deux le nombre réel de clusters présents dans le jeu de données étudié :

$$\mathfrak{c}_{max}(\mathcal{C}^{(K)}, \mathcal{C}^{(M)}) = \arg\max_{K,M}(\mathfrak{f}(\mathcal{C}^{(K)}, \mathcal{C}^{(M)})) - \mathfrak{f}(\mathcal{C}^{(L)}, \mathcal{C}^{(L)}) \tag{4.23}$$

avec $\mathcal{C}^{(L)}$ un résultat à L clusters, L étant le nombre de clusters réel. L'évaluation de ce critère est minimisée si la valeur de $\mathfrak{f}$ est maximale quand les deux résultats ont un nombre de clusters identique à celui du jeu de données. Une valeur faible de $\mathfrak{c}_{max}$ indique que le critère de pureté permet d'identifier correctement le bon nombre de clusters. Plus la valeur retournée par la fonction est élevée, plus le couple de résultats est considéré comme pur, c'est-à-dire de bonne qualité.

Le deuxième indice évalue l'ensemble de la fonction, en pénalisant des valeurs importantes quand le nombre de clusters proposé par les méthodes s'éloigne du nombre réel de clusters. Plus le critère est faible, plus la fonction aura le comportement attendu.

$$\mathfrak{c}_{sup}(\mathcal{C}^{(K)}, \mathcal{C}^{(M)}) = \frac{\sum_{K,M}^{L}((|K-L|+|M-L|) \times \mathfrak{f}(\mathcal{C}^{(K)}, \mathcal{C}^{(M)}))}{\sum_{K,M}^{L}(|K-L|+|M-L|)} \tag{4.24}$$

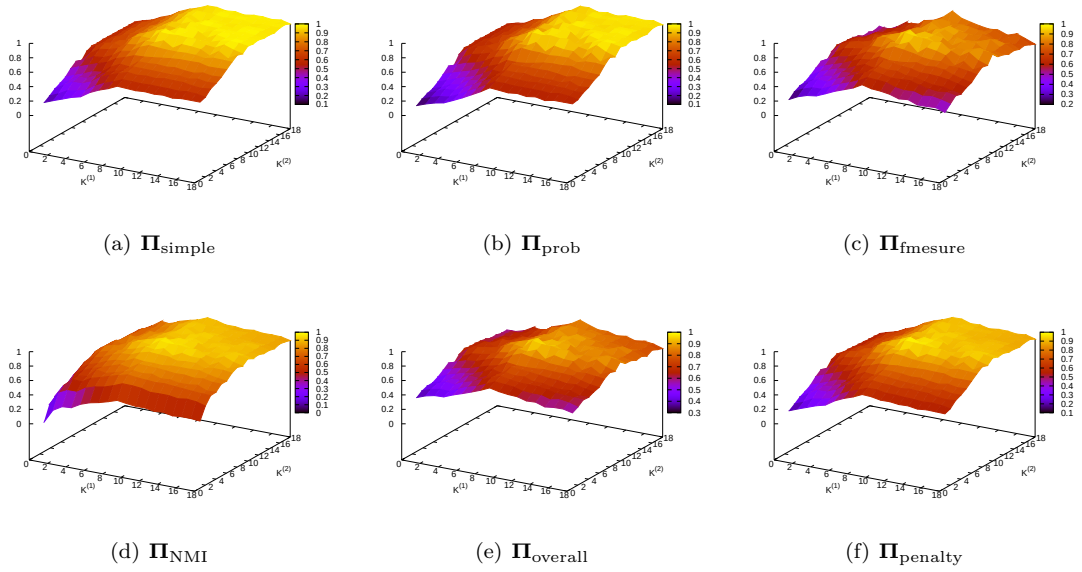
Pour illustrer le comportement de ces indices et évaluer les différents critères de pureté, nous avons effectué des expériences dans lesquelles deux méthodes ont collaboré. L'algorithme KMEANS a été utilisé sur le jeu de données 9-Diamonds (voir figure 4.13) présentant neuf clusters en forme de losanges [Salvador et Chan, 2004]. Ces données proviennent de l'équipe Apprentissage de l'Université de Houston [4]. Malgré l'apparente simplicité du jeu de données, il arrive couramment que l'algorithme KMEANS ne trouve pas les 9 clusters même avec K égal à 9. En effet, en fonction de l'initialisation, l'algorithme peut converger rapidement vers un optimum local et ne pas trouver la structure telle qu'elle est visible.

Des clusterings ont été calculés avec un nombre de clusters allant de 2 à 18 (18 étant deux fois le nombre de clusters attendu), et une initialisation aléatoire des centroïdes. Pour chacun de ces clusterings, les valeurs des critères de pureté présentés dans la section 4.2.3.1 ont été calculées avec 10 % des données étiquetées. Pour chaque couple de résultats, la fonction $\mathfrak{f}$ a été calculée. La figure 4.14 représente cette fonction $\mathfrak{f}(\mathcal{C}^{(K)}, \mathcal{C}^{(M)})\ \forall\ K, M \in [2; 18]$ pour chacun des critères de pureté étudiés. Dans notre exemple, le nombre de clusters du jeu de données étant 9, il est attendu que cette fonction soit maximale lorsque les deux résultats comparés ont 9 clusters. Les deux indices

[4] `http://www.tlc2.uh.edu/dmmlg/`

	$\mathbf{\Pi}_{\text{simple}}$	$\mathbf{\Pi}_{\text{prob}}$	$\mathbf{\Pi}_{\text{overall}}$	$\mathbf{\Pi}_{\text{penalty}}$	$\mathbf{\Pi}_{\text{NMI}}$	$\mathbf{\Pi}_{\text{rand}}$	$\mathbf{\Pi}_{\text{fmesure}}$
$\mathfrak{c}_{max}$	0.003	0.012	0.0	0.003	0.0	0.0	0.0
$\mathfrak{c}_{sup}$	0.744	0.703	0.709	0.708	0.730	0.839	0.661

Tab. 4.3: *Résultats des évaluations de l'utilisation des critères de pureté en clustering collaboratif.*



(a) $\mathbf{\Pi}_{\text{simple}}$ (b) $\mathbf{\Pi}_{\text{prob}}$ (c) $\mathbf{\Pi}_{\text{fmesure}}$

(d) $\mathbf{\Pi}_{\text{NMI}}$ (e) $\mathbf{\Pi}_{\text{overall}}$ (f) $\mathbf{\Pi}_{\text{penalty}}$

Fig. 4.14: *Étude du comportement des différents critères de pureté en clustering collaboratif.*

$\mathfrak{c}_{max}$ et $\mathfrak{c}_{sup}$ ont été calculés à partir des résultats illustrés sur la figure 4.14 et sont présentés dans le tableau 4.3.

Seul trois des sept indices évalués ne possèdent pas leur maximum à l'endroit attendu. Parmi ces trois indices, sont présents les deux indices de pureté ($\mathbf{\Pi}_{\text{simple}}$ et $\mathbf{\Pi}_{\text{prob}}$) dont la tendance à surévaluer un résultat avec un nombre trop important de clusters a déjà été soulevée précédemment. L'évaluation de l'ensemble des résultats à l'aide du deuxième critère permet de prendre en compte le comportement global des critères. Sur ce jeu de données, l'indice $\mathbf{\Pi}_{\text{fmesure}}$ permet d'obtenir le meilleur résultat (0.661). L'indice $\mathbf{\Pi}_{\text{rand}}$ est le plus mauvais avec un score de 0.839.

4.3.3.3 Impact de l'intégration des connaissances dans la méthode collaborative

Pour étudier l'impact de l'intégration des différents types de connaissances sur la méthode collaborative, nous avons calculé le coefficient local d'évaluation (équation (3.13)) entre les méthodes en fonction de ceux-ci. Les figures 4.15 (a - h) représentent les valeurs de l'évaluation du coefficient local d'évaluation entre deux méthodes en faisant varier le nombre de clusters des deux résultats. L'algorithme KMEANS a été utilisé pour créer les clusters du jeu de données avec des nombres de clusters allant de 2 à 18. Ces différentes figures illustrent comment les différents types de connaissances peuvent influer sur la recherche de solution et la production de résultats.

La figure 4.15 (a) présente les valeurs du coefficient local d'évaluation en utilisant uniquement la similarité des résultats (équation (3.13)). Il apparaît sur cette figure plusieurs optima locaux dans l'espace des solutions car les résultats avec très peu de clusters ont tendance à être fortement similaires. Pour résoudre ce problème, l'utilisation d'une connaissance sur une estimation d'un intervalle du nombre de clusters attendu pour le résultat de clustering peut être envisagée.

La figure 4.15 (c) présente les valeurs du coefficient local d'évaluation en utilisant la similarité

ainsi qu'une connaissance sur l'estimation du nombre de clusters, ici l'intervalle $[7; 11]$. On peut voir sur cette figure que la méthode tire parti de cette information puisqu'on constate une réduction de la valeur de l'évaluation des solutions dont le nombre de clusters est situé à l'extérieur de cet intervalle.

De plus, si des objets étiquetés sont disponibles, une mesure de pureté peut également être intégrée pour guider la collaboration. Ceci est illustré avec la figure 4.15 (d) où un indice de pureté a été calculé sur 5% des objets étiquetés. On constate que cette connaissance améliore considérablement la forme de l'espace des solutions.

Enfin, les différentes connaissances peuvent être utilisées ensemble comme dans la figure 4.15 (e) où l'estimation du nombre de clusters ainsi que la connaissance des objets étiquetés sont utilisées. L'espace des solutions résultant de cette combinaison est clairement plus simple à explorer et ne contient plus d'optima locaux. La solution optimale (9 clusters) est fortement mise en avant.

Cet exemple relativement simple permet d'illustrer comment l'intégration de connaissances au sein de la méthode de clustering collaboratif permet de guider le processus et de simplifier la recherche de solutions. Les différents types de connaissances apportent des informations pertinentes qui permettent de faire des choix en accord avec les attentes de l'expert. Dans la section suivante, nous allons évaluer numériquement l'intérêt d'utiliser ces connaissances dans le processus sur des jeux de données réels issus de la base UCI [Asuncion et Newman, 2007].

4.3.3.4 Évaluation de l'intégration de connaissances dans la méthode

Dans cette section, nous présentons une série d'expériences afin de montrer la pertinence de l'intégration de connaissances en clustering collaboratif. Les expériences ont deux buts. Le premier est de montrer l'intérêt du clustering collaboratif pour améliorer la qualité d'un ensemble de résultats de clustering. La seconde est de montrer l'intérêt d'ajouter des connaissances dans le processus collaboratif pour produire des résultats encore meilleurs.

Deux séries d'expériences ont été menées :

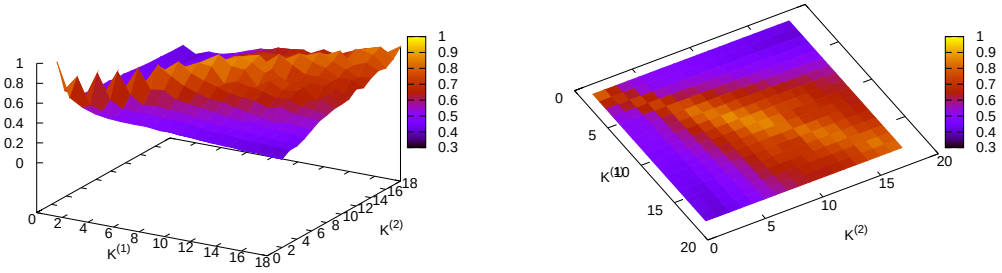
1. La première porte sur l'évaluation de la qualité d'un ensemble de résultats de clustering, premièrement sans aucune collaboration, puis avec collaboration, et enfin avec une collaboration avec intégration de connaissances. Plusieurs indices de qualité ont été utilisés pour évaluer la qualité de ces ensembles de résultats de clustering.
2. La deuxième expérience a consisté en une évaluation en cascade [Candillier et al., 2006]. Cette approche est basée sur l'enrichissement d'un jeu de données par des résultats de clustering, puis par l'utilisation d'une méthode de classification supervisée pour évaluer l'intérêt d'ajouter une telle information au jeu de données.

Pour toutes les expériences avec la méthode collaborative nous avons utilisé l'algorithme KMEANS comme méthode de base. Cinq occurrences de cette méthode ont été initialisées de manière aléatoire avec un nombre de clusters choisi aléatoirement dans l'intervalle $[2; 10]$. L'étape de raffinement des résultats a été paramétrée pour trouver un résultat avec un nombre de clusters dans $[2; 10]$ (c'est-à-dire $n_{\text{inf}} = 2, n_{\text{sup}} = 10$ dans l'équation (4.19)).

Pour évaluer le bénéfice de l'intégration de connaissances dans l'étape de raffinement, nous avons sélectionné aléatoirement 10% des objets des jeux de données comme connaissances disponibles. Cet ensemble d'objets étiquetés a été utilisé pour guider la collaboration à travers l'évaluation de la pureté des résultats. Cinq jeux de données de l'UCI [Asuncion et Newman, 2007] ont été utilisés pour les expériences.

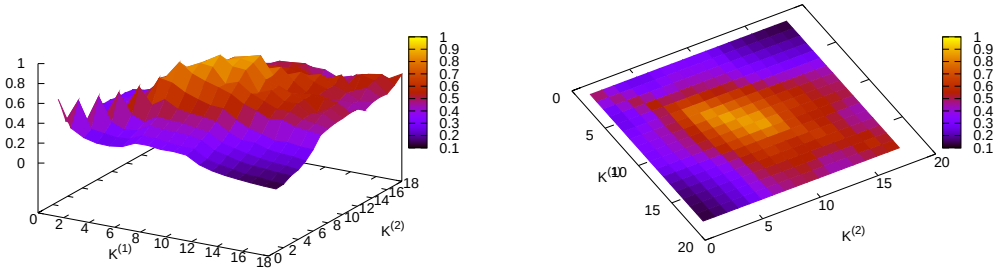
Comparaison par des indices de qualité

Dans cette première expérience, nous avons évalué la qualité de l'ensemble de résultats de clustering sans collaboration (configuration appelée par la suite $\neg col$), avec collaboration (configuration appelée par la suite col), et enfin avec collaboration utilisant des connaissances (configuration appelée par la suite $kcol$). Chaque ensemble obtenu correspond aux différents résultats fournis par les



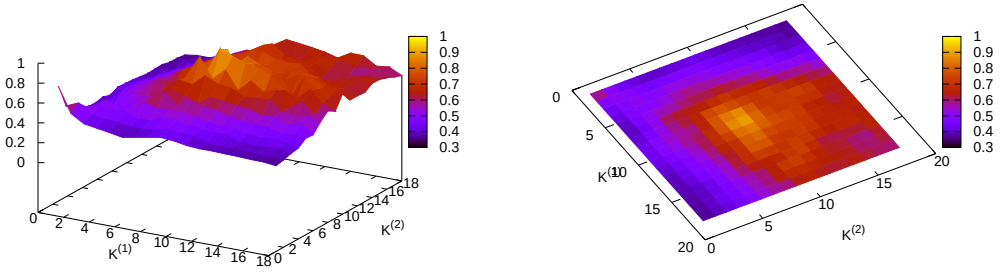
(a) Vue 3D de l'espace des solutions. (b) Vue planne de l'espace des solutions.

Espace des solutions uniquement avec la similarité.



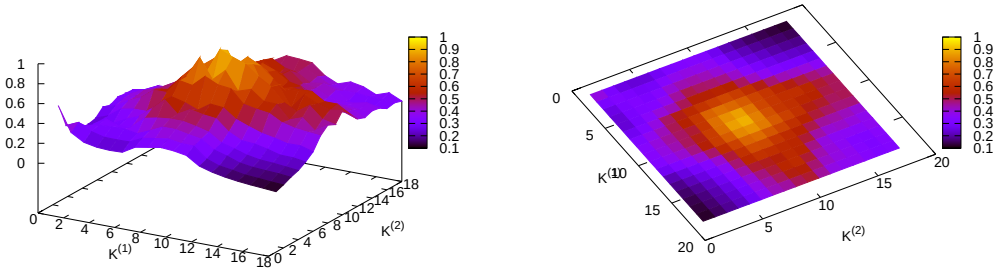
(c) Vue 3D de l'espace des solutions. (d) Vue planne de l'espace des solutions.

Espace des solutions avec l'estimation du nombre de clusters.



(e) Vue 3D de l'espace des solutions. (f) Vue planne de l'espace des solutions.

Espace des solutions en utilisant la connaissance des objets étiquetés.



(g) Vue 3D de l'espace des solutions. (h) Vue planne de l'espace des solutions.

Espace des solutions en utilisant l'estimation du nombre de clusters et la connaissances des objets étiquetés.

Fig. 4.15: *Illustrations de l'espace des solutions en fonction des différents types de connaissances disponibles.*

	Stratégie	Γ	Rand	Jaccard	Folks & Mallows	F-Mesure
2d-4c-no0	$\neg col$	0,655	$0,898\ (\pm 0,020)$	$0,667\ (\pm 0,015)$	$0,805\ (\pm 0,010)$	$0,786\ (\pm 0,013)$
	col	0,913	$0,961\ (\pm 0,019)$	$0,874\ (\pm 0,037)$	$0,932\ (\pm 0,022)$	$0,930\ (\pm 0,025)$
	$kcol$	0,952	$\mathbf{0,970\ (\pm 0,001)}$	$\mathbf{0,887\ (\pm 0,007)}$	$\mathbf{0,941\ (\pm 0,003)}$	$\mathbf{0,940\ (\pm 0,004)}$
2d-4c-no1	$\neg col$	0,677	$0,873\ (\pm 0,019)$	$0,577\ (\pm 0,035)$	$0,743\ (\pm 0,023)$	$0,726\ (\pm 0,029)$
	col	0,826	$0,889\ (\pm 0,018)$	$0,648\ (\pm 0,039)$	$0,789\ (\pm 0,025)$	$0,784\ (\pm 0,030)$
	$kcol$	0,836	$\mathbf{0,901\ (\pm 0,019)}$	$\mathbf{0,682\ (\pm 0,029)}$	$\mathbf{0,812\ (\pm 0,017)}$	$\mathbf{0,810\ (\pm 0,021)}$
2d-4c-no2	$\neg col$	0,613	$0,845\ (\pm 0,028)$	$0,555\ (\pm 0,041)$	$0,728\ (\pm 0,029)$	$0,710\ (\pm 0,033)$
	col	0,861	$0,887\ (\pm 0,141)$	$0,751\ (\pm 0,186)$	$0,855\ (\pm 0,122)$	$0,842\ (\pm 0,146)$
	$kcol$	0,875	$\mathbf{0,944\ (\pm 0,029)}$	$\mathbf{0,826\ (\pm 0,041)}$	$\mathbf{0,903\ (\pm 0,026)}$	$\mathbf{0,901\ (\pm 0,031)}$
2d-4c-no3	$\neg col$	0,707	$0,921\ (\pm 0,015)$	$0,702\ (\pm 0,056)$	$0,827\ (\pm 0,035)$	$0,812\ (\pm 0,041)$
	col	0,955	$0,979\ (\pm 0,001)$	$0,921\ (\pm 0,002)$	$0,959\ (\pm 0,001)$	$0,959\ (\pm 0,001)$
	$kcol$	0,958	$\mathbf{0,980\ (\pm 0,001)}$	$\mathbf{0,924\ (\pm 0,004)}$	$\mathbf{0,961\ (\pm 0,002)}$	$\mathbf{0,961\ (\pm 0,002)}$
10d-4c-no0	$\neg col$	0,508	$0,866\ (\pm 0,023)$	$0,610\ (\pm 0,040)$	$0,772\ (\pm 0,025)$	$0,753\ (\pm 0,030)$
	col	0,824	$0,935\ (\pm 0,060)$	$0,833\ (\pm 0,145)$	$0,907\ (\pm 0,082)$	$0,898\ (\pm 0,090)$
	$kcol$	0,955	$\mathbf{0,971\ (\pm 0,056)}$	$\mathbf{0,926\ (\pm 0,144)}$	$\mathbf{0,958\ (\pm 0,081)}$	$\mathbf{0,955\ (\pm 0,088)}$
10d-4c-no1	$\neg col$	0,528	$0,901\ (\pm 0,035)$	$0,705\ (\pm 0,028)$	$0,830\ (\pm 0,018)$	$0,814\ (\pm 0,021)$
	col	0,857	$0,978\ (\pm 0,026)$	$0,940\ (\pm 0,068)$	$0,966\ (\pm 0,038)$	$0,964\ (\pm 0,041)$
	$kcol$	0,993	$\mathbf{0,998\ (\pm 0,000)}$	$\mathbf{0,992\ (\pm 0,002)}$	$\mathbf{0,996\ (\pm 0,001)}$	$\mathbf{0,996\ (\pm 0,001)}$
iris	$\neg col$	0,632	$0,811\ (\pm 0,027)$	$0,505\ (\pm 0,060)$	$0,677\ (\pm 0,047)$	$0,656\ (\pm 0,053)$
	col	0,745	$0,849\ (\pm 0,171)$	$0,716\ (\pm 0,131)$	$0,832\ (\pm 0,087)$	$0,825\ (\pm 0,109)$
	$kcol$	0,862	$\mathbf{0,925\ (\pm 0,012)}$	$\mathbf{0,802\ (\pm 0,029)}$	$\mathbf{0,889\ (\pm 0,018)}$	$\mathbf{0,889\ (\pm 0,018)}$
wine	$\neg col$	0,448	$0,804\ (\pm 0,024)$	$0,494\ (\pm 0,074)$	$0,666\ (\pm 0,057)$	$0,644\ (\pm 0,065)$
	col	0,744	$0,870\ (\pm 0,079)$	$0,715\ (\pm 0,098)$	$0,833\ (\pm 0,063)$	$0,828\ (\pm 0,073)$
	$kcol$	0,840	$\mathbf{0,920\ (\pm 0,014)}$	$\mathbf{0,789\ (\pm 0,032)}$	$\mathbf{0,881\ (\pm 0,020)}$	$\mathbf{0,881\ (\pm 0,020)}$
segment	$\neg col$	0,499	$0,814\ (\pm 0,045)$	$0,363\ (\pm 0,022)$	$0,554\ (\pm 0,014)$	$0,530\ (\pm 0,026)$
	col	0,746	$\mathbf{0,872\ (\pm 0,008)}$	$\mathbf{0,394\ (\pm 0,025)}$	$0,565\ (\pm 0,025)$	$\mathbf{0,564\ (\pm 0,025)}$
	$kcol$	0,598	$0,799\ (\pm 0,096)$	$0,372\ (\pm 0,059)$	$\mathbf{0,572\ (\pm 0,046)}$	$0,540\ (\pm 0,067)$
ionosphere	$\neg col$	0,540	$0,577\ (\pm 0,007)$	$0,339\ (\pm 0,017)$	$0,524\ (\pm 0,016)$	$0,503\ (\pm 0,020)$
	col	0,720	$\mathbf{0,583\ (\pm 0,004)}$	$0,417\ (\pm 0,017)$	$0,590\ (\pm 0,015)$	$0,589\ (\pm 0,017)$
	$kcol$	0,634	$0,582\ (\pm 0,006)$	$\mathbf{0,424\ (\pm 0,013)}$	$\mathbf{0,596\ (\pm 0,012)}$	$\mathbf{0,595\ (\pm 0,013)}$
vehicle	$\neg col$	0,545	$0,514\ (\pm 0,089)$	$0,198\ (\pm 0,015)$	$0,335\ (\pm 0,020)$	$0,330\ (\pm 0,021)$
	col	0,859	$\mathbf{0,683\ (\pm 0,017)}$	$\mathbf{0,261\ (\pm 0,005)}$	$\mathbf{0,451\ (\pm 0,017)}$	$\mathbf{0,414\ (\pm 0,007)}$
	$kcol$	0,609	$0,561\ (\pm 0,028)$	$0,260\ (\pm 0,009)$	$0,438\ (\pm 0,018)$	$0,412\ (\pm 0,011)$

Tab. 4.4: *Résultats des évaluations de la méthode collaborative avec et sans utilisation de connaissances.*

méthodes avant unification. Nous avons utilisé différents indices de qualité (Rand, Jaccard, Folks & Mallows et F-Mesure) pour évaluer les résultats (une description détaillée de ces indices est donnée en Annexe B).

Pour chaque ensemble de résultats de clustering ($\neg col$, col and $kcol$), la moyenne de la qualité de chaque résultat composant l'ensemble a été calculée pour définir la qualité de l'ensemble. Nous avons effectué 100 exécutions. À chaque fois, les ensembles de résultats initiaux ont été recalculés puis utilisés dans la méthode collaborative et dans la méthode collaborative utilisant des connaissances. L'évaluation des résultats est fournie dans le tableau 4.4, où les valeurs correspondent aux moyennes et aux écarts-types des résultats sur les 100 exécutions.

Comme on peut le constater sur les résultats, la qualité de l'ensemble ayant collaboré (col) est presque toujours supérieure à celle de l'ensemble n'ayant pas collaboré ($\neg col$). En effet, pour chaque jeu de données, la qualité est supérieure sur au moins 5 des 6 indices de qualité. De plus, pour tous les jeux de données, la qualité de l'ensemble fourni par la collaboration utilisant des connaissances ($kcol$) donne encore de meilleurs résultats.

Tous ces résultats montrent, premièrement, que l'étape de collaboration de SAMARAH permet d'améliorer les résultats, et deuxièmement, que l'intégration de connaissances présentées dans ce chapitre est pertinente.

Comparaison par l'évaluation en cascade

L'évaluation en cascade [Candillier et al., 2006] est une approche assez récente pour évaluer la qualité et l'intérêt de résultats de clustering. L'idée est de voir si l'information issue d'un clustering

est pertinente et peut aider l'apprentissage d'un classifieur supervisé. La méthode est basée sur l'enrichissement d'un jeu de données par des résultats de clustering, et de l'utilisation d'une méthode d'apprentissage supervisé pour évaluer l'intérêt de l'ajout d'une telle information.

La méthode consiste à évaluer et comparer le résultat d'un classifieur supervisé quand il est aidé ou non par l'information issue d'un clustering. Si le résultat du classifieur est amélioré par l'information ajoutée par le clustering, il est alors postulé que le clustering porte une information pertinente. De plus, différents résultats de clustering peuvent être comparés, si l'un améliore mieux le résultat fourni par le classifieur supervisé il est fort probable qu'il soit celui qui porte la meilleure information, et donc celui de meilleure qualité.

Nous avons utilisé cette technique pour évaluer en cascade notre clustering collaboratif. Nous nous sommes focalisé sur l'évaluation de la pertinence de l'étape de raffinement présente dans le clustering collaboratif. Nous voulons voir si le raffinement améliore les différents résultats de clustering à travers l'étape de collaboration. Nous souhaitons montrer que l'ensemble des résultats contient un résultat plus pertinent que l'ensemble initial non raffiné.

Pour évaluer les bénéfices de l'utilisation de l'information produite par notre méthode par un algorithme supervisé, nous avons créé différents jeux de données à partir du jeu de données initial, puis nous avons effectué un apprentissage avec un algorithme supervisé.

Soit $X = \{x_1, \ldots, x_n\}$ le jeu de données initial à traiter où l'objet x_i est décrit par m attributs $a_1(x_i), \ldots, a_m(x_i)$. Soit $\mathcal{C}^{(j)}(x_i)$ le cluster de l'objet x_i dans le j-ième résultat initial de clustering. Soit $\mathcal{R}^{(j)}(x_i)$ le cluster de l'objet x_i dans le j-ième résultat de clustering raffiné. Soit $\mathcal{S}^{(j)}(x_i)$ le cluster de l'objet x_i dans le j-ième résultat de clustering raffiné avec les connaissances.

Pour chaque jeu de données initial X, trois jeux de données ont été créés pour intégrer les connaissances provenant des résultats de clustering :

- $D^{(1)} = \{x_1^{(1)}, \ldots, x_n^{(1)}\}$
  où $x_i^{(1)} = \big(a_1(x_i), \ldots, a_m(x_i), \mathcal{C}^{(1)}(x_i), \ldots, \mathcal{C}^{(N)}(x_i)\big)$
- $D^{(2)} = \{x_1^{(2)}, \ldots, x_n^{(2)}\}$
  où $x_i^{(2)} = \big(a_1(x_i), \ldots, a_m(x_i), \mathcal{R}^{(1)}(x_i), \ldots, \mathcal{R}^{(N)}(x_i)\big)$
- $D^{(3)} = \{x_1^{(3)}, \ldots, x_n^{(3)}\}$
  où $x_i^{(3)} = \big(a_1(x_i), \ldots, a_m(x_i), \mathcal{S}^{(1)}(x_i), \ldots, \mathcal{K}^{(N)}(x_i)\big)$

L'algorithme C4.5 [Quinlan, 1996] a été sélectionné comme algorithme d'apprentissage supervisé pour sa capacité à gérer les attributs numériques ainsi que les attributs catégoriels. Nous avons effectué 100 exécutions en validation croisée 10 pour chacune des trois versions du jeu de données.

Les résultats de ces expériences sont présentés dans le tableau 4.5 où les valeurs sont les moyennes et les écarts-types ainsi que les valeurs maximales des précisions pour les quatre jeux de données (c'est-à-dire le jeu de données initial D, le jeu de données enrichi par les clusterings initiaux $D^{(1)}$, le jeu de données enrichi par les clusterings raffinés $D^{(2)}$, le jeu de données enrichi par les clusterings raffinés en utilisant des connaissances $D^{(3)}$). On peut observer que les jeux de données enrichis par les résultats de clustering raffinés donnent de meilleurs résultats sur 6 des 7 jeux de données. On notera que l'étape de raffinement ne dégrade les résultats que dans le cas où les résultats de clustering non raffinés dégradaient également les résultats. Ceci peut être expliqué par un manque de correspondance entre les classes des objets et leur distribution dans l'espace des données. Par conséquent, ajouter une information de clustering ne fait qu'ajouter du bruit au jeu de données.

De plus, on peut également observer une augmentation de la stabilité des résultats. En effet, les écarts-types diminuent de manière significative quand les résultats raffinés sont utilisés à la place des résultats non raffinés. Les résultats raffinés utilisant des connaissances lors de la collaboration donnent encore de meilleurs résultats que les résultats obtenus sans l'utilisation de connaissances. Cependant, les résultats sont moins stables (écarts-types plus élevés) sur la moitié des jeux de données. Ceci peut être expliqué par la forte part d'aléatoire dans la sélection des objets étiquetés

	D	$D^{(1)}$	$D^{(2)}$	$D^{(3)}$
iris	93.33	94.67 (±0.47) ⋄ 94.67	95.47 (±0.30) ⋄ 96.00	**96.40 (±0.60) ⋄ 96.67**
wine	92.13	93.48 (±1.52) ⋄ 95.51	95.73 (±0.50) ⋄ 96.07	**96.18 (±0.73) ⋄ 97.19**
segment	**95.93**	95.79 (±0.17) ⋄ 96.02	95.77 (±0.15) ⋄ 95.97	95.79 (±0.14) ⋄ 95.97
ionosphere	88.03	89.00 (±1.25) ⋄ 90.60	89.40 (±0.93) ⋄ 90.88	**90.94 (±0.31) ⋄ 91.17**
vehicle	69.47	69.17 (±0.87) ⋄ 70.35	69.77 (±0.91) ⋄ 71.23	**70.55 (±0.82) ⋄ 71.61**

Tab. 4.5: *Évaluation de l'intégration des connaissances par l'évaluation en cascade.*

utilisés comme connaissances. Si ces objets sont bien distribués dans l'espace des données et parmi les différents clusters, ils apporteront une information pertinente et aideront à améliorer la collaboration. Il semble envisageable que ce problème soit résolu si les objets étiquetés provenaient d'un expert qui les aurait étiqueté (et non le fruit d'une sélection aléatoire). En effet, l'expert est plus à même de sélectionner des exemples pertinents. Une approche par sélection active (voir section 4.2.2.2) est également envisageable et permettrait à l'expert de fournir des connaissances pendant le processus collaboratif.

4.3.4 Utilisation des connaissances dans les méthodes

Le dernier niveau de connaissance se situe directement au cœur des méthodes utilisées pendant la collaboration. En effet, il est possible de faire collaborer directement des méthodes de clustering semi-supervisées. Il est cependant nécessaire de modifier ces méthodes pour qu'il soit possible d'appliquer les opérateurs de scission, fusion et reclustering (voir section 3.3.2.2) nécessaire à la modification des résultats pendant la collaboration. Une étude préliminaire a été effectuée pour étudier comment modifier la méthode KMEANS méthode et ses opérateurs. Cette approche a donné des résultats encourageants mais nécessite une modification trop importante des différentes méthodes et rend donc difficile son application. Une des étapes de nos futures recherches est d'étudier plus en détail ce problème.

4.4 Bilan

L'intégration de connaissances dans les algorithmes de clustering représente un enjeu important. Dans ce chapitre, nous avons présenté les principales approches existantes permettant cette intégration. Les deux principales représentations de connaissances sont les contraintes entre les objets et l'étiquetage d'objets. Nous avons présenté différentes mesures de pureté qui permettent de tirer parti de la connaissance des étiquettes d'un ensemble d'objets. Il en ressort que les critères évaluant la pureté sans prendre en compte le nombre de clusters peuvent rapidement surévaluer la qualité des résultats. Pour résoudre ce problème, il est possible de prendre en compte une mesure qui va pénaliser les résultats avec un nombre de clusters trop important. D'autres types de critères qui comparent le regroupement de couples d'objets prennent en compte implicitement le nombre de clusters. De plus, nous avons présenté comment de tels critères peuvent être intégrés dans le cadre du clustering collaboratif. Enfin, différents types d'intégration de connaissances en clustering collaboratif ont été présentés et évalués. Les résultats obtenus confirment l'intérêt d'utiliser des connaissances en clustering collaboratif.

Contributions et valorisations

Les recherches présentés dans ce chapitre ont permis de faire évoluer la méthode collaborative existante en lui rajoutant des possibilités d'intégration des connaissances. De nombreux développements ont été réalisés au sein de la méthode pour permettre cette intégration. Ces développements ont été effectués de manière générique pour permettre leur extension à d'autres types de connaissances. Les résultats obtenus et présentés dans ce chapitre ont notamment pu être valorisés à travers trois publications. La première [Forestier et al., 2008e] dans un atelier associé à l'une

des importantes conférences internationales en fouille de données (*IEEE International Conference on Data Mining*). Cette première publication regroupe l'aspect intégration a posteriori des connaissances dans la méthode collaborative. La seconde [Forestier et al., 2010b] dans une conférence internationale (*International Conference on Knowledge Science, Engineering & Management*) où l'intégration de connaissances en clustering sous la forme de critères de pureté a été présentée. Enfin, la troisième publication [Forestier et al., 2010a] a été publiée dans un journal international (*Data & Knowledge Engeneering*) et présente en détail les différentes formes d'intégration proposées dans la méthode collaborative. Ces valorisations ont permis de valider les différentes étapes d'intégration de connaissances et ouvrent des perspectives intéressantes pour le développement de futures extensions de la méthode.

Troisième partie :

Applications en observation de la Terre

Chapitre 5

Connaissances en observation de la Terre

Sommaire

5.1 Introduction

Les géosciences et plus particulièrement l'observation de la Terre via les images de télédétection ont été le domaine privilégié d'application des propositions faites lors de cette thèse. L'observation de la Terre consiste à étudier la surface terrestre afin d'en analyser les propriétés et les évolutions. Les applications sont nombreuses comme par exemple l'étude de l'occupation du sol, des dynamiques urbaines, ou également de l'agriculture. Le domaine de la télédétection fourni de nombreuses données permettant d'effectuer ces analyses. Dans ce chapitre, nous nous intéressons à l'utilisation d'images optiques issues de capteurs embarqués à bord de satellites ou aéroportés. L'image de télédétection est le prototype même d'une donnée complexe de par sa structure physique mais aussi par le fossé sémantique entre les informations bas niveau (les valeurs radiométriques des pixels) et les informations à extraire (par exemple l'occupation du sol). Ce fossé sémantique est défini comme le manque de concordance entre l'information bas niveau et l'interprétation faite par un expert [Smeulders et al., 2000].

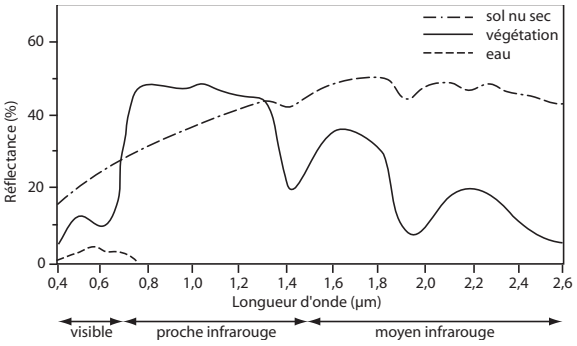


Fig. 5.1: *Exemple de courbe de réflectance caratéristique de trois surfaces.*

5.1.1 Principes et historique de la télédétection

Le rayonnement solaire dont est à l'origine la lumière, permet la perception des objets par un processus appelé illumination. L'énergie produite par le soleil se propage sous forme d'ondes jusqu'à la surface terrestre. Une partie de ce rayonnement est absorbée par l'atmosphère (25%), une autre partie est diffusée (25%) et le reste atteint la surface terrestre (50%) [Desjardins, 2000]. La partie du rayonnement qui atteint la surface est absorbée et réfléchie. C'est ce rayonnement qui est capturé par les instruments de télédétection à différentes positions sur le spectre électromagnétique.

Les propriétés des objets conditionnent à la façon dont ils vont réagir au rayonnement. Les propriétés comme l'absorption, la réflexion et la transmission sont particulièrement importantes. C'est le pouvoir d'absorption et de réflexion spectrale propre à chaque objet qui est à l'origine de sa couleur. Par exemple, un objet bleu absorbe les ondes dans le vert et le rouge tout en réfléchissant celles dans le bleu. La figure 5.1 montre un exemple de trois courbes théoriques de réflectance de trois types de surface le long du spectre électromagnétique. L'objectif des instruments de télédétection est, au sens large, de capturer ces phénomènes d'ondes de manière numérique. La figure 5.2 donne une illustration d'une image satellitaire qui se représente sous la forme d'une matrice de pixels. Chaque pixel est décrit par un ensemble de valeurs correspondant aux bandes du capteur utilisé pour capturer l'image.

Les premières tentatives d'observation de la Terre par photographie depuis un ballon remontent au milieu du 19$^{\text{ème}}$ siècle. Les missions aériennes qui ont eu cours lors du 20$^{\text{ème}}$ siècle ont ensuite permis de constituer une masse de données importante. Ces premières données étaient essentiellement des photo aériennes et ont amené un ensemble de développements dans le domaine de la photo-interprétation. La *photo-interprétation* consiste à identifier un objet au moyen d'une analyse méthodique poussée permettant d'obtenir, par déduction, des renseignements qui ne sont pas directement repérables sur les photographies aériennes, en remontant des apparences de cet objet à sa réalité. Cette démarche suppose, de la part de l'interprète, des connaissances suffisantes des phénomènes et des processus associés à l'objet [Léo Provencher, 2007]. Dans une suite logique, avec l'avènement des satellites à capteurs imageurs dans les années 1970, il est devenu possible d'analyser et d'interpréter automatiquement des images avec une approche similaire.

Le premier satellite de télédétection LANDSAT ne fut mis en orbite qu'en 1972. Il offre des images de la Terre prises à une altitude de 900 km environ. L'engouement pour ce nouveau type de données s'est très vite développé et de nombreux pays ont rapidement compris l'intérêt de posséder de tels moyens d'observation. En effet, la télédétection est à la fois un outil d'inventaire (en occupation des sols, par exemple), d'analyse (en urbanisme, par exemple) et d'aide à la prévision (en agriculture, par exemple). Les principaux enjeux de l'observation de la Terre depuis l'espace sont donc scientifiques, économiques et stratégiques [1].

Différents systèmes d'observation de la Terre ont été conçus et réalisés par des nations ou des groupes de nations pour répondre à ces besoins. À cet égard, la France occupe une place de premier

[1] `http://www.educnet.education.fr/obter/`

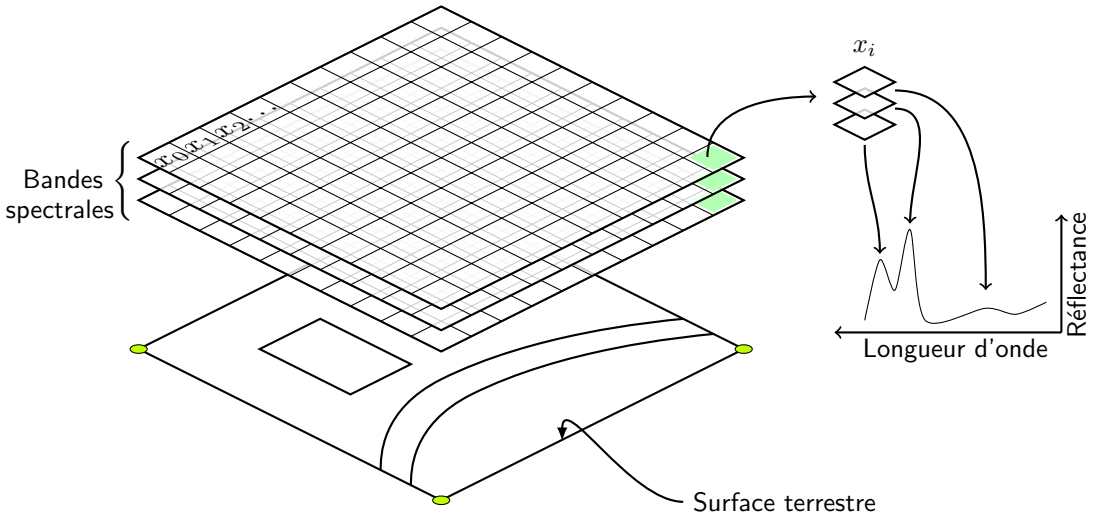


Fig. 5.2: *Schématisation d'une image satellitaire multispectrale.*

plan, notamment par le biais du programme SPOT et de l'agence spatiale française : le Centre National d'Études Spatiales (CNES). Les données acquises par les satellites de télédétection sont maintenant largement diffusées, et le marché des données satellitaires s'est fortement développé.

5.1.2 Une image est une donnée complexe

Il existe de nombreux capteurs utilisées en télédétection permettant de capturer des informations sur la réflectance des surfaces. Les deux principales caractéristiques de ces capteurs sont la résolution spatiale et la résolution spectrale.

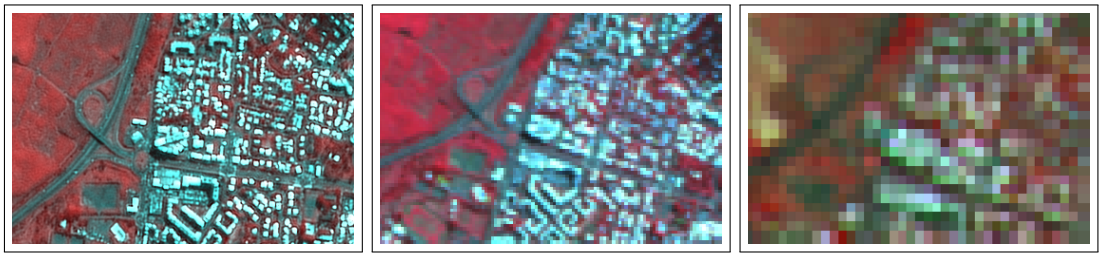
La **résolution spatiale,** c'est-à-dire la capacité du capteur à distinguer deux objets à une certaine résolution, varie de plusieurs dizaines de mètres à une résolution inférieure au mètre. On distingue généralement, dans le domaine civil, la basse résolution (1000m), la moyenne résolution (80m), la haute résolution (HR) (10 à 30m) et la très haute résolution (THR) (inférieure à 5m). En pratique, cette résolution peut se voir comme la taille des pixels. La figure 5.3 présente des images de l'île de La Réunion capturées par le satellite SPOT extraites de la base de données Kalideos-Réunion (© CNES) à trois résolutions différentes (2m, 10m et 20m).

La **résolution spectrale**, c'est-à-dire la taille de l'intervalle de chaque bande spectrale varie environ de 0,2μm à 10μm suivant les capteurs. Le nombre de bandes peut aller de trois ou quatre (© SPOT, © QUICKBIRD) jusqu'à plusieurs centaines (© DAIS). On distingue généralement quatre régions spectrales [Bonn et Rochon, 1992] :

- le visible : 0,4μm à 0,7μm ;
- le proche infrarouge : 0,7μm à 1,5μm ;
- le moyen infrarouge : 1,5μm à 3μm ;
- l'infrarouge lointain : 3μm à 15μm.

L'extraction d'informations contenues dans une image de télédétection peut être réalisée manuellement par un photo-interprète. Ce processus d'interprétation visuelle est cependant consommateur de temps, d'autant plus que la fréquence d'acquisition augmente avec l'amélioration des technologies. Les futurs capteurs à forte revisite temporelle (Venμs, Sentinel-2, etc.) proposeront dès 2011 des acquisitions tous les deux à trois jours. L'automatisation de l'extraction d'informations devient alors encore plus nécessaire. L'extraction automatique se fait principalement par des techniques de traitement d'images et des techniques de fouille de données.

Jusqu'à la fin des années 1990, l'un des principaux problèmes liés aux images de télédétection résidait dans le fait qu'un pixel peut représenter une zone composée de plusieurs objets de natures



(a) Image à 2 m (b) Image à 10 m (c) Image à 20 m

Fig. 5.3: *Même zone capturée par différents capteurs des satellites SPOT (Ile de La Réunion) à différentes résolutions spatiales.*

très différentes. On parle de pixel mixte. De fait, les images à faible résolution ne permettaient qu'une analyse de zones thématiques (zone bâtie, végétation, eau, etc.). De nombreuses méthodes pour analyser de telles images ont été développées et ont prouvé leur efficacité. Les méthodologies utilisées par les géographes dans les approches par pixels sont principalement axées sur des approches supervisées. La classification non supervisée est généralement utilisée par l'expert pour obtenir des informations sur la structure des classes.

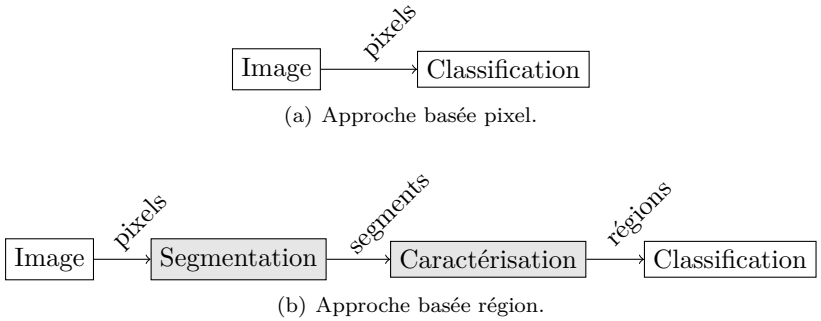
Avec l'arrivée des capteurs à très haute résolution (THR) spatiale, les données sont devenues de plus en plus complexes. En effet, l'augmentation de la résolution spatiale des images a conduit à une augmentation importante des détails présents dans celles-ci. Les approches qui ont été développées ces vingt dernières années pour le traitement des images HR montrent leurs limites pour le traitement de ce nouveau type de données. En effet, alors que les images à faible résolution ne permettaient qu'une analyse de zones thématiques, les images à très haute résolution permette une analyse des objets géographiques eux-mêmes. Paradoxalement, l'augmentation de la précision spatiale des images entrainent également des perturbations dans l'analyse. En effet, des objets invisibles auparavant font leur apparition (voiture, passage piéton, travaux, etc.). L'ombre portée joue également une rôle plus important dans ces images. L'avènement de la très haute résolution spatiale a ainsi conduit au développement de nouvelles méthodes de traitement et d'interprétation des images satellitaires.

5.1.3 Paradigme basé région

La nécessité de prendre en compte le contexte des pixels, l'augmentation de la complexité de l'information contenue dans les images, ou encore l'augmentation de la fréquence d'acquisition de celles-ci ont débouché récemment sur des propositions de nouvelles approches. De manière schématique, on peut regrouper les approches de traitement des images de télédétection en deux groupes avec d'une part, les méthodes basées pixel qui sont principalement utilisées pour traiter les images HR, et d'autre part, les méthodes basées région principalement utilisées pour traiter les images THR. La figure 5.4 illustre les étapes de ces deux approches.

Les **approches basées pixel** (voir figure 5.4 (a)) consistent à considérer les pixels de l'image comme les données à traiter. L'objet élémentaire est un pixel, et le nombre d'objets à traiter est le nombre de pixels dans l'image. Il sera à la charge de l'expert de faire correspondre les groupes identifiés automatiquement avec une sémantique c'est-à-dire aux classes d'occupation du sol.

Les **approches basées région** (voir figure 5.4 (b)) consistent à effectuer une première étape de segmentation de l'image. Le terme *orienté objet* a également été utilisé pour décrire ces méthodes mais a été progressivement abandonné pour éviter toute confusion avec le concept homonyme utilisé pour les langages de programmation. On appellera *objet réel* ou *objet géographique* un objet présent dans la scène étudiée (par exemple un pavillon, une route, etc.). L'étape de segmentation utilisée consiste à regrouper des pixels en vue de constituer des agrégats de pixels homogènes connexes appelées régions qui sont ensuite caractérisées par de multiple attributs (taille, forme, etc.). Chaque



(a) Approche basée pixel.

(b) Approche basée région.

Fig. 5.4: *Étapes des paradigmes pixel (a) et région (b) en interprétation d'images de télédétection.*

région est ainsi décrite selon ses propriétés intrinsèques. Ce sont ensuite ces régions qui seront classées. Le nombre d'objets à classer est donc le nombre de régions issues de la segmentation de l'image. Il est important de noter que le mécanisme de construction de ces régions n'est pas une étape triviale. En effet, il n'existe pas à l'heure actuelle d'algorithme de segmentation parfait. Des pixels correspondants à d'autres objets réels peuvent être inclus dans la région associée à un objet réels. Réciproquement, des pixels correspondants à l'objet réel peuvent avoir été omis. Enfin, certaines régions peuvent ne correspondre à aucun objet réel de la scène observée. Le problème de la caractérisation des régions pose également de nombreux verrous, car le nombre de caractéristiques potentiellement calculables pour une région est très élevé.

Ce nouveau paradigme région permet d'aborder le problème de l'interprétation d'image de manière différente qu'avec une approche pixel. En effet, ce nouveau cadre de raisonnement permet de faire directement le parallèle entre les régions et les objets réels ce qui n'était pas possible avec les pixels. Dans les approches pixel, les éléments individuels à classer (c'est-à-dire les pixels) n'avaient pas vraiment de sémantique car ils ne portaient que peu d'information individuellement. La sémantique était portée par les groupes de pixels qui leur donnaient un sens. Dans le paradigme région, les éléments individuels ont maintenant également une sémantique. On va par exemple chercher à identifier les *pavillons* qui est un concept que l'on peut directement relier à sa représentation physique dans le monde réel. On parlera donc de concept comme étant une représentation générale et abstraite de la réalité.

5.2 Connaissances en observation de la Terre

La définition de l'interprétation d'images donnée dans Léo Provencher [2007] fait appel à la notion de *connaissances* que doit posséder l'interpréteur humain pour être capable d'effectuer l'interprétation d'une image. Dans un but d'automatisation du processus d'interprétation des images à très haute résolution, nous allons voir dans la suite de ce chapitre nos propositions pour représenter et utiliser des connaissances expertes. Dans nos recherches, nous nous intéressons plus particulièrement à la représentation des connaissances sur les objets urbains. En effet, ces travaux ont été réalisés dans le cadre de l'ACI Masse de Données Fodomust en collaboration avec le laboratoire Image, Ville, Environnement [2].

5.2.1 Acquisition des connaissances sur les objets géographiques

Au cours de ces dernières années, plusieurs auteurs se sont intéressés à la modélisation de la connaissance sur les objets géographiques. Smith et Mark [2000] ont mené une série d'expérimentations pour essayer d'établir comment des sujets non experts conceptualisent les objets géographiques. Des questions, comme par exemple *"Qu'est ce qu'un concept géographique pour vous ?"* sont posées

[2] `http://imaville.u-strasbg.fr/`

à des sujets qui doivent essayer d'en donner une définition. Les auteurs ont constaté de très grandes variations dans les réponses en fonction des sujets. Une autre étude menée par Mark et al. [1999] avec des questions de la forme *"Lister 5 choses qui sont généralement vraies à propos de X, où X est un terme géographique (lac, montagne, rivière, etc.)"* montre aussi une hétérogénéité des réponses. Plus récemment une étude détaillée a été effectuée par Gahegan et Pike [2006] sur la représentation de connaissances en géographie et soulève les problèmes récurrents en modélisation de l'information géographique (par exemple la complexité de définir une représentation générique).

Les connaissances géographiques sont assez facilement accessibles en langage naturel, dans un livre de géographie ou en questionnant un géographe. Cependant en intelligence artificielle il est nécessaire que ces connaissances soient exploitables par une machine. Celles-ci doivent être formalisées pour que la machine puisse leur donner un sens et agir sur elles. Nous allons étudier quelles sont les sources de connaissances généralement disponibles puis nous verrons comment ces connaissances ont été représentées afin de les utiliser.

5.2.1.1 Sources des connaissances

La principale source de connaissances est celle de l'expert du domaine, c'est-à-dire le photo-interprète cartographe des objets géographiques. En effet, celui-ci est souvent capable de fournir une interprétation des images à analyser. Il dispose a priori des connaissances suffisantes lui permettant d'effectuer l'interprétation des images. Cependant, la réalité est plus complexe et la majorité des processus mis en œuvre par l'expert lors d'une interprétation est fondée sur des critères utilisés dans le domaine de la photo-interprétation tel que la couleur/teinte, la forme, la taille, la texture, la structure, le contexte [Léo Provencher, 2007]. Le fait d'exprimer sa connaissance et les processus utilisés pour l'interprétation est une tâche complexe. Il existe un fossé sémantique (*semantic gap*) de représentation entre ce que peut formaliser l'expert et la réalité des images. Si l'expert photo-interprète peut reconnaître une route immédiatement de part sa couleur grise et sa forme (linéaire et continue dans l'espace), il peut lui être difficile d'exprimer directement des valeurs numériques utilisables par un système informatique. Il existe un fossé sémantique entre sa connaissance et les données à traiter qui ne sont pas de même nature. Pour autant, il est possible d'extraire un ensemble de connaissances de l'expert. En effet, il nous semble possible (voir indispensable) de se baser sur les connaissances de l'expert pour mener à bien la tâche d'interprétation.

Un autre ensemble de connaissances peut également être extrait en faisant appel à des processus de fouille de données tels que des calculs statistiques sur des bases de données d'objets existants. On citera par exemple la BD TOPO (© IGN) qui contient une description d'éléments du paysage sous forme de vecteurs de précision métrique. De plus, acquérir de la connaissance en effectuant des entretiens prend en général beaucoup de temps, problème bien connu dans la communauté de l'intelligence artificielle. Ainsi, pour faciliter l'acquisition des connaissances, des mécanismes d'apprentissage ont été mis en place pour extraire automatiquement des connaissances des données brutes (les images). Des outils d'apprentissage symbolique ont été utilisés par David Sheeren lors de son post-doctorat au sein de notre laboratoire [Sheeren et al., 2006a,b]. Cette étape importante a permis de modéliser une partie des connaissances en réduisant le fossé sémantique entre les données et l'expert. Des connaissances supplémentaire peuvent également être extraites de librairies spectrales, qui sont des bases de données de spectres décrivant en détail les propriétés spectrales de certains type de surface. Ces informations ont été utilisées pour formuler la description spectrale des objets. Nous verrons plus en détail l'utilisation des librairies spectrales dans le chapitre suivant.

Nous allons présenter dans le cadre de cette thèse, comment les connaissances provenant de ces différentes sources ont été renseignées dans un dictionnaire de données. Puis, nous étudierons comment ce document a été utilisé pour générer un formalisme plus structuré séparé en une hiérarchie de concepts et un ensemble de connaissances sur ces concepts. En effet, la description dans le dictionnaire de données est majoritaire textuelle, et il a été nécessaire de mieux représenter les informations pour pouvoir les utiliser dans un processus d'identification d'objets géographiques. Il est à noter que notre volonté n'était nullement de décrire de manière exhaustive le domaine de la géographie urbaine mais plus particulièrement de pallier au manque de représentation des connaissances du domaine opérables nécessaires à nos travaux sur l'interprétation d'images.

THR1 – QB MS (2.8 m)			
Bâtiment de l'habitat collectif discontinu (type grand ensemble)	 Bâtiment de l'habitat collectif	 Bâtiment de l'habitat collectif de type "barre"	 Bâtiment de l'habitat collectif de type "tour"

Fig. 5.5: *Illustration graphique du concept Bâtiment.*

5.2.1.2 Dictionnaire des données

Le recueil des connaissances du domaine par l'expert en photo-interprétation d'images en milieu urbain est résumé dans un dictionnaire de données réalisé par le laboratoire Image, Ville, Environnement. Ce dictionnaire est un document textuel présenté sous la forme de fiches précisant pour chaque classe d'objets (également appelée *concept*) susceptible d'être identifiée sur une image THR en milieu urbain, les éléments suivants :

1. **Identification** :
   - Son nom et son éventuel lien hiérarchique avec les autres concepts.
2. **Description dans le monde réel** :
   - Définition textuelle. Par exemple :

     « La classe *bâtiment de l'habitat collectif discontinu* appelés aussi *immeuble* appartient à la catégorie de classe élémentaire *bâtiment*. Il désigne une construction durable importante, destinée à l'habitation collective en appartements ou à des activités secondaires ou tertiaires, à plusieurs étages. On distingue le plus souvent les immeubles en *barre* et *tour*. Une *barre* est un bâtiment de forme rectangulaire (étroit et allongé au sol) de plus de 4 étages. Une *tour* est un bâtiment de forme carrée, de volume allongé vers le haut, de plus de 6 étages. Les tours et les barres sont organisées en TU (tissus urbain) discontinu de type Grand Ensemble (GE). Les immeubles de ce type de TU se caractérisent par la régularité des directions principales (parallélisme, orthogonalité). En général, un immeuble : est situé dans un îlot physique (domaine privé) ; a une emprise au sol de plus de 30 m^2 ; est associé à des surfaces artificialisées (route, parking, espaces verts, aire de jeux) organisées en quartiers (généralement relativement récents, construits après-guerre) »
   - Illustration graphique : voir figure 5.5
3. **Description dans l'image** :
   - Nature de l'objet : quelle forme (simple ou composition) prend l'objet selon la résolution spatiale utilisée.
   - Définition textuelle relative à son identification dans l'image. Par exemple :

     « Un objet de la classe *immeuble* est représenté graphiquement par un polygone dont la surface correspond à l'emprise au sol du bâtiment. »
   - Principales relations : notamment l'adjacence avec d'autres classes, s'il existe une notion d'alignement entre des objets de cette classe ainsi que la distance entre des objets de cette classe si cela est approprié. Par exemple :

     « Adjacence possible de la classe avec les objets de type végétation et de type route. La distance entre les objets immeubles est moyenne à élevée. »
   - Attributs : type de signature spectrale, longueur, largeur, élongation, surface, critères géométriques, type de texture, etc.
4. **Commentaires** : liens vers d'autres sources de données ou d'exemples possibles. Par exemple

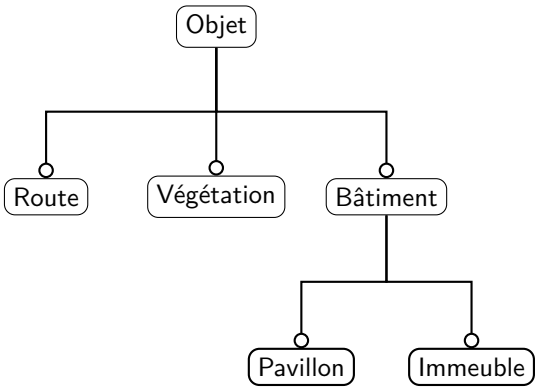


Fig. 5.6: *Exemple de hiérarchie de concepts.*

« Il existe une BD des bâtiments de l'agglomération : elle correspond à la BD TOPO (© IGN). Cette production, qui date de 1989, n'est pas mise à jour. »

Le dictionnaire des données construit pour le projet Fodomust comprend environ 90 concepts mais tous n'ont pas pu être renseignés par manque de temps et de données disponibles.

5.2.1.3 Formalisme

Le dictionnaire des données représente un dépôt des connaissances de l'expert. Cependant, les connaissances formulées dans ce document ne sont pas utilisables directement d'un point de vue informatique. Pour cela, il est nécessaire de convertir les informations dans une forme plus adaptée. Nous avons adopté une représentation XML[3]. XML est un langage de représentation de données semi-structurées. Son principal intérêt est qu'il est simple à écrire et à lire que ce soit informatiquement ou humainement. Il autorise aussi une grande liberté sur la manière de représenter la sémantique de la base de connaissances.

La base de connaissances présentée ici peut être apparentée à une ontologie. D'après Gruber [1995], une ontologie est une spécification explicitée d'une conceptualisation. Plus simplement, une ontologie décrit le vocabulaire utilisé pour étudier un domaine. A contrario, une base de connaissances a pour vocation d'être utilisée pour résoudre un problème ou des requêtes particulières dans un domaine. Il existe donc une différence fonctionnelle entre la base de connaissances, utilisée ici, et une ontologie.

La traduction du dictionnaire des données au formalisme de la base de connaissances a été faite de façon manuelle. Nous avons ici séparé la connaissances issue du dictionnaire des données en deux parties : une hiérarchie de concepts, et les caractéristiques et attributs des concepts.

5.2.1.4 Hiérarchie de concepts

Il existe dans le dictionnaire un lien hiérarchique entre certains concepts. Par exemple, le concept Immeuble est une spécialisation du concept Bâtiment. Inversement, le concept Bâtiment est subsumé par le concept Immeuble, c'est-à-dire que tout objet du concept Immeuble appartient aussi au concept Bâtiment. La première étape a donc consisté à créer une hiérarchie de concepts à partir des informations disponibles dans le dictionnaire des données.

Un exemple de hiérarchie de concepts est donné à la figure 5.6. En haut se situe le concept le plus général Objet. Sur cet exemple, trois concepts spécialisent le concept Objet, il s'agit de Bâtiment, Route et Végétation. Le concept Bâtiment peut se spécialiser en Immeuble et Pavillon.

Le dictionnaire des données contient de nombreux concepts (environ 90), ce qui permet de définir une hiérarchie relativement imposante. Cependant, cette information ne donne qu'une information

[3]eXtensible Markup Language, voir `http://www.w3.org/TR/REC-xml/`

sur les liens hiérarchiques entre les concepts. Il est également nécessaire d'avoir des informations sur les concepts eux mêmes pour pouvoir être capable de les identifier dans une image.

5.2.1.5 Connaissances sur les concepts

Une fois la hiérarchie de concept créée, il est nécessaire d'affecter à chaque concept la connaissance qui a pu être issue du dictionnaire de données. Plusieurs types de connaissances peuvent être ajoutés :

- **Intervalle sur des attributs quantitatifs**. Nous avons vu avec l'exemple du concept Immeuble, que les objets de ce concept ont leurs attributs *superficie* (emprise au sol) supérieur à 30m^2 d'après l'expert. On peut en déduire que pour être défini comme un immeuble (règle nécessaire), il faut que la superficie d'une région soit dans l'intervalle $[30; \infty[$ m^2.
- **Méta-connaissance sur les attributs discriminants du concept**. Pour chaque concept, les attributs discriminants ne sont pas forcément les mêmes. Par exemple, pour discriminer un objets du concept Végétation la forme a peu d'importance car les différents types de végétation peuvent revêtir de nombreuses formes différentes alors que pour le concept Pavillon la forme a une grande importance, car classiquement un pavillon est de forme rectangulaire. Ainsi, à chaque attribut est associé une pondération (actuellement fixée par l'expert) prise dans l'intervalle $[0; 1]$ avec 0 montrant que l'attribut est sans intérêt.
- **Attributs qualitatifs.** Cette catégorie comprend des informations qualitatives sur la classe. On y retrouve par exemple les connaissances sur les relations des objets de cette classe avec d'autres objets (héritage). Par exemple, l'adjacence ou la proximité avec des objets d'autres classes.

À partir de ces connaissances sur les concepts, il est possible d'utiliser des mécanismes d'inférence pour réaliser des déductions. En effet, l'utilisateur a fourni des connaissances du domaine qui ont été formalisées dans la base de connaissances. Il est possible, à partir de ces connaissances, de les combiner pour déduire de nouvelles connaissances qui n'étaient pas explicites. Il est par exemple possible de déduire que les objets du concept Immeuble appartiennent également au concept Bâtiment, car le concept Bâtiment subsume le concept Immeuble. De fait, il est possible de déduire des connaissances supplémentaires en transférant les connaissances du concept Bâtiment au concept Immeuble (par exemple que l'emprise au sol doit être supérieure à 30m^2).

5.2.2 Modélisation des connaissances sur les objets géographiques

L'objectif est de modéliser les connaissances et de proposer un processus d'identification permettant d'utiliser cette base de connaissances pour identifier des objets géographiques dans des images de télédétection. Nous allons décrire dans cette section comment nous avons utilisé les informations présentes dans la hiérarchie et dans les concepts pour effectuer de l'identification d'objets géographiques. Notre objectif est de proposer un mécanisme qui, étant donné une image et sa segmentation, affecte un concept à chacune des régions issues de la segmentation. Nous allons donc chercher, pour chaque région, le concept le plus similaire à celle-ci si celui-ci existe dans la base de connaissances. Ce processus, qui consiste à chercher le ou les concepts correspondant à une région, est appelé *appariement* au encore *identification*.

5.2.2.1 Systèmes à base de connaissances

Les systèmes à base de connaissances ont prouvé leur efficacité pour l'identification d'objets complexes [Liu et al., 1994], ainsi qu'en interprétation d'images [Ogiela et Tadeusiewicz, 2008]. Par exemple, les systèmes SIGMA [Matsuyama et Hwang, 1990] et SCHEMA [Draper et al., 1989] effectuent des analyses d'images aériennes en utilisant plusieurs descripteurs de régions. Ces systèmes permettent d'accéder à un niveau sémantique élevé. Néanmoins, comme le soulignent Crevier et Lepage [1997], ces systèmes sont très dépendants du domaine car ils intègrent des connaissances a priori sur l'image à analyser. Leurs inconvénients sont que les connaissances du domaine ne sont pas clairement séparées de la partie procédurale et que la base de connaissances est difficile à produire.

C'est pourquoi des travaux récents ont proposé d'utiliser des systèmes à base d'ontologies pour décrire plus clairement les connaissances du domaine. Zlatoff et al. [2004] utilisent les relations spatiales entre les concepts pour fusionner les régions et identifier les objets susceptibles d'appartenir aux concepts. L'utilisation quasi-exclusive des relations spatiales se prête difficilement à notre domaine d'application (imagerie satellitaire à très haute résolution) comme la structure des images est très complexe. Cependant, des travaux prometteurs [Vanegas et al., 2009] sont en cours de réalisation sur l'utilisation de contraintes spatiales floues. Le système RCC-8 est notamment très populaire pour représenter les relations entres les objets géographiques [Alboody et al., 2009].

De manière similaire, Maillot et Thonnat [2008] ont proposé un processus d'apprentissage basé sur une ontologie pour créer un système d'identification en analyse d'images. Un point intéressant est la séparation entre une mesure d'appariement local et un mécanisme d'appariement global (c'est-à-dire une mesure qui combine les mesures locales calculées lors des appariements avec les concepts). Les descripteurs utilisés lors de l'appariement sont des *concepts visuels* qui sont acquis lors d'une phase d'apprentissage. La fonction d'appariement est dépendante de ces concepts visuels. Les auteurs proposent que la mesure d'appariement global prenne en compte la hiérarchie des concepts. Panagi et al. [2006] proposent quant à eux d'utiliser un algorithme génétique d'analyse sémantique basé sur une ontologie. Des attributs de bas niveau sont extraits de l'image et utilisés pour correspondre avec l'ontologie. Un ensemble d'hypothèses (régions, liste des concepts possibles et degré de confiance) sont testées avec un algorithme génétique pour déterminer si l'interprétation de l'image est acceptable. Encore une fois, seules les relations spatiales sont utilisées dans ce système. Enfin, Athanasiadias et al. [2007] présentent une plateforme pour la segmentation et l'identification d'objets en utilisant une ontologie dans le domaine du multimédia.

Nous allons à présent proposer un algorithme d'identification guidé par des connaissances du domaine inspiré de ces travaux et prenant en compte le formalisme spécifique des connaissances disponibles dans notre domaine.

5.2.2.2 Formalisation des connaissances

Parmi les travaux présentés dans Roussey [2001] pour modéliser de la connaissance, les approches à base de *frames* semblent les plus appropriées pour les problèmes de classification d'image [Marino Drews, 1993]. Dans ces systèmes [Minsky, 1975], les connaissances sont regroupées dans une hiérarchie de *frames*. Un *frame* est un prototype (c'est-à-dire un objet représentatif d'une famille) composé d'un ensemble de *slots* (attributs) décrivant les propriétés du prototype. Nous nous basons sur cette représentation dans nos travaux.

Ainsi la formalisation des connaissances retenue est la suivante :

- Soit $\mathscr{C}_i$ un concept de la base de connaissances ;
- Soit Θ l'ensemble des concepts disponibles ;
- Soit $\rho(\mathscr{C}_i)$ la profondeur du concept $\mathscr{C}_i$ dans la hiérarchie ;
- Soit $\mathscr{A}$ l'ensemble de tous les attributs présents dans la base de connaissances.

Définition (Sous-concept) La relation de sous-concept $\preceq_\Theta$ entre deux concepts est un ordre partiel défini par :

$$\forall(\mathscr{C}_i, \mathscr{C}_j) \in \Theta^2,\ \mathscr{C}_i \preceq_\Theta \mathscr{C}_j \text{ signifie que } \mathscr{C}_i \text{ est un sous-concept de } \mathscr{C}_j.$$

Un tel concept $\mathscr{C}_j$ est appelé *super-concept de* $\mathscr{C}_i$, c'est-à-dire que $\mathscr{C}_j$ est un concept plus générique que $\mathscr{C}_i$.

Par exemple, $\mathsf{Pavillon} \preceq_\Theta \mathsf{Bâtiment}$, signifie que la classe Bâtiment est plus générique que Pavillon. En effet, la classe Bâtiment regroupe tous les bâtiments y compris les pavillons.

Définition (Attributs) L'ensemble des attributs $\mathscr{A}_{\mathscr{C}_i}$ associés à un concept $\mathscr{C}_i \in \Theta$ est défini comme l'ensemble des attributs du concept $\mathscr{C}_i$ sur lesquels un ensemble de définition (plus restreint que l'espace de définition original) a été donné par l'expert.

Type d'attribut	Attribut	Poids	Intervalle	
			min	max
Spectral	Bleu	1	21.7	62.3
	Vert	1	19.4	80.1
	Rouge	1	29.7	135.1
	Proche IR	1	34.8	139
	NDVI	1	50.2	108
Spatial	Diamètre(m)	0.8	13	61
	Surface (m^2)	1	10	600
	Élongation (m)	0.6	1	3.1

Tab. 5.1: *Exemple du concept Pavillon.*

$\omega(a, \mathscr{C}_i)$ est le poids de l'attribut a associé au concept $\mathscr{C}_i$. Ce poids reflète l'importance de l'attribut a pour décrire le concept $\mathscr{C}_i$ selon l'expert.

Définition (Valeurs et espace de définition des attributs) Soit $a \in \mathscr{A}$ un attribut, $\mathcal{V}(R, a)$ est la valeur de l'attribut a pour la région R. $min(\mathscr{C}_i, a)$ (respectivement $max(\mathscr{C}_i, a)$) représente la valeur minimale (respectivement maximale) que peut prendre l'attribut a pour pouvoir décrire un objet du concept $\mathscr{C}_i$.

Le concept Pavillon est donné en exemple dans le tableau 5.1.

Définition (Attributs spécifiques et hérités) L'ensemble des attributs spécifiques $\mathscr{A}^{(s)}_{\mathscr{C}_i}$ est l'ensemble des attributs dont l'espace de définition est défini dans le concept $\mathscr{C}_i$. L'ensemble des attributs hérités $\mathscr{A}^{(h)}_{\mathscr{C}_i}$ est l'ensemble des attributs dont l'espace de définition n'est pas défini dans le concept $\mathscr{C}_i$ mais s'applique à ce concept car défini dans un super-concept de $\mathscr{C}_i$.

5.2.3 Connaissances pour l'identification d'objets géographiques

Dans la section précédente nous avons étudié comment les connaissances sont formalisées, nous allons maintenant étudier le mécanisme les utilisant pour l'identification d'objets. Nous allons d'abord présenter le calcul du score d'appariement utilisé pour calculer le taux de similarité entre une région issue d'une segmentation et un concept de notre base de connaissances. Nous présentons ensuite l'algorithme de parcours (Algorithme 3) dans la hiérarchie de concepts. Enfin, nous présentons des expériences pour illustrer les résultats produits par la méthode d'identification proposée.

5.2.3.1 Mise en œuvre des connaissances

Le calcul du score d'appariement que nous proposons est basé sur une approche orientée attributs. Il consiste à vérifier la validité des valeurs des attributs d'une région par rapport aux intervalles définis dans les concepts de la base de connaissances. Néanmoins, les descriptions des concepts n'ont pas structure sémantique unifiée. Par exemple, le concept Pavillon est défini par un grand nombre d'attributs de forme (élongation, taille, etc.) et certains attributs spectraux, alors que la classe Ombre n'est définie que par des attributs spectraux, les attributs de forme étant non pertinents pour ce concept. Une région peut, a priori, être comparée à n'importe quel concept. L'ensemble des attributs utilisés comme description peut être différent pour chaque concept. Sans connaissance a priori, cette dissymétrie implique de calculer tous les attributs d'une région, même si une partie de ces attributs est non pertinente relativement au concept de la région non encore connue. Il n'est donc pas possible d'utiliser les mesures comme MDSM [Rodriguez et Egenhofer, 2003] ou les mesures présentées dans [Tversky, 1977] et [Schwering et Raubal, 2005].

Nous proposons donc un nouveau score d'appariement basé sur la similarité entre les valeurs des attributs pour une région donnée et l'espace de définition de cet attribut pour le concept donné.

Ce score est composé d'une similarité locale (correspondance d'une région avec un concept donné) et une similarité globale (correspondance d'une région avec une hiérarchie de concepts).

La mesure de similarité locale compare les valeurs des attributs d'une région avec les attributs spécifiques d'un concept. Pour définir la mesure de similarité locale, nous avons besoin de définir le degré de validité entre un concept $\mathscr{C}_i$ et une région R pour un attribut a donné qui mesure à quel point la région R respecte les contraintes du concept $\mathscr{C}_i$ sur l'attribut a.

Définition (Degré de validité) Soit $\texttt{Valid}(a, \mathscr{C}_i, R)$ le degré de validité pour un attribut a entre une région R et un concept $\mathscr{C}_i$.

$$\texttt{Valid}(a, \mathscr{C}_i, R) = \begin{cases} 1 & si \;\; \mathcal{V}(R,a) \in [\min(\mathscr{C}_i, a); \max(\mathscr{C}_i, a)] \\ \frac{\mathcal{V}(R,a)}{\min(\mathscr{C}_i, a)} & si \;\; \mathcal{V}(R,a) < \min(\mathscr{C}_i, a) \\ \frac{\max(\mathscr{C}_i, a)}{\mathcal{V}(R,a)} & si \;\; \mathcal{V}(R,a) > \max(\mathscr{C}_i, a) \end{cases} \tag{5.1}$$

Définition (Similarité locale) Soit $\texttt{Sim}(R, \mathscr{C}_i)$ la similarité locale entre une région R et un concept $\mathscr{C}_i$.

$$\texttt{Sim}(R, \mathscr{C}_i) = \frac{\sum\limits_{a \in \mathscr{A}^{(s)}_{\mathscr{C}_i}} \omega(a, \mathscr{C}_i)\ \texttt{Valid}(a, \mathscr{C}_i, R)}{\sum\limits_{a \in \mathscr{A}^{(s)}_{\mathscr{C}_i}} \omega(a, \mathscr{C}_i)} \tag{5.2}$$

Le score de similarité globale évalue la pertinence de la correspondance entre une région R et un concept $\mathscr{C}_i$ en prenant en compte la hiérarchie des concepts.

Le score d'appariement est une combinaison linéaire entre les similarités locales. En effet, les similarités locales des concepts génériques sont propagées par héritage aux concepts plus spécifiques. Dans ce calcul, nous intégrons un coefficient de spécialisation ρ égal à la profondeur du concept dans la hiérarchie. Par ce moyen, la mesure favorise la spécialisation des concepts, indiquant qu'il est toujours plus intéressant d'identifier les concepts le plus bas possible dans la hiérarchie.

Définition (Score d'appariement) Soit $\mathcal{P}(\mathscr{C}_i)$ le chemin dans la hiérarchie de concepts commençant à la racine et finissant sur le concept $\mathscr{C}_i$ défini par :

$$\mathcal{P}(\mathscr{C}_i) = \{\mathscr{C}_j \mid \mathscr{C}_i \preceq_\Theta \cdots \preceq_\Theta \mathscr{C}_2 \preceq_\Theta \mathscr{C}_1\}$$

Le score d'appariement $\texttt{Score}(R, \mathscr{C}_i)$ entre une région R et un concept $\mathscr{C}_i$ est défini par :

$$\texttt{Score}(R, \mathscr{C}_i) = \frac{\sum\limits_{\mathscr{C}_j \in \mathcal{P}(\mathscr{C}_i)} \rho(\mathscr{C}_j)\ \texttt{Sim}(R, \mathscr{C}_j)}{\sum\limits_{C_j \in \mathcal{P}(\mathscr{C}_i)} \rho(\mathscr{C}_j)} \tag{5.3}$$

À présent que le score d'appariement entre une région et un concept a été défini, il est nécessaire de détailler l'algorithme d'exploration de la hiérarchie de concepts pour trouver la ou les meilleurs concepts pour une région. L'algorithme que nous proposons est basé sur des heuristiques afin de réduire l'espace de recherche. Le processus général de l'exploration est le suivant : si la région peut correspondre au concept courant, l'algorithme va descendre d'un niveau dans la hiérarchie définie par l'ordre partiel $\preceq_\Theta$. Si la région ne peut pas correspondre au concept courant, celui-ci est abandonné et ses sous-concepts ne seront pas explorés. Nous définissons le seuil *minScore* comme étant la valeur minimale du score de correspondance entre une région et une classe pour considérer possible que la région soit effectivement un objet de ce concept.

Fig. 5.7: *Image Quickbird fusionnée d'un quartier de Strasbourg (Zone 1), résolution 0,7m*

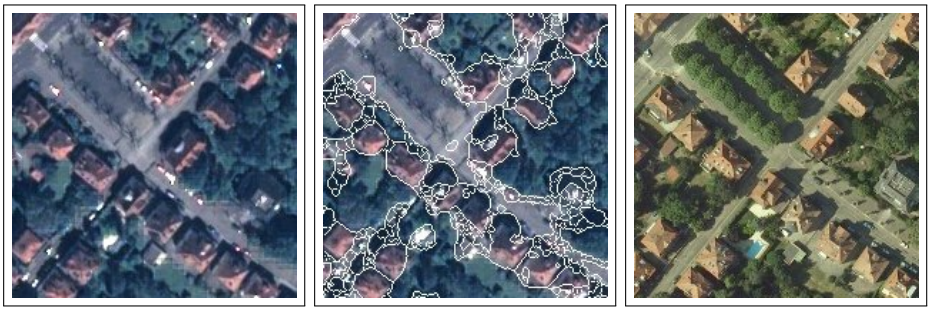
Soit $\mathscr{L} : \Theta \rightarrow \Theta$ telle que $\mathscr{L}(R)$ est l'ensemble des classes qui peuvent correspondre à la région R tenant compte du seuil *minScore*.

$$\mathscr{L}(R) \quad = \quad \{(\mathscr{C}_i, \texttt{Score}(R, \mathscr{C}_i)) \mid \texttt{Score}(R, \mathscr{C}_i) \geq minScore\} \tag{5.4}$$

L'exploration de la hiérarchie de concepts est présentée dans l'Algorithme 3. Ce processus peut être répété pour chaque région d'une image segmentée afin de donner une interprétation de l'image complète. Cependant, si un seuil *minScore* trop restrictif est donné en entrée à l'algorithme, il est probable que peu de régions soient identifiées.

Algorithme 3: Exploration de la hiérarchie de classe

Entrées : une région R, une hiérarchie de classes Θ, un seuil $minScore$
Sorties : $\mathcal{L}(R)$
$\mathcal{L}(R) = \emptyset$
$\mathcal{RC} = \{racine\}$
tant que $\mathcal{RC} \neq \emptyset$ **faire**
  $meilleursProfondeur = \emptyset$
  pour tous les $C_i \in \mathcal{RC}$ **faire**
    $s = \text{Score}(R, C_i)$
    si $s \geq minScore$ **alors**
      $\mathcal{L}(R) = \mathcal{L}(R) \cup \{(C_i, s)\}$
      $meilleursProfondeur = meilleursProfondeur \cup \{C_i\}$
  $\mathcal{RC} = \emptyset$
  pour tous les $C_j \in meilleursProfondeur$ **faire**
    $\mathcal{RC} = \mathcal{RC} \cup \{C_i | C_i \preceq_\Theta C_j\}$
renvoyer $\mathcal{L}(R)$



(a) Image brute (b) Image segmentée (c) Image aérienne

Fig. 5.8: *Exemple de segmentation, le bord des régions est affiché en blanc.*

5.2.3.2 Expériences

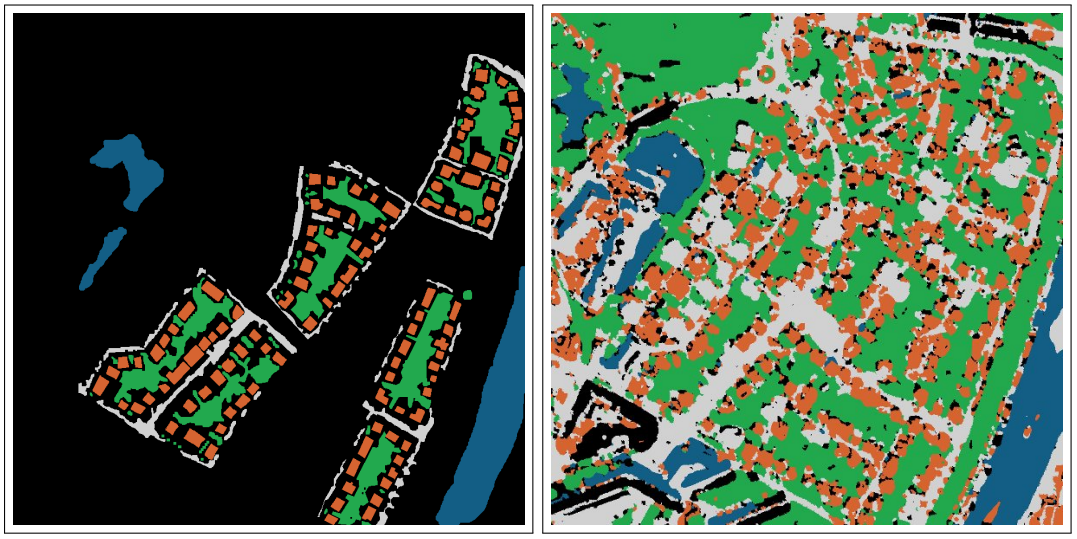
Pour illustrer le fonctionnement de l'approche, nous avons effectué des expériences sur des images de zones urbaines de la ville de Strasbourg. Nous avons utilisé pour cela des images provenant du satellite QUICKBIRD. Le satellite QUICKBIRD produit deux types d'images : des images panchromatiques (une seule bande spectrale) à 0,7m de résolution spatiale et des images multispectrales (4 bandes spectrales) à 2,8m de résolution. Ces deux types d'images ont été fusionnés [Puissant et al., 2003] pour obtenir une image à une résolution de 0,7m et quatre bandes spectrales. Une image d'un des extraits utilisés est présentée en figure 5.7, celle-ci sera appelée Zone 1. Elle représente un quartier résidentiel de la ville de Strasbourg et est représentative du type d'images classiquement traitées par un expert photo-interprète en milieu urbain.

Les zones étudiées sont principalement composées de routes (ou parkings), de végétation, d'eau et de pavillons. C'est pourquoi nous avons concentré notre étude sur l'identification des concepts Route, Végétation, Eau et Pavillon. Rappelons que la première étape du processus d'identification est la segmentation des images. Dans ces expériences nous avons utilisé une approche de segmentation développée par Derivaux et al. [2006]. Cette approche utilise une classification floue des pixels, puis applique l'algorithme de la ligne de partage des eaux [Vincent et Soille, 1991] dans l'espace des probabilités d'appartenance aux classes. La figure 5.8 montre la segmentation obtenue sur un petit extrait de l'image en figure 5.7. Nous verrons plus en détail la problématique de la segmentation dans la suite de ce chapitre.

Pour évaluer les résultats d'identification, nous avons utilisé des vérités terrain disponibles sur la zone. La figure 5.9 (a) présente la vérité terrain pour l'image de la figure 5.7. La méthode d'identification utilisant la base de connaissances a été utilisée pour identifier les régions issues de la segmentation. Ces résultats d'identification ont été comparés à la vérité terrain pour évaluer la qualité de cette identification. Cette évaluation a été effectuée au niveau pixel, c'est-à-dire en comparant l'affectation des pixels des objets identifiés avec les pixels de la vérité terrain disponible. Nous avons utilisé les indices de précision, rappel et f-mesure classiquement utilisés en recherche d'information (voir Annexe B). Le tableau 5.2 présente les résultats en fonction de différentes valeurs du seuil *minScore* allant de 0.75 à 1.

D'après ces résultats, nous pouvons constater que le paramètre *minScore* a une forte influence sur les résultats obtenus. En effet, avec un seuil de 1 seuls 66% des pixels du jeu d'évaluation sont reconnus comme appartenant à un concept, ce qui est relativement faible. Cependant, ce résultat peut s'expliquer par le fait qu'avec un seuil aussi haut, pour qu'une région soit reconnue comme appartenant à un concept, celle-ci doit avoir toutes ses valeurs d'attributs dans les bornes acceptées par le concept.

Pour relâcher cette contrainte, il est possible de baisser le paramètre *minScore*. En effet, plus ce paramètre va diminuer, plus l'algorithme va autoriser les régions à ne pas respecter toutes les contraintes définies dans le concept pour être considérées comme appartenant à celui-ci. Ceci se vérifie dans le tableau de résultat, où plus le seuil *minScore* diminue et plus le rappel augmente



(a) Vérité terrain. (b) Résultat de l'identification.

Fig. 5.9: *Résultat de la classification de l'image du quartier de Strasbourg.*

	minScore	Précision	Rappel	F-Mesure
Zone 1 (figure 5.7)	0.75	0.854	0.859	0.857
	0.80	0.854	0.859	0.857
	0.85	0.856	0.858	0.857
	0.90	0.862	0.854	0.858
	0.95	0.873	0.837	0.855
	1.00	0.883	0.662	0.757

Tab. 5.2: *Résultats obtenus sur l'évaluation de la vérité terrain de la Zone 1.*

ce qui montre une augmentation du nombre de régions reconnues. Cependant, relâcher certaines conditions a un prix, et quand le seuil *minScore* diminue, la précision de classification diminue également. En effet, en autorisant plus de régions à être identifiées on augmente le risque de faire des erreurs. Ainsi, le paramètre *minScore* permet à l'utilisateur de configurer le système en fonction du type de résultat qu'il souhaite obtenir, en trouvant un compromis entre exhaustivité de l'identification et absence d'erreur d'identification. La figure 5.9 (b) illustre le résultat de la classification de notre image de test avec un $minScore = 0,90$. Les régions non identifiées sont en noir.

Le tableau 5.3 présente les résultats en détail en fonction de chaque concept identifié. Ce tableau nous permet de mener une analyse de la qualité de l'identification en fonction des concepts.

- Route : le réseau routier est relativement bien reconnu. Le paramètre *minScore* a peu d'effet sur cette classe tant pour le rappel que pour la précision.
- Végétation : la classe végétation est bien identifiée, il n'y a que quelques régions de végétation ombrée qui ne sont pas reconnues.
- Pavillon : une partie importante des pavillons est détectée. Pour ce concept une valeur de *minScore* faible est préférable. En effet, quand le seuil *minScore* est élevé, le nombre de régions détectées de ce concept est très faible. Nous verrons par la suite les différentes raisons

concept \ indice	Précision					
	1.00	0.95	0.90	0.85	0.80	0.75
Pavillon	0.852	0.836	0.828	0.819	0.813	0.812
Végétation	0.984	0.983	0.982	0.982	0.982	0.982
Route	0.697	0.674	0.639	0.639	0.639	0.639
Eau	1.000	1.000	0.997	0.985	0.984	0.984

concept \ indice	Rappel					
	1.00	0.95	0.90	0.85	0.80	0.75
Pavillon	0.598	0.620	0.675	0.690	0.694	0.696
Végétation	0.969	0.973	0.976	0.976	0.976	0.976
Route	0.805	0.815	0.823	0.824	0.824	0.824
Eau	0.276	0.941	0.941	0.941	0.941	0.941

concept \ indice	F-Mesure					
	1.00	0.95	0.90	0.85	0.80	0.75
Pavillon	0.703	0.712	0.744	0.749	0.749	0.749
Végétation	0.977	0.978	0.979	0.979	0.979	0.979
Route	0.747	0.738	0.720	0.719	0.719	0.719
Eau	0.433	0.970	0.968	0.963	0.962	0.962

Tab. 5.3: *Résultats obtenus sur l'évaluation de la vérité terrain de la Zone 1 en fonction de chaque concept.*

expliquant ce phénomène.

– Eau : certaines grandes régions d'ombre ont été classées comme de l'eau. Il faut savoir que les réponses spectrales entre ces deux concepts ne sont pas discriminantes. Ce concept étant peu décrit par des indices de forme, ceux-ci ne peuvent servir à supprimer ces régions. À partir d'un seuil de $minScore = 0.95$, les erreurs sur les petites régions d'ombre classées par erreur comme de l'eau ne sont plus commises grâce à la connaissance de la taille minimale d'une surface d'eau.

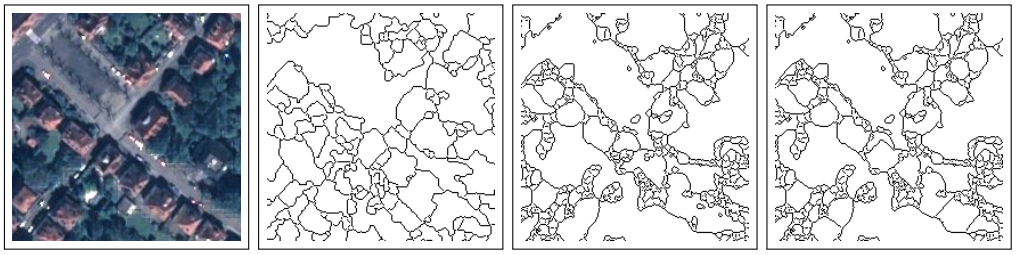
Comme nous pouvons le constater, il y a deux principales formes d'erreurs. La première est due au manque de précision des connaissances. Ce problème se manifeste par la difficulté de discrimination entre les classes Pavillon et Route : ces concepts semblent différentiables spectralement, l'algorithme proposé n'arrive cependant pas clairement à les séparer.

Le second type d'erreur provient de la qualité des régions construites lors de la segmentation. En effet, si la segmentation n'a pas correctement créé les régions, celles-ci auront des valeurs spectrales et de forme qui ne pourront pas coïncider avec les connaissances fournies par l'expert, et il sera alors difficile voire impossible de les identifier. Nous allons étudier dans la section suivante l'influence de la qualité de la segmentation sur les résultats d'identification.

5.2.3.3 Influence de la segmentation

Comme nous l'avons expliqué dans la section précédente, les résultats de l'approche sont très dépendants de la qualité de la segmentation. En effet, si celle-ci ne produit pas des régions de bonne qualité il sera difficile d'utiliser la connaissance de l'expert pour leur donner un sens. Prenons l'exemple des pavillons, l'expert a exprimé sa connaissance du fait que l'emprise au sol est en général de forme rectangulaire, ce qui a été traduit dans le concept pavillon par une contrainte de forme utilisant des indices géométriques. Si l'étape de segmentation ne produit pas des régions rectangulaires pour les pavillons, il sera impossible de les identifier.

Pour confirmer et illustrer ce phénomène nous avons mené un ensemble d'expériences en utilisant plusieurs segmentations de l'image 5.7. La figure 5.10 présente trois segmentations différentes. La



(a) Image brute (b) Ligne de partage des eaux (c) Segmentation supervisée (d) Segmentation supervisée avec modifications manuelles

Fig. 5.10: *Trois extraits de segmentations générées avec différents algorithmes.*

Segmentation	*minScore*	Précision	Rappel	F-Mesure
Ligne de partage des eaux	0.75	0.813	0.816	0.815
	0.80	0.813	0.815	0.814
	0.85	0.813	0.814	0.813
	0.90	0.823	0.758	0.789
	0.95	0.827	0.735	0.778
	1.00	0.789	0.649	0.712
Segmentation supervisée	0.75	0.838	0.842	0.840
	0.80	0.839	0.842	0.840
	0.85	0.841	0.840	0.841
	0.90	0.846	0.837	0.841
	0.95	0.856	0.820	0.837
	1.00	0.865	0.644	0.739
Segmentation supervisée avec modifications manuelles	0.75	0.854	0.859	0.857
	0.80	0.854	0.859	0.857
	0.85	0.856	0.858	0.857
	0.90	0.862	0.854	0.858
	0.95	0.873	0.837	0.855
	1.00	0.883	0.662	0.757

Tab. 5.4: *Résultats obtenus en fonction des trois segmentations.*

première est obtenue en utilisant l'algorithme classique de la ligne de partage des eaux, la seconde en appliquant la méthode basée sur une classification floue des pixels utilisée précédemment. Enfin, la troisième est construite à partir de la seconde segmentation, mais a été corrigée manuellement par un expert en coupant ou en fusionnant quelques régions.

Nous avons utilisé le mécanisme d'identification proposé avec ces trois segmentations en entrée. Les résultats sont présentés dans le tableau 5.4. Comme attendu, meilleure est la segmentation, meilleurs sont les résultats. La segmentation avec des corrections de l'expert offre les meilleurs scores d'identification. La méthode utilisant une classification floue des pixels vient en deuxième, et en dernier l'algorithme classique de la ligne de partage des eaux. Ces résultats montrent qu'il semble difficile d'obtenir une bonne segmentation sans supervision et qu'il est difficile de remplacer la connaissance de l'expert pour la création des régions [Derivaux, 2009].

5.2.3.4 Bilan et limites de l'approche

L'approche que nous avons présentée permet de formaliser les connaissances de l'expert, puis de les utiliser pour effectuer l'identification d'objets dans des images de télédétection. Comme nous l'avons vu dans les sections précédentes, les premiers résultats obtenus sont satisfaisants. Cependant, il convient de discuter plus en détail des limites de l'approche :

– **Précision de la caractérisation des concepts** : L'expert spécifie pour chaque concept des

espaces de définition sur un ou plusieurs attributs. Ceux-ci prennent la forme d'intervalles de valeurs acceptées pour ce concept. Il s'agit par conséquent de définir des hypercubes dans l'espace de données. Même définis au mieux, une grande partie de ces hypercubes peut ne pas correspondre à la classe définie à cause de la simplicité de la représentation. Cette simplicité est néanmoins nécessaire pour que l'expert puisse fournir ses connaissances dans un formalisme simple et facilement adaptable.

– **Qualité de la segmentation** : Si les valeurs des attributs des régions sont différentes des espaces de définition pour les concepts de ces régions, l'algorithme ne pourra identifier correctement ces régions. Cela peut se produire si la segmentation n'est pas de qualité suffisante.

Le paramètre *minScore* apporte une solution partielle à ces deux problèmes. En effet, ce seuil permet à l'algorithme d'être moins restrictif. Par conséquent, si les concepts ont été mal définis, par exemple sur un attribut (l'expert a pu mal évaluer la taille potentielle des pavillons), l'algorithme identifiera tout de même les régions. De plus, si la segmentation n'est pas parfaite, et que par conséquent les valeurs des régions ne correspondent pas parfaitement aux contraintes définies dans les concepts, un seuil abaissé permettra aux régions légèrement mal construites d'être tout de même identifiées.

Il est cependant à noter que le fait de baisser le seuil *minScore* implique des risques d'erreurs d'identification. En voulant pallier une mauvaise définition des concepts ou une mauvaise segmentation, on arrive rapidement à une situation où de nombreuses régions sont affectées à tort à des concepts. Il est donc, en pratique, assez difficile de choisir ce paramètre, et comme tout seuil, l'expert est souvent confronté à devoir faire de nombreuses expériences pour décider de la valeur qui convient à son application. Pour tenter de résoudre ces problèmes, nous allons voir dans les sections suivantes deux propositions. La première consiste à utiliser les connaissances dès l'étape de segmentation. La seconde consiste à utiliser la méthode de clustering collaboratif SAMARAH vue au Chapitre 2 ainsi que l'intégration de connaissances vue au Chapitre 3 pour effectuer de l'étiquetage de clusters de régions en vue d'une meilleure identification des objets dans l'image.

5.3 Intégration de connaissances pour la segmentation

5.3.1 Problématique liée à la segmentation d'image

5.3.1.1 La segmentation d'image

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant un critère pré-défini. Les pixels sont ainsi regroupés en régions et constituent une partition de l'image. Il existe de nombreuses méthodes de segmentation et des nouvelles sont proposées régulièrement. On pourra notamment noter trois grands groupes d'algorithmes. La segmentation par approche région, qui consiste à manipuler directement les régions. Les méthodes partent généralement d'un ensemble de régions puis les fusionnent itérativement en suivant l'évolution d'un critère. La segmentation par approche frontière cherche quant à elle à exploiter les transitions détectables entre les régions connexes. Enfin, les méthodes par seuillage définissent des règles permettant la séparation entre les différentes zones de l'image selon un ou plusieurs seuils.

Soit $\mathcal{I}$ l'image d'entrée et $\mathscr{X}$ un ensemble de régions, le processus de segmentation peut être représenté par la fonction s telle que :

$$s(\mathcal{I}) \mapsto \mathscr{X} \tag{5.5}$$

Dans le cadre des nos recherches, nous avons utilisé l'algorithme de la ligne de partage des eaux (LPE) [Vincent et Soille, 1991]. Cette méthode de segmentation est la méthode principale de la morphologie mathématique [Soille, 2003]. Elle considère l'image à traiter comme une surface topographique, en utilisant généralement le gradient de l'image. Une image de gradient contient en chaque pixel la dérivée des valeurs locales de l'image originale, de telle sorte qu'un contour dans

l'image originale correspondra à une forte valeur dans l'image de gradient. Une fois cette surface topographique définie, elle est inondée à partir de ses minima locaux générant ainsi des bassins de rétention qui s'étendent au fur et à mesure du processus d'inondation. Des barrages sont construits aux endroits de contact d'eau de bassins différents, ce qui créera les frontières entre les régions.

5.3.1.2 Évaluation de résultats de segmentation

L'évaluation d'une segmentation, à l'image de l'évaluation d'un clustering, reste un problème ouvert. Dans la littérature, de nombreux critères d'évaluation ont été proposés. Le lecteur pourra se référer aux travaux d'états de l'art [Cardoso et Corte-Real, 2005; Chabrier et al., 2006; Zhang, 1996] pour une étude détaillée. On distingue classiquement trois catégories de critères :

- **les critères analytiques** : ils se basent sur l'algorithme lui-même et étudient ses principes et ses propriétés. L'avantage de cette approche est qu'elle n'est pas dépendante des images sur lesquelles l'algorithme est évalué. Son inconvénient est qu'elle est uniquement qualitative ;
- **les critères empiriques non supervisés** : ils utilisent des mesures comme l'uniformité intra-région ou le contraste inter-région en s'appuyant sur des caractéristiques spectrales ou texturales. Ces critères ne sont pas adaptés aux images complexes car l'uniformité intrarégion par rapport aux caractéristiques des pixels n'est pas toujours pertinente car non liée à la sémantique, qui, dans le cas non-supervisé, ne peut pas être connue ;
- **les critères empiriques supervisés** : ces critères nécessitent l'intervention de l'utilisateur pour fournir des régions de référence afin d'évaluer le désaccord entre la segmentation obtenue et les régions fournies par l'utilisateur. Ces critères permettent de donner une idée précise des résultats qui peuvent être attendus par l'algorithme. Ils ont néanmoins une grande dépendance aux images de référence utilisées.

Il devient courant d'utiliser des critères hybrides qui prennent en compte plusieurs type d'information pour éviter le biais lié à l'utilisation d'un unique critère.

5.3.2 Utilisation de connaissances pour guider la segmentation

Partant du constat que la qualité de la segmentation a une forte importance quant à l'identification des régions, nous proposons d'utiliser les connaissances disponibles dans la base de connaissances dès l'étape de segmentation. En effet, dans la section précédente les connaissances étaient utilisées pour identifier les régions qui étaient supposées construites.

Pour influer sur la construction des régions au cours de l'étape de segmentation nous avons décidé de travailler sur les paramètres de l'algorithme de segmentation. La majorité des algorithmes de segmentation fonctionnent avec des paramètres ou seuils qui conditionnent le type de résultats obtenus (forme des régions, nombres de régions, etc.). L'idée est de les faire varier afin de trouver le jeu de paramètres qui permet d'effectuer une segmentation dont les régions sont le plus facilement identifiables grâce aux connaissances disponibles. Pour ce faire, nous allons utiliser un algorithme génétique dont l'objectif est d'optimiser la qualité de la segmentation, qui est évaluée en fonction des connaissances disponibles, c'est-à-dire la hiérarchie de concepts et la méthode d'identification associée. Pour évaluer la qualité d'une segmentation, la méthode d'identification vue précédemment est utilisée sur l'ensemble des régions produites par l'algorithme. Un indice est ensuite calculé pour évaluer la surface de l'image qui est reconnue par la méthode.

Soit $\mathscr{X}$ les régions d'une segmentation et $\mathscr{X}_o$ les régions identifiées par la méthode utilisant les connaissances du domaine ($\mathscr{X}_o \subseteq \mathscr{X}$). Le pourcentage de la surface de l'image reconnue par la méthode d'identification est défini par :

$$\texttt{reconnaissance}(\mathscr{X}, \mathscr{X}_o) = \frac{\sum_{R \in \mathscr{X}_o} \text{Aire}(R)}{\sum_{R \in \mathscr{X}} \text{Aire}(R)} \tag{5.6}$$

Nous avons utilisé la ligne de partage des eaux [Vincent et Soille, 1991] avec trois paramètres permettant d'influer sur les résultats obtenus. Ces paramètres prennent leurs valeurs dans $[0; 1]$.

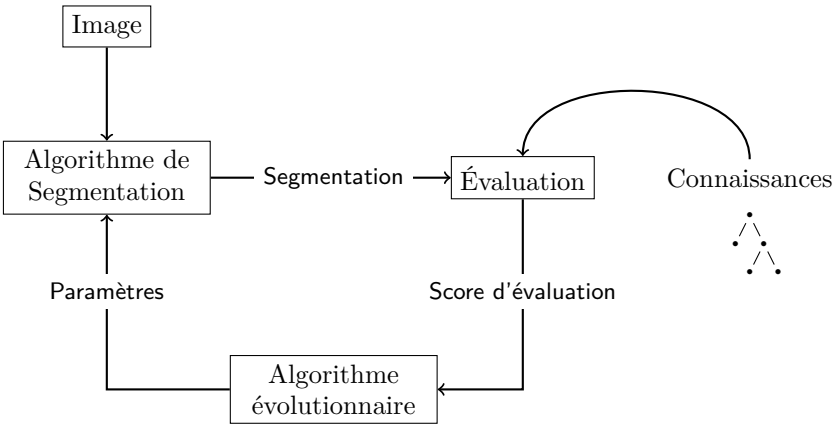


Fig. 5.11: *Processus de segmentation guidée par les connaissances.*

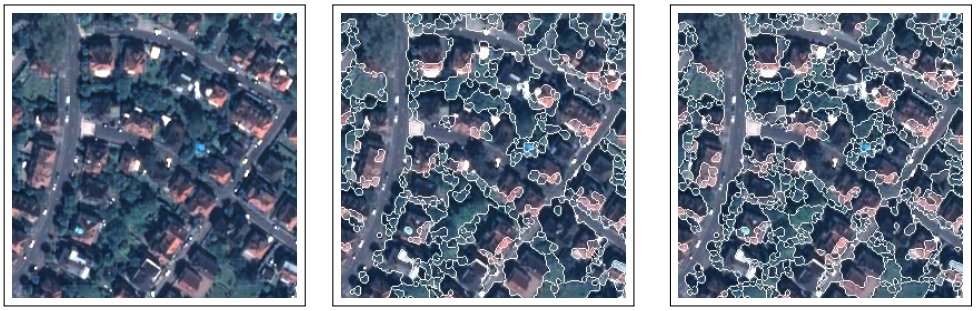
L'algorithme évolutionnaire recherche le triplet de valeurs réelles qui optimise l'évaluation de la segmentation. Chaque triplet de valeurs, qui correspond à des paramètres de segmentation, est évalué grâce à la surface de l'image reconnue en utilisant la segmentation issue de ces paramètres. La figure 5.11 illustre les différentes étapes du processus.

5.3.3 Évaluation

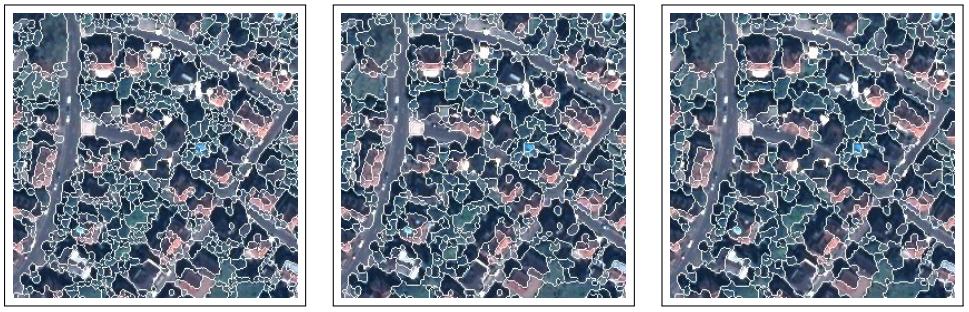
Pour évaluer la méthode de segmentation guidée par les connaissances que nous proposons, nous avons utilisé l'image du quartier de Strasbourg présentée en figure 5.7. La figure 5.12 montre un ensemble de vignettes représentant des extraits de segmentations obtenues au cours de l'exécution de l'algorithme évolutionnaire. On peut constater qu'au cours des générations, le pourcentage de l'image qui est reconnu augmente, et on observe une meilleure définition des frontières des régions. Ceci s'explique par le fait que l'algorithme a tendance à favoriser les paramètres qui produisent des segmentations dont les régions sont identifiables par la méthode de détection. Il est à noter que pour la méthode d'identification, le *minScore* a été paramétré à 1, c'est-à-dire que la méthode n'identifie que les régions qui correspondent parfaitement à la définition des concepts. Ce paramétrage a été nécessaire pour assurer une certaine qualité de l'identification. En effet, avec un seuil plus faible l'algorithme génétique n'aurait pas pu se baser sur cette valeur pour juger de manière précise de la qualité des individus.

Pour évaluer cette approche plus en détail, nous avons utilisé la vérité terrain disponible sur la zone, et nous avons évalué les segmentations obtenues au cours de l'exécution de l'algorithme évolutionnaire. Le tableau 5.5 présente les résultats de l'évaluation de la segmentation en utilisant les vérités terrains au cours des générations. Plusieurs critères ont été sélectionnés, la précision, le rappel, la f-mesure, ainsi que deux indices d'évaluation de segmentation supervisée Janssen et Feitosa [Janssen et Molenaar, 1995]. Tous ces indices sont à maximiser sauf Feitosa qui est à minimiser. Ces deux derniers critères consistent à observer la correspondance entre les régions produites par l'expert et les régions proposées par la segmentation. Les valeurs affichées correspondent aux moyennes calculées sur toutes les régions disponibles dans les vérités terrains (voir figure 5.9). Le but de cette analyse est de montrer que l'optimisation de notre critère conduit bien à une amélioration de la qualité de la segmentation. Les valeurs de rappel sont également données à titre indicatif avec le pourcentage d'identification sur les miniatures de la figure 5.12.

De plus, au cours d'une évolution, nous avons évalué chaque individu (c'est-à-dire chaque paramétrage de segmentation) en fonction de ces différents critères (précision, rappel, f-mesure, Janssen et Feitosa). Ainsi, 145 paramétrages possibles ont été évalués (tous les paramétrages testés par l'algorithme évolutionnaire). La figure 5.13 (a) présente les nuages de points avec l'évaluation sur notre critère de reconnaissance en abscisse et le rappel en ordonnée, la figure 5.13 (b) utilise l'indice de Janssen en ordonnée, et la figure 5.13 (c) l'indice de Feitosa. Nous pouvons constater que



(a) image brute (b) 37% reconnu, 29% de rappel ($1^{\text{ère}}$ génération) (c) 45% reconnu, 49% de rappel ($2^{\text{ème}}$ génération)



(d) 48% reconnu, 57% de rappel ($7^{\text{ème}}$ génération) (e) 51% reconnu, 57% de rappel ($12^{\text{ème}}$ génération) (f) 56% reconnu, 60% de rappel ($15^{\text{ème}}$ génération)

Fig. 5.12: *Extraits de segmentations obtenues à différentes générations au cours d'une évolution. Le contour des régions est affiché en blanc.*

Génération	Reconnaissance	Précision	Rappel	F-Mesure	Janssen	Feitosa
$1^{ère}$	37, 3%	0.760	0.294	0.424	0.490	0.856
$2^{ème}$	44, 5%	0.878	0.485	0.625	0.525	0.826
$7^{ème}$	47, 7%	0.881	0.566	0.689	0.529	0.800
$12^{ème}$	50, 8%	0.868	0.570	0.688	0.554	0.781
$15^{ème}$	55, 9%	0.872	0.596	0.708	0,543	0.816

Les valeurs sont exprimées en pourcentages à l'exception de Janssen et de Feitosa.

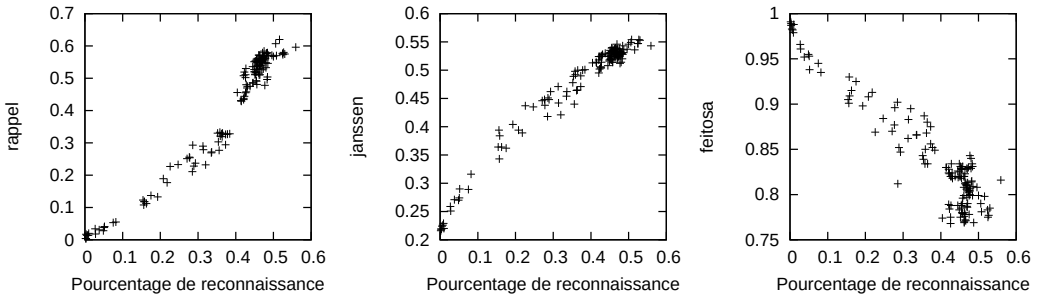
Tab. 5.5: *Résultat de l'évaluation de la méthode sur les régions d'exemples pour cinq générations.*

dans les trois cas, les nuages de points indiquent des corrélations entre l'évaluation de la reconnaissance et ces critères. Ces résultats montrent que l'optimisation de notre critère de reconnaissance conduit bien à une amélioration de la segmentation.

5.4 Connaissances et clustering collaboratif de régions

5.4.1 Problématique

Dans cette section, nous présentons comment nous avons adapté la méthode de clustering collaboratif vue au Chapitre 2 ainsi que l'intégration de connaissances vue au Chapitre 3 pour effectuer l'étiquetage de clusters de régions en vue d'une meilleure identification des objets dans l'image. Rappelons que la méthode d'identification vue en début de chapitre permet d'affecter un concept à des régions à l'issue d'une étape de segmentation. Cependant, la méthode ne permet d'identifier que peu de régions avec certitude. Le paramètre *minScore* qui permet d'augmenter le nombre d'objets reconnus en réduisant le respect des contraintes imposées par les concepts lève partiellement ce verrou. Cependant, quand ce seuil baisse, de nombreuses erreurs d'identification



(a) Rappel (b) Janssen (c) Feitosa

Fig. 5.13: *Évolution des fonctions d'évaluation pour* 145 *individus ordonnés par le taux de reconnaissance de la méthode de classification basée sur les connaissances du domaine.*

apparaissent. Nous proposons ici une autre approche consistant à utiliser la méthodes de clustering collaboratif sur les régions issues d'une segmentation. Chaque cluster se verra ensuite affecté une étiquette en fonction des objets lui appartenant. Cette approche a deux intérêts : le premier est de proposer un étiquetage automatique des clusters, et la seconde est de permettre une augmentation significative du nombre d'objets identifiés.

5.4.2 Étiquetage des clusters

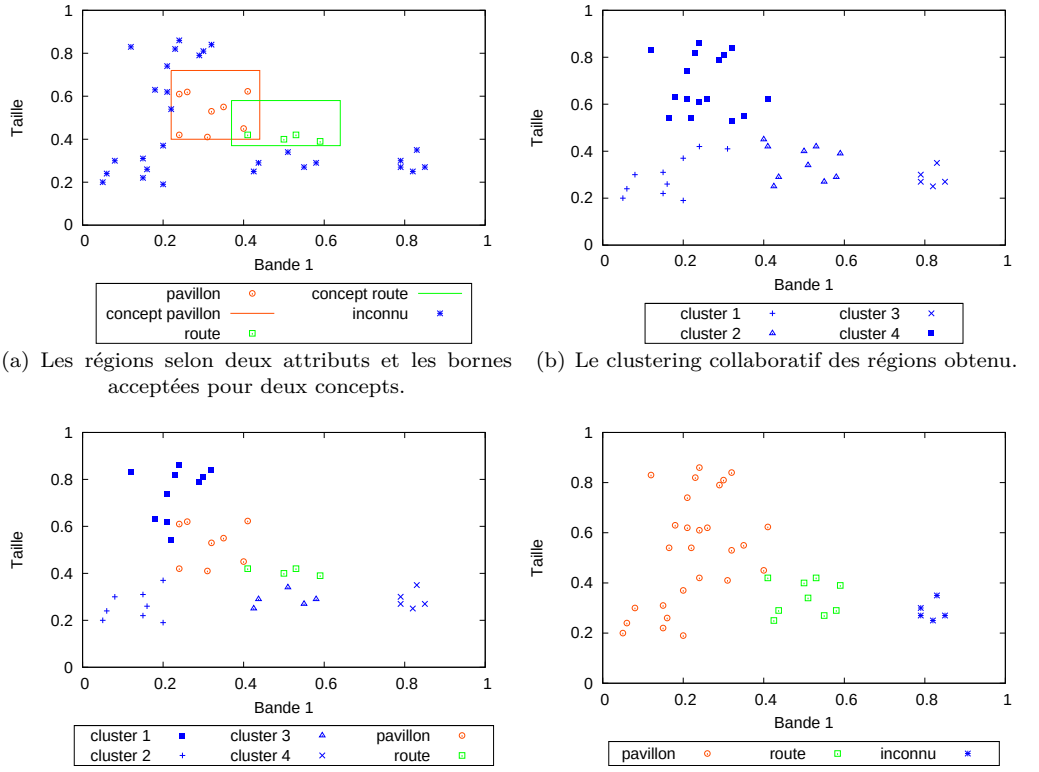
Le processus de clustering collaboratif (voir Chapitre 2) offre à l'expert des mécanismes efficaces de regroupement d'objets similaires sous la forme de clusters. Cependant, il ne permet pas d'affecter directement une sémantique aux clusters, c'est-à-dire une correspondance directe avec un concept ou une classe du monde réel. Pour pouvoir affecter cette sémantique, l'expert doit utiliser sa connaissance pour faire correspondre un groupe d'objets à un concept. Cette tâche, dite d'étiquetage des clusters, n'est pas une tâche facile pour l'expert. De plus, elle est généralement fastidieuse et peu gratifiante. La méthode d'étiquetage d'objets géographiques vues précédemment va nous permettre d'automatiser cette tâche. L'objectif est d'identifier pour chacun des clusters découverts par la méthode collaborative, si celui-ci correspond à un concept d'objets géographiques. Pour cela, les objets appartenant aux clusters, c'est-à-dire les régions, vont être utilisés en entrée de la méthode d'identification. Puis, dans chaque cluster, un vote à la majorité prenant en compte le score d'appariement (voir équation (5.3)) est effectué pour identifier le concept majoritaire.

La méthode appliquée est similaire à celle présentée au Chapitre 4 à la section 4.3.2. À l'issue de la segmentation, les régions sont caractérisées par des attributs spectraux et de forme. Le clustering collaboratif est alors appliqué sur ces régions caractérisées, qui sont ensuite identifiées en utilisant la base de connaissances.

Soit $\mathscr{X}$ l'ensemble des régions d'une segmentation et $\mathscr{X}_o$ l'ensemble des régions identifiées par la méthode utilisant les connaissances du domaine ($\mathscr{X}_o \subseteq \mathscr{X}$). L'objectif est d'utiliser ces régions identifiées pour étiqueter les clusters d'un résultat de clustering $\mathcal{C} = \{C_1, \ldots, C_K\}$. Le système d'étiquetage des clusters prend en compte le score d'appariement (voir équation (5.3)) affecté par la méthode d'identification aux régions en fonction du concept identifié. Le cluster est alors étiqueté par l'étiquette du concept dont la somme des scores est la plus importante parmi les régions présentes dans ce cluster :

$$\texttt{étiquetage}(C) = \arg\max_{\mathscr{C}} \sum_{R \in C} Score(R, \mathscr{C}) \quad | \quad \mathscr{C} \neq \texttt{inconnu} \tag{5.7}$$

À l'issue de cette étape d'étiquetage, chaque cluster C du résultat aura soit un concept $\mathscr{C}$ qui lui aura été affecté, soit le concept `inconnu` si aucune région du cluster n'a été identifiée. En effet, en fonction du seuil sélectionné, certaines régions peuvent rester inconnues. Il est conseillé de choisir



(a) Les régions selon deux attributs et les bornes acceptées pour deux concepts.

(b) Le clustering collaboratif des régions obtenu.

(c) La combinaison d'étiquetage et du clustering.

(d) Le résultat final avec l'étiquetage des clusters effectué.

Fig. 5.14: *Illustration du processus d'étiquetage des clusters dans l'espace des données.*

un seuil relativement élevé ($\geq 0,90$) pour éviter de trop nombreuses erreurs d'identification. Une étape de ré-étiquetage des régions est ensuite engagée. Lors de cette étape, dans chaque cluster ayant été étiqueté par un concept, les régions n'ayant pas encore de concept se voient affecter le concept assigné à leur cluster d'appartenance. Ce mécanisme permet d'augmenter considérablement le nombre de régions reconnues. Ainsi, de nombreuses régions vont passer du concept `inconnu` au concept associé à leur cluster lors de cette étape. Il est également envisageable de ré-étiqueter les régions présentant un concept différent que le concept majoritaire de leur cluster d'appartenance. Ce choix est laissé à l'utilisateur et ces régions (peu nombreuses en pratique) sont généralement traitées au cas par cas.

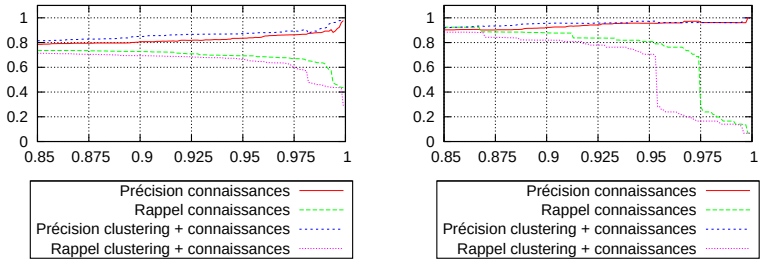
La figure 5.14 présente une illustration dans l'espace des données de ce mécanisme d'étiquetage des clusters. Chaque point sur ces figures représente une région, décrite ici par les attributs correspondant à la moyenne des valeurs sur la Bande 1 des pixels composant la région et à la taille de la région. La figure 5.14 (a) présente le résultat de l'identification utilisant les connaissances. Les bornes des concepts Pavillon et Route sont également illustrées. La figure 5.14 (b) présente le résultat de clustering obtenu en considérant les régions comme les objets à classer. La figure 5.14 (c) présente la combinaison de ces deux informations (clustering + connaissances). Enfin, la figure 5.14 (d) présente le résultat final, une fois la règle d'étiquetage des clusters (voir équation (5.7)) appliquée et les régions ré-rétiquetées avec le concept majoritaire de leur cluster.

5.4.3 Expériences

Des expériences ont été menées pour illustrer le fonctionnement de cette approche. La figure 5.15 présente deux zones d'étude (appelées Zone 2 et Zone 3) représentant deux quartiers de Strasbourg. Pour chacune de ces deux images, nous avons utilisé la base de connaissances en faisant varier le

(a) Zone 2. (b) Zone 3.

Fig. 5.15: *Images de quartiers de Strasbourg utilisées pour les expériences.*



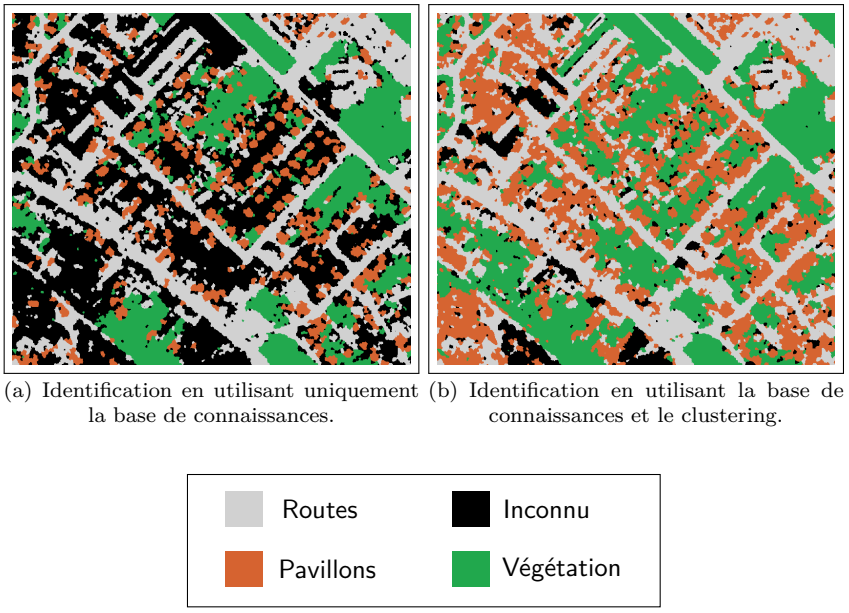
(a) Évolution de la précision et du rappel sur la Zone 2. (b) Évolution de la précision et du rappel sur la Zone 3.

Fig. 5.16: *Évolution de la précision et du rappel sur la Zone 2 et 3 pour la méthode utilisant uniquement les connaissances et la méthode utilisant les connaissances et le clustering.*

seuil *minScore* de $0,85$ à 1. Pour chaque seuil, les régions ont été appariées aux concepts et cette connaissance a été utilisée avec la méthode de clustering collaboratif pour étiqueter les clusters obtenus. Nous avons fait collaborer cinq instances de l'algorithme KMEANS pour rechercher entre 8 et 12 clusters. La connaissance de l'appartenance des régions aux concepts a également été utilisée dans la méthode pour construire des clusters les plus purs possibles en terme de concept (voir Chapitre 4).

Des vérités terrains ont été utilisées pour comparer les résultats obtenus avec les deux approches. La première en utilisant uniquement la méthode d'identification des régions. La seconde en utilisant la méthode de clustering collaboratif avec étiquetage des clusters comme connaissance. La figure 5.16 présente l'évolution de la précision et du rappel en fonction du seuil *minScore* pour les deux méthodes comparées (connaissances, clustering + connaissances). Comme on peut l'observer sur ces graphiques, la précision est sensiblement identique pour les deux approches. Cependant, le rappel est plus élevé pour la méthode utilisant le clustering avec les connaissances. Ceci s'explique par le fait que de nombreuses régions qui n'étaient pas reconnues par la méthode ont été étiquetées grâce à l'étape d'étiquetage des clusters.

La figure 5.17 et la figure 5.18 présentent un exemple de résultat obtenu pour un seuil *minScore* égal à 0.96. On peut voir sur ces images que la méthode utilisant les connaissances n'a identifié qu'un nombre limité de régions. Au contraire, la méthode utilisant les connaissances ainsi que la méthode collaborative présente un nombre de régions identifiées bien plus important.



(a) Identification en utilisant uniquement la base de connaissances.

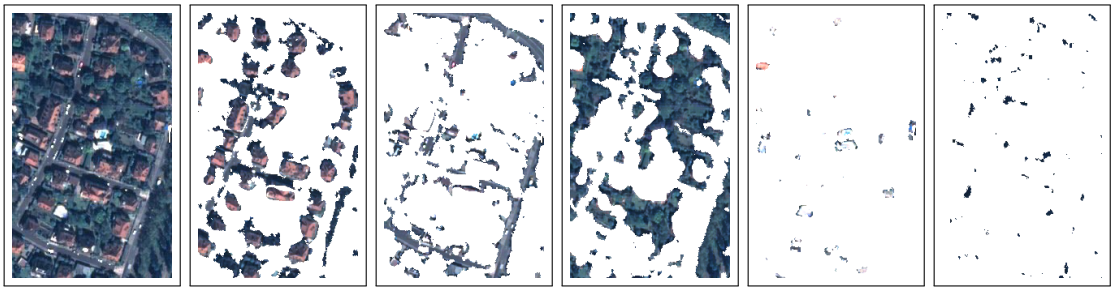
(b) Identification en utilisant la base de connaissances et le clustering.

Fig. 5.17: *Résultats de l'identification en utilisant uniquement les connaissances et les connaissances couplées au clustering (Zone 2).*

(a) Identification en utilisant uniquement la base de connaissances.

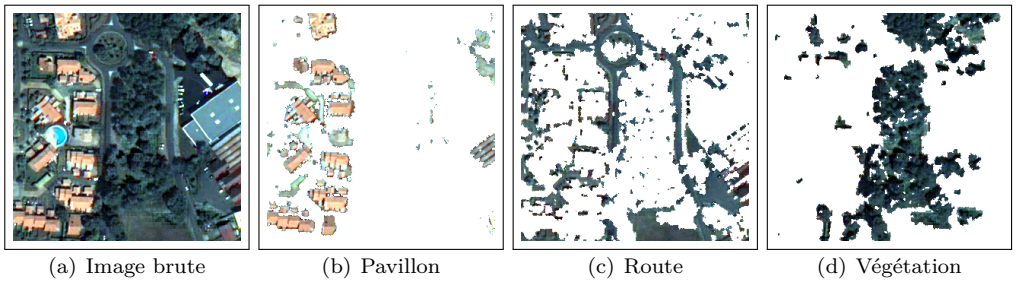
(b) Identification en utilisant la base de connaissances et le clustering.

Fig. 5.18: *Résultats de l'identification en utilisant uniquement les connaissances et les connaissances couplées au clustering (Zone 3).*



(a) Image brute (b) Pavillon (c) Route (d) Végétation (e) Inconnu 1 (f) Inconnu 2

Fig. 5.19: *Exemple de résultat obtenu avec la méthode utilisant les connaissances et le clustering sur un extrait de la Zone 1 de Strasbourg.*



(a) Image brute (b) Pavillon (c) Route (d) Végétation

(e) Inconnu 1 (f) Inconnu 2 (g) Inconnu 3

Fig. 5.20: *Exemple de résultat obtenu avec la méthode utilisant les connaissances et le clustering sur un extrait d'une image Quickbird de Bayonne.*

5.4.4 Découverte de concepts

Un intérêt important du mécanisme d'étiquetage de clusters est qu'il permet de ne pas étiqueter tous les clusters et permet ainsi de garder un degré de liberté pour la découverte de concepts qui ne sont pas encore présents dans la base de connaissances. En effet, la méthode d'identification utilisant la base de connaissances ne permet que d'identifier des régions appartenant aux concepts présents dans cette base. Une des fonctionnalités de la méthode d'étiquetage des clusters est de ne pas étiqueter les clusters dans lesquels aucune région n'a été identifiée. Ceci est par exemple visible sur l'exemple de la figure 5.14 (d) où il reste un cluster inconnu à la fin du processus. Ces clusters inconnus sont ensuite présentés à l'expert qui identifie si les objets regroupés correspondent à un concept. En effet, en fonction de la qualité de la segmentation, un cluster regroupant un ensemble d'objets mal segmentés a également pu être trouvé. Ce cluster n'aura donc pas de sémantique particulière.

Un exemple illustratif est proposé en figure 5.19 qui montre un résultat d'identification suivi de la méthode d'étiquetage de clusters. L'image utilisée est un extrait de la Zone 1 de Strasbourg (voir figure 5.7, page 115). Les régions identifiées comme appartenant aux concepts Pavillon, Route et Végétation sont présentées respectivement à la figure 5.19 (b), (c) et (d). Les figures 5.19 (e) et (f)

présentent deux clusters de régions dans lesquels aucune région n'a été étiquetée, il en résulte deux clusters inconnus. Après avoir présenté ces clusters à des experts, il a été possible d'identifier que le cluster présenté en figure (e) représente des sols clairs présents autour de piscines ou dans des cours. Ce concept n'étant pas présent dans la base il n'avait pas pu être identifié par la méthode utilisant les connaissances. Cependant, la méthode collaborative a regroupé ces régions avec succès dans un unique cluster. Le deuxième cluster inconnu, image (f), n'a pas de sémantique et correspond à des régions mal segmentées contenant de la route et de l'ombre.

Un deuxième exemple est présenté sur la figure 5.20 sur un extrait d'une image QUICKBIRD de la ville de Bayonne. Les régions identifiées comme appartenant aux concepts Pavillon, Route et Végétation sont présentées respectivement à la figure 5.20 (b), (c) et (d). Les figures 5.20 (e), (f) et (g) présentent trois clusters dont aucun objet n'a été identifié en utilisant les connaissances. On peut cependant identifier une piscine sur le cluster (b), ce cluster pouvant donc servir à ajouter ce concept dans la base. Le cluster (g) contient quant à lui un grand bâtiment industriel qui ne correspond pas à un concept. Ce cluster peut également être utilisé pour enrichir les connaissances. Enfin le cluster (e) contient majoritairement des régions mal segmentées comprenant des pixels hétérogènes (eau, ombre, route).

5.5 Bilan

Le domaine de la télédétection fourni de nombreuses données permettant d'effectuer une analyse de la surface terrestre. Dans ce chapitre, nous nous sommes intéressés à l'utilisation d'images issues de capteurs embarqués à bord de satellites ou aéroportés. Nous avons présenté un mécanisme permettant la construction d'une base de connaissances ainsi que son utilisation pour l'identification d'objets géographiques dans les images. Cette base de connaissances permet, à l'issue d'une segmentation, d'affecter un concept géographique aux différentes régions composant l'image. Pour lever les verrous concernant la dépendance de la méthode à la segmentation fournie en entrée du système, nous avons proposé deux solutions. La première consiste à utiliser les connaissances directement pendant l'étape de segmentation. Cette approche permet de construire des objets qui pourront être plus facilement reconnus par la base de connaissances. La seconde approche proposée, consiste à utiliser le clustering collaboratif pour créer des clusters de régions similaires. Ces clusters de régions similaires sont ensuite étiquetés en observant la répartition des concepts présents dans ces clusters. Loin de pouvoir fournir un système totalement automatique, la solution proposée permet d'aider l'expert dans le processus d'interprétation en lui fournissant des outils lui permettant de faciliter l'interprétation des images et la découverte de nouveaux concepts.

Contributions et valorisations

Les travaux effectués dans ce chapitre ont été validés par plusieurs publications. Un premier article [Durand et al., 2007] présenté à la conférence *IEEE International Conference on Tools with Artificial Intelligence* décrit le processus d'appariement ainsi que la mise en place de la base de connaissances. Les résultats obtenus par la suite concernant l'utilisation de la base de connaissances ont donné lieu à deux articles, l'un dans un workshop international [Forestier et al., 2008a] et l'autre dans une conférence nationale [Forestier et al., 2008b]. Enfin, la méthode d'étiquetage des clusters a été validée par une publication dans une conférence internationale de géosciences *IEEE Geoscience and Remote Sensing Symposium* [Forestier et al., 2008d]. Tous ces développements sont venus enrichir les plates-formes existantes *Mustic* et *Fodomust* développées au sein du laboratoire.

Chapitre 6

Utilisation de données multisources

Sommaire

6.1 Introduction

L'utilisation de données multisources consiste à prendre en compte différentes sources d'informations dans un processus commun de fouille de données. Les approches multisources sont de plus en plus utilisées pour traiter des problèmes où les données disponibles sont complexes et fortement hétérogènes. En effet, il est devenu courant de disposer de plusieurs sources de données proposant des informations différentes et potentiellement complémentaires. Cependant, il n'est pas toujours possible d'agréger toutes ces informations dans un unique jeu de données, et il n'est par conséquent pas toujours possible d'utiliser une approche classique monosource (c'est-à-dire traitant une seule source de données). De plus, transmettre une large quantité de données sur un site central peut également être impossible à réaliser. Enfin, les propriétaires des données peuvent vouloir partager les résultats mais pas leurs données ce qui amène à des problèmes de confidentialité [Agrawal et Srikant, 2000; Clifton, 2001; Verykios et al., 2004]. Dans ces approches, on cherche à combiner les résultats provenant de différentes sources de données en partageant des informations sur les résultats obtenus sans jamais partager les données.

Dans ce chapitre, nous allons présenter les travaux effectués lors de cette thèse sur l'utilisation de données multisources. Les solutions proposées sont fortement liées au domaine d'application de l'interprétation d'images de télédétection. En effet, il est difficile de développer des méthodes totalement génériques, car les données multisources sont souvent très spécifiques au domaine. Nous avons principalement étudié l'utilisation de deux types de données multisources différentes.

Le premier type consiste à utiliser des images de télédétection similaires à celles vues dans le chapitre précédent. En effet, l'expert possède souvent plusieurs images différentes de la même zone d'étude. Ces images peuvent être fortement hétérogènes et il existe de nombreux verrous quant à leur utilisation conjointe. Ces images peuvent par exemple provenir de satellites différents, avoir été prises à des dates différentes ou encore ne pas avoir la même résolution. Nous présenterons nos propositions pour lever ces différents verrous. Ces travaux ont principalement été menés lors d'un contrat de recherche et développement avec le CNES (*Centre Nationale d'Étude Spatiale*).

Le second type de données étudiées provient de librairies spectrales, qui sont des regroupements de spectres mesurés à l'aide d'un spectromètre ou de capteurs hyperspectraux. Ces spectres sont d'une précision très fine et offrent de nombreuses mesures le long du spectre électromagnétique. Nous verrons qu'il est possible de dégrader l'information contenue dans ces spectres, dans le but de simuler des données à la résolution de capteurs moins précis. Ces simulations permettent par la suite d'étudier les différents capteurs et comparer leurs performances. De plus, il est également possible d'étudier l'utilisation conjointe de plusieurs données simulées, pour évaluer l'intérêt potentiel d'utiliser des données provenant de capteurs différents. Ces travaux ont principalement été menés lors d'un séjour doctoral effectué au CNES au cours de cette thèse.

Nous allons présenter dans la suite de ce chapitre, comment nous avons utilisé et adapté la méthode de clustering collaboratif SAMARAH présentée au Chapitre 3 pour pouvoir traiter et tirer parti de ces données multiples.

6.2 Données multisources en classification d'images

6.2.1 Problématique

Dans le domaine de l'observation de la Terre, la majorité des méthodes d'analyse multisources part du principe qu'il existe un objet *réel* O (ici la scène à analyser) qui peut être vu suivant différents points de vue, le but étant de trouver une description de cet objet prenant en compte ces différents points de vue. Chaque vue $V^{(i)}$ est donnée par une image $\mathcal{I}^{(i)}$ et est composée d'un ensemble de pixels décrits par des attributs radiométriques.

Plusieurs situations sont alors envisagées (voir la figure 6.1) :

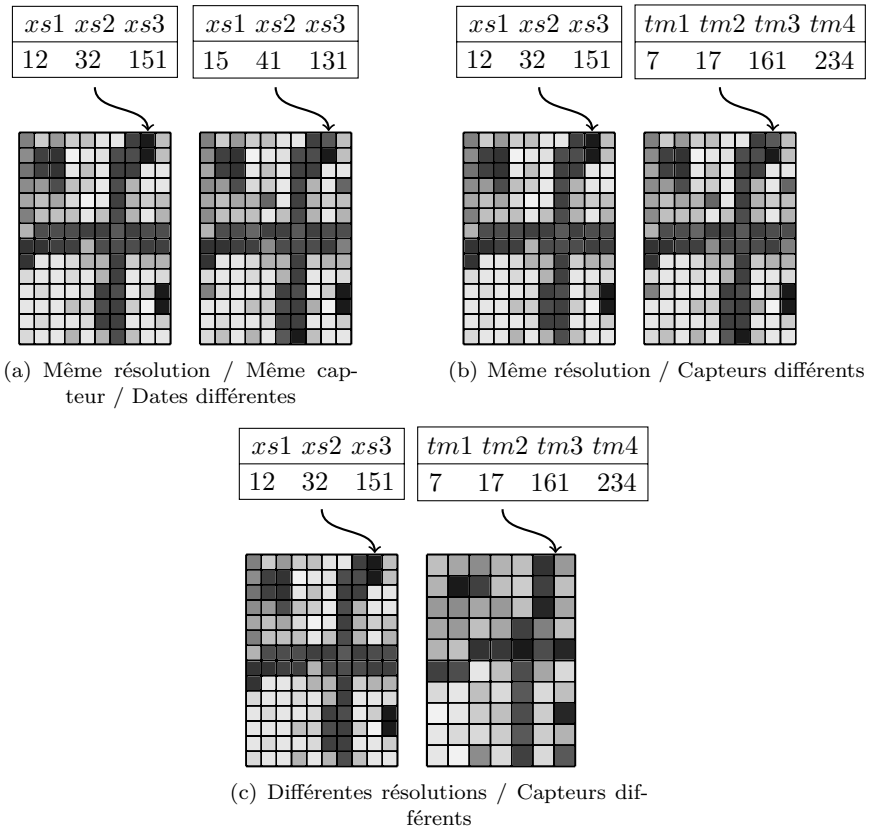
(a) Toutes les images $\mathcal{I}^{(i)}$ contiennent les mêmes objets et ceux-ci sont décrits par les mêmes attributs (radiométriques pour l'analyse par pixels, caractéristiques pour l'analyse basée région) avec des valeurs différentes dans les différentes images : exemple, deux images de la même zone, du même satellite mais à des dates différentes ;

(b) Toutes les images $\mathcal{I}^{(i)}$ contiennent les mêmes objets et ceux-ci sont décrits par des attributs différents : exemple, deux images de la même zone, à la même résolution mais provenant de capteurs (ou satellites) différents ;

(c) Les images $\mathcal{I}^{(i)}$ ne contiennent pas les mêmes objets et ceux-ci sont décrits par des attributs différents : exemple, deux images de la même zone, à des résolutions différentes provenant de capteurs (ou satellites) différents ou deux segmentations différentes d'une même image ;

Une première technique pour traiter ces images consiste simplement à construire une vue unique par *fusion des données* (c'est-à-dire des différentes images). Ainsi, certains satellites produisent simultanément lors d'une prise de vue, plusieurs images de la même zone : une image panchromatique et une image multispectrale. L'image panchromatique a une bonne résolution spatiale, mais une faible résolution spectrale alors que l'image multispectrale a une bonne résolution spectrale, mais une faible résolution spatiale. Une solution pour utiliser ces deux sources d'information consiste à fusionner l'image panchromatique et l'image multispectrale (*pan-sharpening*). De nombreuses méthodes ont été étudiées au cours des dernières années pour fusionner ces deux types d'images et produire une image de bonne qualité spatiale et spectrale [Dou et al., 2007; Ioannidou, 2007].

Une autre méthode est proposée par Chibani [2005] pour la combinaison d'images multispectrales, panchromatiques et radar. Cette méthode utilise conjointement la transformation Intensité Teinte Saturation et la décomposition en ondelettes. Dans [Chang et al., 2007], une méthode ap-



(a) Même résolution / Même capteur / Dates différentes

(b) Même résolution / Capteurs différents

(c) Différentes résolutions / Capteurs différents

Fig. 6.1: *Trois cas d'images multisources en télédétection.*

pelée GPBF (*Generalized Positive Boolean Function*) est proposée. Celle-ci se base sur la fusion d'images de télédétection en créant des bandes supplémentaires puis en utilisant un classifieur sur ces nouvelles données.

Une approche similaire à la fusion de données, dite *fusion de caractéristiques*, consiste à créer de nouveaux objets à partir des différentes données, puis à les fusionner. Par exemple, des segmentations peuvent être effectuées sur les images puis être fusionnées [Dubuisson et Jain, 1995; Kropatsch, 1990]. Dans [Germain et al., 2004], les auteurs présentent une méthode basée sur la théorie des croyances proposée par Dempster-Shafer [Shafer, 1978]. Ils proposent une approche statistique floue appelée FSEM (*Fuzzy Statistic Estimation Maximization*) pour l'initialisation de la fonction de masse.

Néanmoins, deux limitations s'opposent à une utilisation efficace de ces types de fusion :

- il n'existe pas de méthode de fusion pour tous les types d'images ;
- la résolution de l'image issue de la fusion peut être trop élevée car elle doit être au moins égale au plus petit commun multiple des résolutions initiales ;
- le nombre de bandes ou d'attributs peut exploser au risque de se heurter à la *malédiction de la dimensionnalité* [Bellman, 1961] : la majorité des méthodes basées sur des calculs de distance ne sont plus efficaces devant la (sur)abondance de bandes ou d'attributs car les distances entre les objets ne présentent plus de différences significatives. De plus, l'augmentation de la dimensionnalité spectrale des images, nous confronte aux problèmes liés au phénomène de Hughes [Hughes, 1968] se caractérisant par la difficulté de réaliser de bonnes classifications dans un espace pratiquement vide d'exemples.

Enfin, comme discuté dans [Pohl et Genderen, 1998], cette analyse à sources multiples peut aussi se faire au niveau des décisions où les résultats de l'analyse de chacune des images sont fusionnées en un résultat unique.

La *fusion de décisions* se fait conformément à la décision prise par chacune des méthodes d'extraction de l'information ou de classification. Dans Benediktsson et Kanellopoulos [1999], les auteurs proposent une méthode de fusion de décisions fondée sur la combinaison de schémas statistiques avec les réseaux neuronaux pour la classification des sources multiples. Dans le même esprit, le système exposé dans Bruzzone et al. [2002] est composé d'un ensemble de classifieurs qui est entraîné en deux étapes, d'abord supervisé puis sans supervision.

Dans le cadre de la classification non supervisée, les recherches se sont principalement focalisées sur la fusion au niveau de décisions, et sur la comparaison et la recherche de consensus de résultats de clustering. De nombreuses propositions ont été faites ces dernières années et ont prouvé leur efficacité dans bien des domaines (voir Chapitre 3).

Dans les méthodes basées sur un mécanisme de vote, on s'intéresse à trouver pour *chaque objet* le cluster auquel il appartient dans le résultat unifiant en fonction des clusters auxquels il appartient dans les différents résultats. Or, avec des résolutions différentes se pose le problème de correspondance inter-images : comment *retrouver* un pixel dans une image à une résolution différente ? Malheureusement, cela nécessite bien souvent que le nombre de clusters soit le même dans les différents résultats mais aussi dans le résultat unifiant (éventuellement, un cluster pour les objets non classés est ajouté). Ces mécanismes de vote sont principalement utilisés lorsque les clusters à trouver ne sont pas fortement séparables dans l'espace des données.

Ces contraintes rendent difficile la définition d'une approche multisource car des algorithmes différents produisent en général des résultats peu comparables. Par contre, il est possible d'utiliser des informations sur les objets eux-mêmes et sur la façon dont ils ont été classés.

Contrairement aux approches existantes, nous proposons d'utiliser un processus qui ne se base pas la fusion directe des résultats mais sur la collaboration entre ceux-ci. L'idée est d'utiliser le processus collaboratif présenté au Chapitre 3 de manière multisource, c'est-à-dire en affectant une image différente à chaque algorithme de clustering. Nous allons étudier dans la section suivante les verrous posés par l'utilisation de données multisources. Dans ce cadre, nous allons également présenter deux approches développées lors de cette thèse et permettant de traiter des images multisources.

6.2.2 Vers une approche de clustering collaboratif multisource

Le système de clustering collaboratif SAMARAH présenté en détail dans la seconde partie de cette thèse peut utiliser plusieurs méthodes de clustering simultanément mais sur un même jeu de données (ici, une image) pour toutes les méthodes. L'objectif de notre démarche est d'intégrer un mécanisme permettant d'utiliser plusieurs images simultanément dans ce processus. La figure 6.2 présente une illustration de la différence entre une approche de clustering collaboratif monosource et multisource. Cette modification implique de nombreuses adaptations de la méthode, notamment au niveau de la comparaison des résultats.

Pour intégrer cette nouvelle approche dans notre méthode nous nous sommes heurtés à plusieurs verrous scientifiques. Le premier verrou, et sûrement le plus important, réside dans la différence de niveau de sémantique associé à chacune des différentes résolutions des images (ou groupes de résolutions). En effet, s'il existe dans le domaine de l'analyse de scènes urbaines, un grand nombre de termes pour désigner/qualifier la couverture ou l'occupation des sols, une part importante de ces termes correspondent à des objets dont la visibilité dépend de la résolution de l'image. Ainsi, il existe un large éventail de nomenclatures d'objets associés aux images de télédétection telles que la nomenclature CORINE LAND COVER définie pour les images LANDSAT à 30m, la nomenclature SPOT définie pour les images SPOT de 5m à 20m ou encore la base de données BDCARTO (© IGN) définie pour les photos aériennes et les images SPOT.

Ces nomenclatures sont bien adaptées à l'analyse d'images à résolution moyenne de l'ordre de la dizaine de mètres et correspondent à des analyses entre le $1/100.000^e$ et le $1/50.000^e$. Par exemple, la nomenclature CORINE LAND COVER permet de caractériser des cartes à une échelle allant du $1/100.000^e$ au $1/50.000^e$. Cette nomenclature propose 3 niveaux de sémantique avec 44 classes au niveau 3, 15 au niveau 2 et 5 au premier niveau.

1 :25,000 Level 4 : Niveau des zones	**1 :10,000** Level 5 : Niveau des blocs	**1 :5,000** Level 6 : Niveau des objets urbains
• Tissu urbain de forte densité • Tissu ubrain de faible densité • Zones industrielles • Zones forestières • Zones agricoles • Surface d'eau • Sol nu	• Blocs urbains continus • Bloc urbains discontinus - Blocs urbains individuels - Blocs urbains collectifs • Blocs urbains industriels • Vegetation urbaine • Forêt • Zones agricoles • Surface d'eau • Route	• Batiments/toits : toit résidentiel gris clair, toit de tuile rouge, etc. • Végétation : végétation verte, végétation non photosynthétique. • Transport : rue, parking, etc. • Surfaces d'eau : rivière, étendues d'eau naturelles. • Sol nu • Ombre

Tab. 6.1: *Extrait de la nomenclature définie pour une analyse des images HR et THR.*

Or, de nos jours, avec les images à très haute résolution, il est possible d'extraire les objets urbains eux-même : bâtiments, routes, jardins, etc. Avec ces images, il est possible d'analyser des objets via leurs matériaux (par exemple, les maisons aux toits oranges) ce qui correspond à une analyse au $1/5.000^e$ (voir tableau 6.1 colonne de droite).

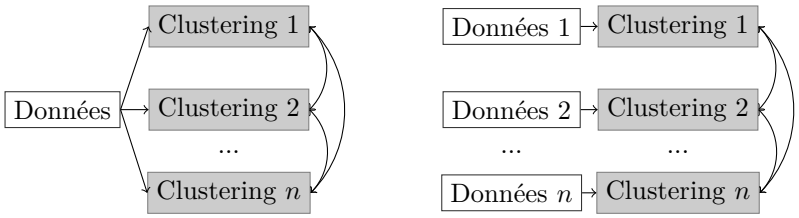
Enfin, dans le domaine de l'analyse urbaine, de plus en plus d'experts ont besoin d'analyser les images en termes de blocs urbains (ou d'îlots). Un tel bloc urbain est souvent défini comme une zone urbaine délimitée par une cycle minimal de moyen de communication. Ce niveau d'analyse correspond à une analyse au $1/10.000^e$ (voir tableau 6.1 colonne centrale). Or, dans ce cas, il n'existe pas de résolution d'images naturelle. En effet, les images HR présentent une résolution trop basse alors que les images THR nécessitent de reconstruire les blocs urbains. Il existe donc un niveau intermédiaire dans les niveaux d'analyse qui ne correspond pas directement à un intervalle de résolutions d'images. Ainsi, le choix du niveau sémantique d'analyse impose, dans une forte mesure, le choix de la résolution des images à utiliser.

Le deuxième verrou scientifique que nous avons tenté de lever, est lui aussi lié à cette différence de niveau sémantique. En effet, en fonction du niveau sémantique associé à l'image à traiter, le nombre de clusters à mettre en évidence peut être différent pour une même scène analysée. Ainsi par exemple, une seule classe de bâti peut être détectée sur des images à 20m alors que la nomenclature propose 4 classes de bâtiments pour les images à 2,5m.

On voit ici, qu'une fusion d'images à des résolutions correspondant à des niveaux sémantiques différents, va premièrement nécessiter de choisir un seul niveau d'analyse et deuxièmement obliger à un choix du nombre de clusters fixe. Dans nos travaux, nous avonc donc cherché à nous libérer de ces deux contraintes.

Une *première phase* dans nos recherches a consisté à étudier et intégrer un mécanisme de clustering multisource capable de traiter des images de résolutions différentes les unes des autres, mais pour lesquelles l'objectif est d'obtenir un même nombre de clusters dans chacun des résultats. Chaque méthode de clustering travaille sur une image de la même zone à différentes résolutions (éventuellement simulées). La correspondance entre les objets (c'est-à-dire les pixels) est réalisée grâce au géoréférencement des images. Celui-ci permet de mettre en correspondance les pixels d'une image par rapport à leurs coordonnées *réelles* sur le globe terrestre. Dans cette première phase, les étapes du processus de raffinement n'ont pas été modifiées car les méthodes cherchent toujours à obtenir le même nombre de clusters. Seul le résultat unifiant est calculé avec la résolution la plus fine des images d'entrée. Cependant, l'amélioration obtenue grâce à l'utilisation de différentes sources s'effectue autant sur les résultats finaux des différentes méthodes que sur le résultat unifiant. Néanmoins, en fonction de l'hétérogénéité des images d'entrée, le résultat unifiant peut perdre du sens (la différence de résolution étant trop importante entre les différentes images). Les développements effectués lors de cette phase sont décrits dans la section 6.2.3.

Une *deuxième phase* dans nos recherches a consisté à étudier les modifications à apporter à ce mécanisme pour permettre à chacune des méthodes de rechercher un nombre de clusters



(a) Monosource (b) Multisource

Fig. 6.2: *Illustration des deux paradigmes de clustering collaboratif monosource (a) et multisource (b)*

pouvant être différent. Cette réflexion a débouché sur la définition d'une méthode de classification multisource pouvant traiter des images ayant une forte différence de résolution. Dans cette méthode, un algorithme de clustering est tout d'abord appliqué sur chacune des deux images. Puis, des régions sont construites indépendamment dans chacune des images en considérant les pixels connexes appartenant au même cluster. Ces régions sont ensuite décrites par des histogrammes, construits en observant la distribution des clusters affectés aux pixels correspondants aux pixels de la région dans l'autre image. Cette correspondance est effectuée grâce au géoréférencement. Enfin, un algorithme de clustering est utilisé avec en entrée les régions décrites par les histogrammes. Les développements effectués lors de cette phase sont présentés dans la section 6.2.4.

6.2.3 Clustering basé pixel à un seul niveau de sémantique

Nous décrivons dans cette section la première phase qui a consisté à étudier et intégrer un mécanisme de clustering collaboratif multisource capable de traiter des images de résolutions différentes, l'objectif étant d'obtenir un même nombre de clusters dans chacun des résultats. Cette approche peut être envisagée quand les résolutions des différentes images utilisées en entrée du système n'ont pas une différence de résolution trop importante, et que l'on recherche un nombre de clusters sensiblement similaire. On peut considérer qu'il est réaliste d'adopter une telle approche tant que $r^{(2)} \leq (2 * r^{(1)})$ et $r^{(1)} \leq r^{(2)}$ avec $r^{(1)}$ et $r^{(2)}$ les résolutions des deux images $\mathcal{I}^{(1)}$ et $\mathcal{I}^{(2)}$.

6.2.3.1 Problématique de l'extension multisource

Le problème principal de l'intégration du mécanisme multisource dans la méthode collaborative est l'absence d'un espace de données commun, qui est nécessaire lors de la comparaison des différents résultats de classification. En effet, la méthode collaborative utilise la notion de matrice de confusion (voir équation 3.6), qui permet d'observer la répartition des objets dans les clusters d'un couple de résultats. Cette matrice de confusion permet d'évaluer la similarité entre deux résultats ainsi que d'y détecter des conflits éventuels.

L'enjeu est de réussir à construire cette matrice de confusion lorsque les deux résultats ne partagent pas un espace de données commun, c'est-à-dire dans notre cas, qu'un des deux résultats possède plus d'objets que le deuxième résultat impliqué dans la comparaison (c'est-à-dire des images à différentes résolutions).

6.2.3.2 Approche mise en oeuvre

Il a été nécessaire de trouver un moyen de comparer deux résultats n'ayant pas le même nombre d'objets. Pour se faire, l'approche la plus intuitive est de trouver une fonction de correspondance λ entre les objets d'un résultat et ceux d'un autre résultat, capable de faire le lien entre les deux espaces de données. Cette correspondance est triviale lors de l'utilisation d'une même représentation des données. Dans le cadre des images de télédétection, il est apparu naturel d'utiliser le

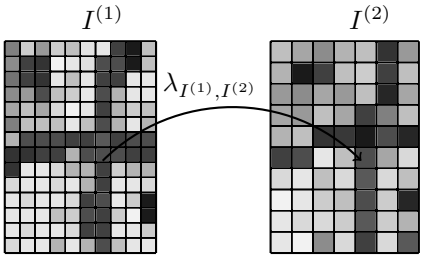


Fig. 6.3: *Exemple de fonction d'association $\lambda_{I^{(1)},I^{(2)}}$ entre deux images ($r^{(1)} \leqslant r^{(2)}$).*

géoréférencement permettant de mettre en correspondance les différents pixels à différentes résolutions, en calculant leurs coordonnées réelles.

Dans la définition initiale de SAMARAH, chaque ligne de la matrice de confusion est donnée par un vecteur de confusion $\alpha_k^{(i,j)}$ du cluster $C_k^{(i)}$ du résultat $\mathcal{C}^{(i)}$ comparé aux $K^{(j)}$ clusters trouvés dans le résultat $\mathcal{C}^{(j)}$:

$$\alpha_k^{(i,j)} = (\alpha_{k,l}^{(i,j)})_{l=1,\ldots,n_j} \ , \ ou \ \ \alpha_{k,l}^{(i,j)} = \frac{|C_k^{(i)} \cap C_l^{(j)}|}{|C_k^{(i)}|} \tag{6.1}$$

Si les deux résultats ne partagent pas le même espace de données, il n'est pas possible de calculer $|C_k^{(i)} \cap C_l^{(j)}|$. C'est pourquoi nous avons proposé une nouvelle définition du vecteur de confusion pour un cluster $C_k^{(i)}$ d'un résultat $\mathcal{C}^{(i)}$ comparé au résultat $\mathcal{C}^{(j)}$.

Matrice de confusion géoréférencée :

La *matrice de confusion géoréférencée* $\alpha_{(geo)}$ entre les clusters $C_k^{(i)}$ du résultat $\mathcal{C}^{(i)}$ et les $K^{(j)}$ clusters trouvés dans le résultat $\mathcal{C}^{(j)}$ est définie par :

$$\alpha_{(geo)k,l}^{(i,j)} = \frac{\#(C_k^{(i)}, \mathcal{I}^{(i)}, \mathcal{I}^{(j)})}{|C_k^{(i)}|} \quad \text{si} \quad r^{(i)} \leqslant r^{(j)}$$

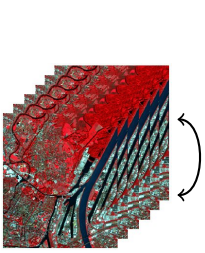
$$\alpha_{(geo)k,l}^{(i,j)} = \frac{\#(C_l^{(j)}, \mathcal{I}^{(j)}, \mathcal{I}^{(i)})}{|C_k^{(i)}|} \times \frac{r^{(j)}}{r^{(i)}} \quad \text{sinon}$$

avec :

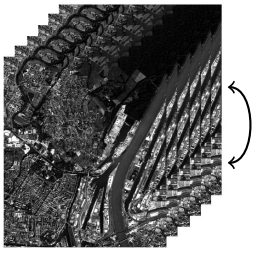
- $r^{(i)}$ et $r^{(j)}$ les résolutions des deux images $\mathcal{I}^{(i)}$ et $\mathcal{I}^{(j)}$
- $\lambda_{\mathcal{I}^{(i)},\mathcal{I}^{(j)}}$ la fonction associant un pixel de l'image $\mathcal{I}^{(i)}$ à un pixel de l'image $\mathcal{I}^{(j)}$, avec $r^{(i)} \leqslant r^{(j)}$
- $\#(\mathcal{C}, \mathcal{I}^{(i)}, \mathcal{I}^{(j)}) = |\{p \in \mathcal{C} : cluster(\lambda_{\mathcal{I}^{(i)},\mathcal{I}^{(j)}}(p)) = \mathcal{C}\}|$.

Avec cette nouvelle définition du vecteur de confusion, les résultats peuvent être comparés, même s'ils ne sont pas construits à partir d'images de même résolution. De fait, l'évaluation de la similarité entre résultats ainsi que la phase de recherche et de résolution des conflits entre résultats restent inchangées, tout en étant basées sur le calcul de la matrice de confusion entre deux résultats. La méthode SAMARAH a ainsi pu être directement utilisée avec cette nouvelle définition.

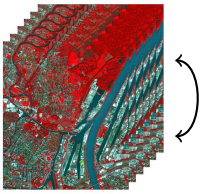
Cependant, comme les images n'ont pas la même résolution, il n'est pas possible d'utiliser la procédure d'unification des résultats, afin de ne proposer qu'un résultat représentant l'ensemble des résultats proposés. Si l'on souhaite construire un résultat unique, il est nécessaire de choisir une résolution et de la fixer. Ensuite, grâce à la fonction d'association λ, l'algorithme de vote décrit dans la méthode SAMARAH peut être appliqué (voir section 3.3.2.3), en plongeant chacun des résultats dans la résolution choisie. Le choix de la résolution se fait naturellement sur la résolution la plus haute présente dans l'ensemble de données.



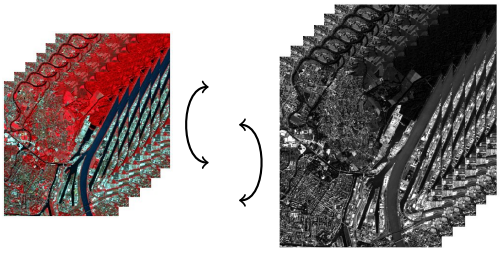
(a) Multispectrale : indépendamment sur l'image multispectrale.



(b) Panchromatique : indépendamment sur l'image panchromatique.



(c) Fusion : sur la fusion de l'image panchromatique et de l'image multispectrale



(d) Multisource : utilisant conjointement les images panchromatique et multispectrale

Fig. 6.4: *Les quatre cas de test de classification non supervisée collaborative.*

6.2.3.3 Expérimentations

Des expériences ont été effectuées sur des images SPOT5 (© CNES) de la ville de Strasbourg. Dans ces tests, différentes méthodes utilisent soit une image multispectrale soit une image panchromatique. Différentes configurations de SAMARAH ont été évaluées :

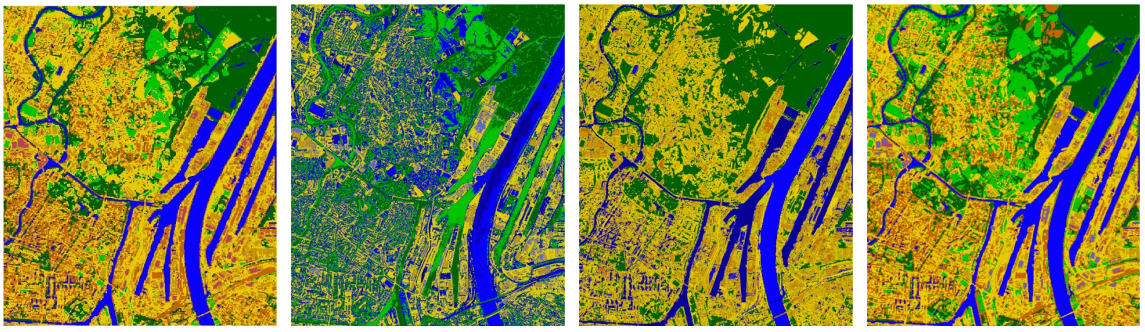
a) six méthodes de classification non supervisée utilisant l'image multispectrale ;
b) six méthodes de classification non supervisée utilisant l'image panchromatique ;
c) six méthodes de classification non supervisée utilisant la fusion des deux images ;
d) trois méthodes de classification non supervisée utilisant l'image panchromatique et trois méthodes de classification non supervisée utilisant l'image multispectrale.

L'évaluation des résultats a été effectuée sur l'unification des résultats à l'issue du processus de raffinement modifié. Les résultats ont montré que la collaboration entre l'image panchromatique et l'image multispectrale tendait à donner des résultats plus intéressants (voir figure 6.5).

6.2.4 Clustering basé régions à plusieurs niveaux de sémantique

Dans cette section, nous présentons notre étude sur une approche par cohérence de zone : le clustering ne porte plus sur les pixels mais sur des régions construites dans les images. Nous présentons également une série d'expériences réalisées sur cette méthode.

Dans le domaine de la planification et la gestion urbaine, de nombreux utilisateurs ont besoin de cartographier le territoire à l'échelle des blocs urbains dont chacun peut être défini comme le cycle minimum fermé par une voie de communication, correspondant à une échelle proche du $1/10.000^{e}$. Dans ce cas, il n'existe pas de résolution optimale : les images HR de 30 à 10m ont une résolution spatiale trop faible alors que les images THR ont une résolution spatiale trop fine pour la carte urbaine de blocs.



(a) Multispectrale (7 clusters) (b) Panchromatique (7 clusters) (c) Fusion (7 clusters) (d) Multisource (7 clusters)

Fig. 6.5: *Résultats obtenus pour les quatre cas.*

Ce problème peut être traité comme un problème de clustering multisource, où les images avec des résolutions spatiales différentes peuvent être utilisées simultanément sans nécessairement de supervision. Pour résoudre ce problème, la question du nombre de clusters sur chaque image n'est pas simple. À faible résolution (HR), les zones urbaines peuvent être classées en 5 à 7 classes associées à la couverture des sols. En haute résolution (THR), le nombre de clusters est plus élevé (10 à 15) car se rapportant aux différents matériaux entrant dans la composition des objets urbains. Par exemple, les bâtiments peuvent être différenciés par les matériels utilisés pour construire les toits. Pour être en mesure d'offrir aux utilisateurs finaux une cartographie au $1/10.000^e$ des zones urbaines, le nombre de classes sémantiques doit varier entre 7 et 9 classes. Or, ces classes sémantiques ne peuvent pas être directement obtenues par un processus de classification unique d'images HR ou THR.

Dans ce contexte, l'objectif de nos recherches a été de proposer une nouvelle méthode qui utilise simultanément les informations contenues dans les deux images présentant deux résolutions complémentaires (HR et THR).

6.2.4.1 Notations pour le clustering multisource d'image

Une image $\mathcal{I}$ peut être vue comme une fonction :

$$\begin{array}{rl} \mathcal{I}: & E \subset \mathbb{Z}^2 \mapsto \mathbb{Z}^b \\ & p \mapsto \mathcal{I}(p) \end{array} \tag{6.2}$$

où $\mathcal{I}(p) = \langle I_1(p), \ldots, I_a(p) \ldots, I_b(p) \rangle$ avec $b \in \mathbb{N}^*$ le nombre de bandes radiométriques et $I_a(p)$ la réponse spectrale du pixel p sur la a-ième bande.

On appelle image *clustering-pixel* $\mathcal{C}$, une image construite à partir d'un clustering des pixels de l'image de la façon suivante :

$$\begin{array}{rl} \mathcal{C}: & E \subset \mathbb{Z}^2 \mapsto [1, K], K \in \mathbb{Z} \\ & p \mapsto \mathcal{C}(p) \end{array} \tag{6.3}$$

où $\mathcal{C}(p)$ est l'étiquette du cluster auquel appartient le pixel p et K le nombre de clusters.

Dans une telle image, une région O_i est définie par :

$$O_i = \{p, q \in \mathcal{I} : \mathcal{C}(p) = \mathcal{C}(q) \wedge \textit{connecté}(p, q) = 1\} \tag{6.4}$$

où *connecté* est la fonction classique de 8-connexité dans $\mathcal{I}$. À noter que le nombre de régions dépend du clustering et ne peut être déterminé *a priori*. Soit N_r ce nombre de régions.

On appele *image-région* $\mathcal{R}$, l'image construite à partir d'un clustering $\mathcal{C}$ des pixels de l'image de la façon suivante :

$$\mathcal{R}(\mathcal{C}) = \{O_n, \forall n \in [1, N_r]\} \tag{6.5}$$

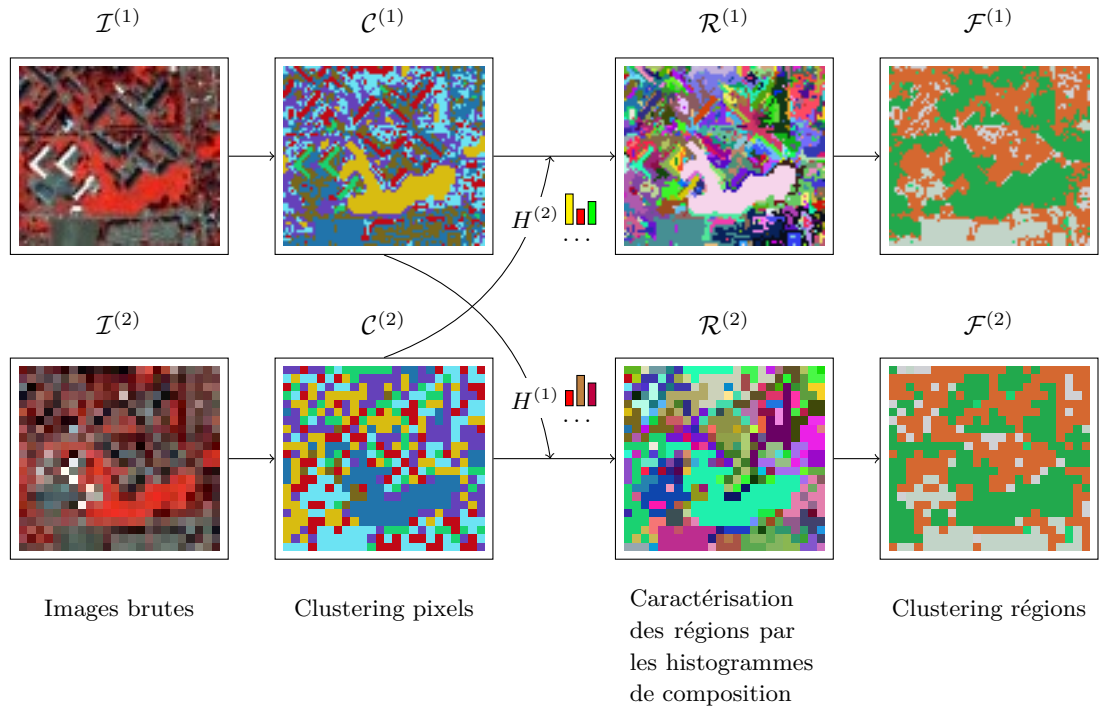


Fig. 6.6: *Étapes de la méthode de clustering multisource.*

Dans la suite, nous nous intéressons uniquement au cas on l'on dispose de deux images $\mathcal{I}^{(n)}$ et $\mathcal{I}^{(n')}$ présentant des résolutions $r^{(n)}$ et $r^{(n')}$ telles que $r^{(n)} \geqslant r^{(n')}$.
Soit $\lambda_{(n,n')}$ une fonction de correspondance qui associe à un pixel de $\mathcal{I}^{(n)}$, un pixel de $\mathcal{I}^{(n')}$. La fonction $\lambda_{(n,n')}$ peut être définie en utilisant par exemple le géoréférencement des deux images.
Soient $\mathcal{C}^{(n)}$ l'image clustering-pixel associée à $\mathcal{I}^{(n)}$, $K^{(n)}$ le nombre de clusters dans $\mathcal{C}^{(n)}$ et $\mathcal{R}^{(n)}$ l'image des régions associée à $\mathcal{C}^{(n)}$. De manière similaire, soient $\mathcal{C}^{(n')}$ l'image clustering-pixel associée à $\mathcal{I}^{(n')}$, $K^{(n')}$ le nombre de clusters dans $\mathcal{C}^{(n')}$ et $\mathcal{R}^{(n')}$ l'image des régions associée à $\mathcal{C}^{(n')}$

6.2.4.2 Description de la méthode proposée

Notre méthode peut être caractérisée de basée régions, car elle manipule des régions. Elle se compose de quatre étapes. Tout d'abord, un clustering des pixels est fait indépendamment sur les deux images. Ensuite, pour chaque image, les régions sont construites et caractérisées en fonction des images clustering-pixel. Puis, un algorithme de clustering est appliqué sur ces régions. Les quatre étapes de notre approche sont donc :

1. *Clusterings initiaux* : les deux images sont classées de façon indépendante afin d'obtenir les images clustering-pixels (voir équation (6.3)).

2. *Construction des régions* : les deux région-images sont construites (voir équation (6.5)) à partir des images clustering-pixels.

3. *Caractérisation des régions* : chaque region $R_i^{(n)} \in \mathcal{R}^{(n)}$ de chaque région-image est caractérisée par sa composition en terme de clusters dans l'image clustering-pixel $\mathcal{C}^{(n')}$: pour chaque région $R_i^{(n)} \in \mathcal{R}^{(n)}$, nous calculons un histogramme représentant la distribution des étiquettes des clusters associés aux pixels de $\mathcal{I}^{(n')}$ correspondant aux pixels de $R_i^{(n)}$ en utilisant la fonction de correspondance $\lambda_{n,n'}$.

   Un histogramme de composition $H^{(n')}(R_i^{(n)})$ associé à une region $R_i^{(n)}$ relativement à une

image clustering-pixel $\mathcal{C}^{(n)}$ est défini par :

$$H^{(n')}(R_i^{(n)}) = \langle h_{i,1}^{(n')}, \ldots, h_{i,k_{n'}}^{(n')} \rangle, R_i^{(n)} \in \mathcal{R}^{(n)} \tag{6.6}$$

où $K^{(n')}$ est le nombre de clusters dans $\mathcal{C}^{(n')}$ et :

$$h_{i,j}^{(n')} = \mid \{q = \lambda_{(n,n')}(p) : \mathcal{C}^{(n')}(q) = j,\ \forall p \in R_i^{(n)}\} \mid \tag{6.7}$$

Ce processus est également effectué dans *l'autre sens*, c'est-à-dire en considérant les régions $R_i^{(n')} \in \mathcal{R}^{(n')}$ et leur composition en clusters dans l'image clustering-pixel $\mathcal{C}^{n'}$.

4. *Classification basée régions* : Pour chaque région-image $\mathcal{R}^{(n)}$ un algorithme de clustering est appliqué de façon indépendante sur l'ensemble de ses objets (c'est-à-dire les régions caractérisées) en utilisant les histogrammes de composition comme attribut. Soit $C_{n'}(R_i^{(n)})$ l'étiquette du cluster associé à la région $R_i^{(n)}$ relativement à une image clustering-pixel $\mathcal{C}^{(n')}$. Enfin, l'image finale $\mathcal{F}_{n'}(n)$ correspondant à la classification des régions construite comme suit :

$$\begin{array}{rl} \mathcal{F}: & E \subset \mathbb{Z}^2 \mapsto [1, L^{(n)}] \\ & p \mapsto C_{n'}(R_i^{(n)}), p \in R_i^{(n)} \end{array} \tag{6.8}$$

où L_n est le nombre de clusters demandé dans le clustering des régions de l'image $\mathcal{I}^{(n)}$.
On notera que $L^{(n)}$ est très souvent différent de $K^{(n)}$. Ceci est dû au fait que le premier clustering est basée pixel alors que le deuxième est basé régions. En général, $K^{(n)} > L^{(n)}$.

Ce clustering région est également effectué dans *l'autre sens*, c'est-à-dire en effectuant un clustering de la région-image $\mathcal{R}^{(n')}$. La figure 6.6 illustre les étapes du processus.

6.2.4.3 Expérimentations

Les expériences ont été faites sur des images multispectrales avec des résolutions spatiales de 2,8m et 20m, sur une zone urbaine de Strasbourg (France). Ces images proviennent de deux capteurs (QUICKBIRD © DigitalGlobe Inc. et SPOT4 © CNES), prises respectivement en mai et juillet 2001. L'image QUICKBIRD propose quatre bandes spectrales (bleu, vert, rouge et proche infrarouge) alors que l'image SPOT4 propose trois bandes spectrales (vert, rouge, proche infrarouge). Les images sont géoréférencées dans la même projection cartographique (Lambert I nord).

Les deux images (figure 6.7(a) et figure 6.7(b)) correspondent à un extrait de la zone urbaine de Strasbourg. Cette zone est un exemple typique de banlieue avec des surfaces d'eau (au centre), une zone forestière dans le sud, des zones industrielles, des zones agricoles (présentant des réponses spectrales différentes en raison de la saison (le sol nu sur l'image THR, prise en mai, peut apparaître en rouge sur l'image HR, prise en juillet) et de zones de bâti pavillonnaire et/ou collectif (en rouge, noir et blanc sur l'image HR, en rouge, bleu et blanc sur l'image THR)

Dans les expériences, nous avons appliqué les 4 étapes décrites dans la section 6.2.4.2.

1. *Clusterings initiaux* : dans toutes les expériences, chaque image est traitée à l'aide de KMEANS, paramétré avec un nombre de clusters dépendant de la résolution spatiale. Pour l'image HR, beaucoup de travaux antérieurs montrent que les zones urbaines peuvent être classées en 6 clusters. Pour l'image THR, le nombre de clusters dépend des matériaux des objets. Afin de trouver le meilleur nombre de clusters en fonction de la zone d'étude, trois expériences avec respectivement 10, 15 et 20 clusters ont été menées.

   Ces expériences ont montré :
   - qu'avec 10 clusters, les régions sont trop vastes et il n'y en a pas assez à classer ;
   - qu'avec 20 clusters, les régions sont trop petites et sont trop proches des pixels (chaque région ne contient que de 3 à 6 pixels).

   Les meilleurs résultats empiriques ont été obtenus avec 15 clusters.

(a) THR Image QUICKBIRD à $2,8m$. (b) HR Image SPOT à $20m$.

Fig. 6.7: *Extrait d'une zone urbaine de Strasbourg (France).*

2. *Construction des régions* : à partir de deux images-clustering, les régions sont construites, en intégrant dans une même région les pixels connexes appartenant au même cluster. La figure 6.8 montre les cartes de régions obtenues.
3. *Caractérisation des régions* : les histogrammes de composition sont calculés.
4. *Classification basée régions* : après exécution de l'algorithme KMEANS indépendamment sur chacune des deux images de régions, nous obtenons les classifications finales.

Les trois premières étapes sont faites une seule fois. La quatrième étape a été testée avec 7, 8 et 9 clusters (correspondant aux nombres de clusters habituellement attendus en analyse niveau *bloc urbain*). Les résultats présentés ici se focalisent sur un extrait de la zone étudiée (nord-ouest de la figure 6.7), avec 7, 8 et 9 clusters.

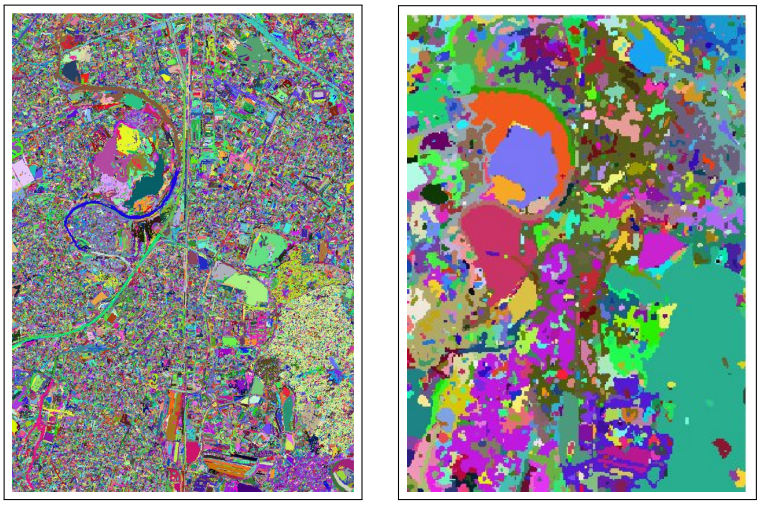
6.2.4.4 Analyse des résultats

Les résultats sont évalués par une comparaison avec une vérité terrain issue d'une base de données (BDOCS 2000 Cigal 2003) utilisé pour une cartographie au $1/10.000^e$. Cette carte contient 8 classes thématiques liées aux blocs urbains. Seulement 7 classes thématiques sont présentes sur l'extrait montré sur la figure 6.9(b).

La figure 6.10 montre les résultats obtenus avec respectivement 7, 8 et 9 clusters sur cet extrait. Les 7 clusters disponibles sur la première image (figure 6.10(a)) ne correspondent pas exactement à ceux de la vérité terrain. En effet, les blocs industriels sont dans le même groupe que les surfaces d'eau, et il y a 2 clusters représentant les blocs urbains discontinus (individuels en orange et collectifs en rouge). Dans l'image à 8 clusters (figure 6.10(b)), les blocs industriels apparaissent dans le 8-è*me* cluster (en violet). Enfin, une nouvelle classe de végétation est mise en évidence dans le résultat à 9 clusters (figure 6.10(c)).

Nous avons choisi d'évaluer nos résultats en utilisant l'indice Kappa [Congalton, 1991]. Cet indice va nous permettre d'évaluer quantitativement la qualité de nos résultats par rapport à une vérité terrain. Cet indice peut être vu comme un indicateur de concordance entre deux classifications. Il permet d'évaluer le pourcentage des bonnes correspondances qui sont dues à la réalité du terrain et non uniquement au hasard. Il est défini par :

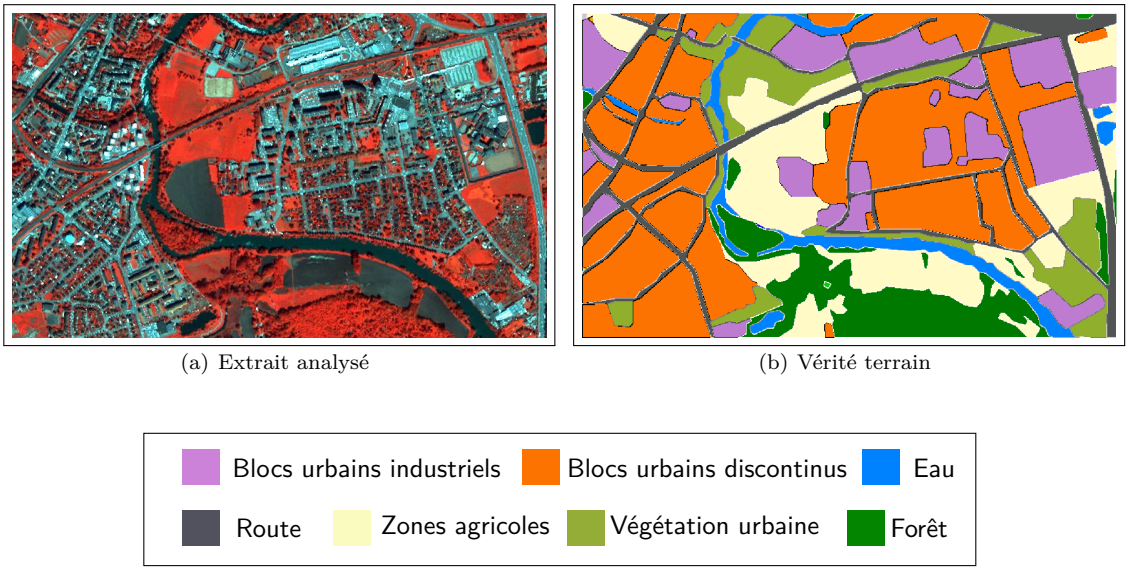
$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{6.9}$$



(a) Image THR (b) Image HR

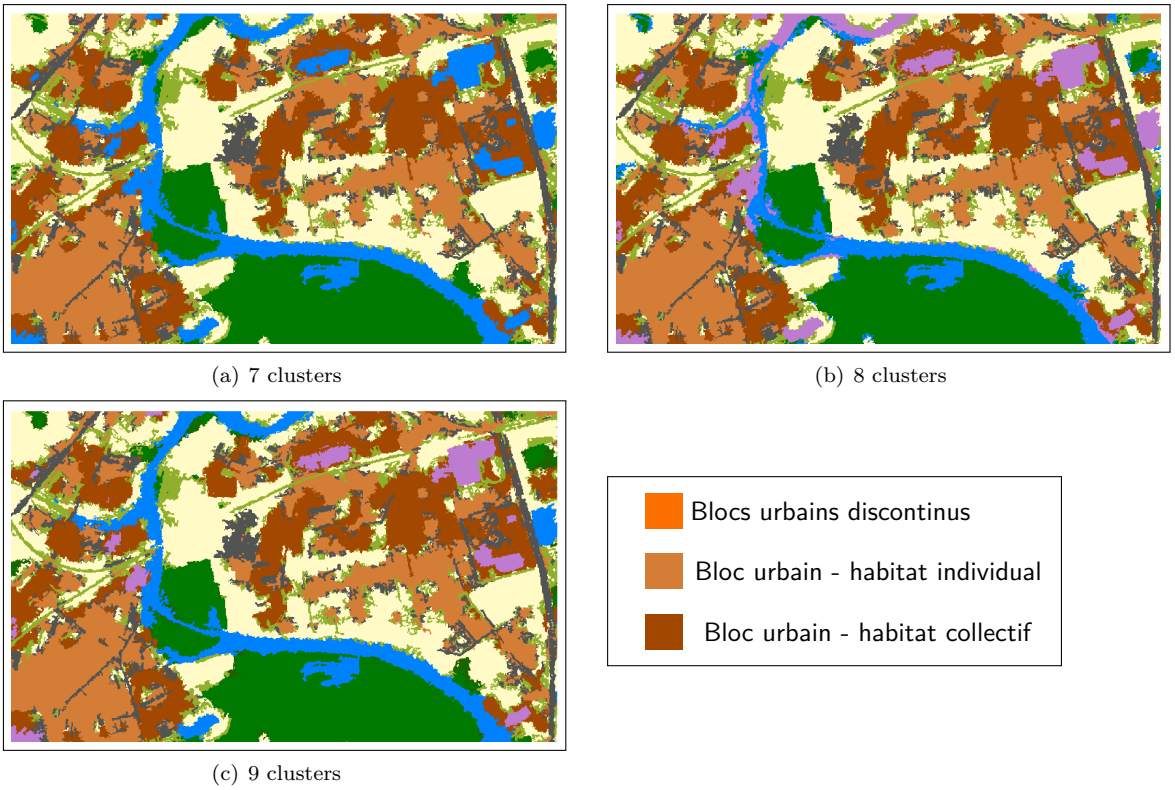
Les couleurs des régions n'ont pas de sémantique et ont été choisies aléatoirement.

Fig. 6.8: *Régions construites à partir des clusterinsg initiaux.*



(a) Extrait analysé (b) Vérité terrain

Fig. 6.9: *Extrait de la zone d'étude et vérité terrain (BDOCS 2000 CIGAL 2003)*



(a) 7 clusters (b) 8 clusters

(c) 9 clusters

Fig. 6.10: *Clusterings des régions avec différent nombre de clusters (7, 8 et 9).*

où Pr(a) est la concordance observée et Pr(e) est la probabilité estimée du hasard. Une valeur entre 1.00 et 0.81 indique une concordance parfaite, entre 0.80 et 0.61 indique une bonne concordance, etc.

Nous avons également évalué nos résultats par rapport à une classification non supervisée de l'image THR :

- clustering pixel avec KMEANS à 7, 8 et 9 clusters ;
- clustering basé région avec KMEANS clustering à 7, 8 et 9 clusters : seules les informations spectrales ont été utilisées, les régions ont été créées à partir d'un clustering pixel à 15 clusters ;
- clustering basé région en utilisant le logiciel Definiens Professional [1] : seules les informations spectrales ont été utilisées.

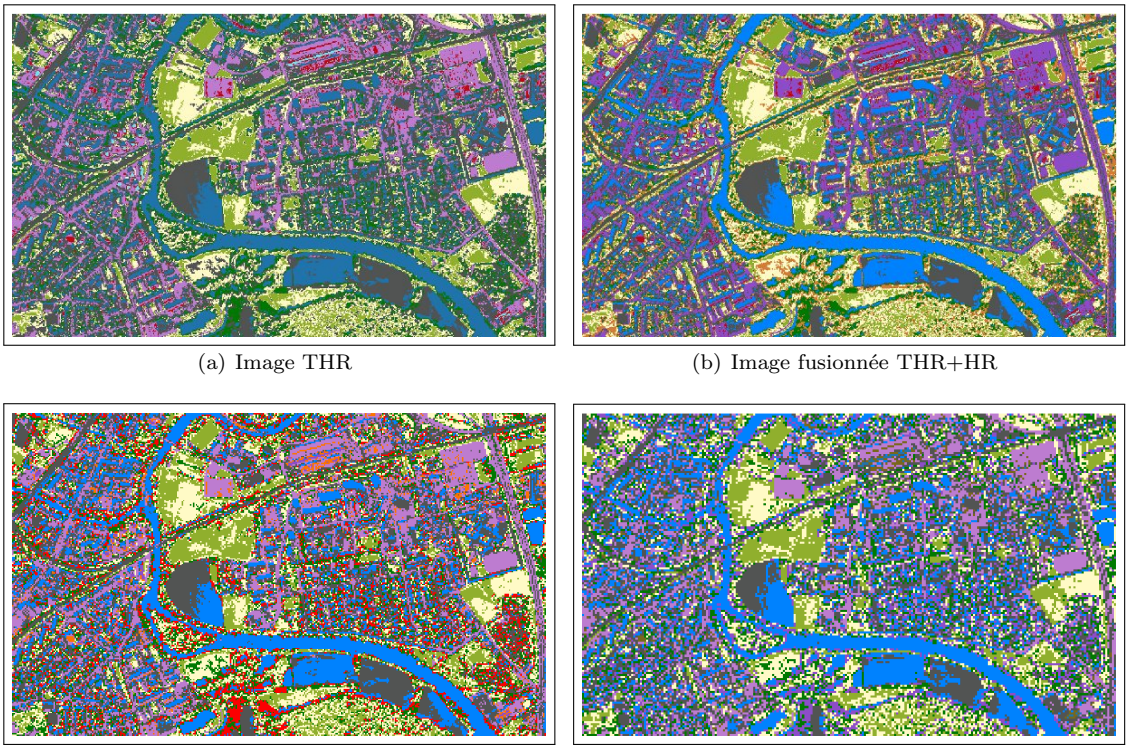
Enfin, nous les avons évalués par rapport aux résultats obtenus par KMEANS :

- sur une image issue de la fusion directe des deux images où, à chaque pixel, sont associées directement toutes les informations radiométriques des images HR et THR ;
- sur deux images dégradées de l'image THR avec un ratio de 1/2 et de 1/3.

La figure 6.11 illustre les résultats obtenus par la classification pixel à 9 clusters. La figure 6.12 montre ceux obtenus par une classification basée régions avec 9 clusters.

Les valeurs du Kappa obtenues sur les différents résultats sont présentées dans le tableau 6.2. Ces valeurs montrent que la qualité globale des résultats obtenus en multirésolution est très comparable aux méthodes basées régions et surpassent les autres.

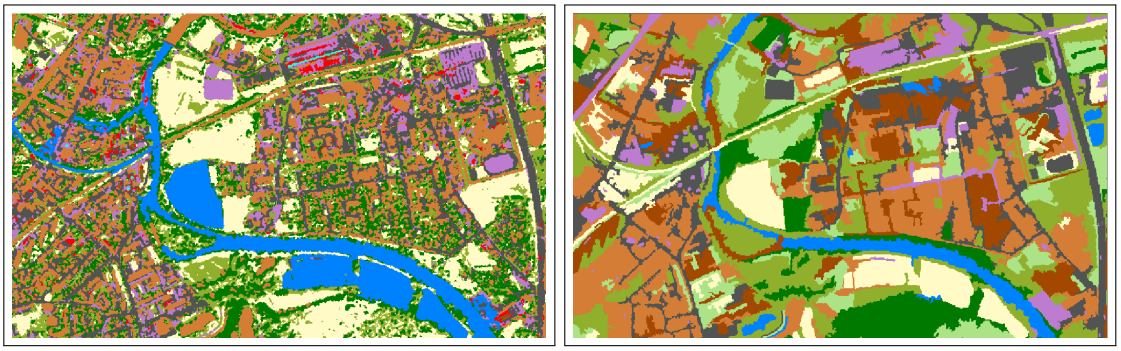
[1] `http://www.definiens.com/`



(a) Image THR (b) Image fusionnée THR+HR

(c) Image THR dégradée (ratio 1/2) (d) Image THR dégradée (ratio 1/3)

Fig. 6.11: *Clustering basé pixel (algorithme* KMEANS*) avec 9 clusters.*



(a) KMEANS (b) Definiens Professional

Fig. 6.12: *Classification des régions de l'image THR avec 9 clusters.*

	Nombre de clusters		
	7	8	9
La méthode multiresolution proposée	**0.73828**	**0.74259**	**0.74501**
KMEANS pixel	0.70805	0.70412	0.71832
KMEANS basé région	0.67864	0.68051	0.69002
Definiens Professional	0.70604	0.72811	0.73843
Image fusionnée	0.68772	0.70412	0.71831
KMEANS sur l'image dégradée (1/2)	0.68729	0.70396	0.71645
KMEANS sur l'image dégradée (1/3)	0.68569	0.70536	0.71620

Tab. 6.2: *Valeur du Kappa pour les différentes expériences*

6.2.5 Bilan

Dans les sections précédentes nous avons présenté nos propositions pour le clustering collaboratif de données multisources pour l'interprétation d'images de télédétection. Dans une première phase, la méthode collaborative SAMARAH présentée au Chapitre 3 a été utilisée pour effectuer du clustering de données multisources. Il a été nécessaire d'adapter la méthode pour qu'elle puisse traiter des données n'ayant pas la même représentation. Le géoréférencement des images a été utilisé pour mettre en correspondance les pixels d'images à différentes résolutions. Quand la différence entre les résolutions des différentes images n'est pas trop importante, cette approche donne de bons résultats.

Dans une deuxième phase, nos recherches ont consisté à étudier les modifications à apporter à ce mécanisme pour permettre à chacune des méthodes de rechercher un nombre de clusters pouvant être différent. Cette réflexion a débouché sur la définition d'une méthode de classification multisource pouvant traiter des images ayant une forte différence de résolution. Dans cette méthode, un algorithme de clustering est tout d'abord appliqué sur chacune des deux images. Puis, des régions sont construites indépendamment dans chacune des images en considérant les pixels connexes appartenant au même cluster. Ces régions sont ensuite décrites par des histogrammes, construits en observant la distribution des clusters affectés aux pixels correspondants aux pixels de la région dans l'autre image. Cette correspondance est effectuée grâce au géoréférencement. Enfin, un algorithme de clustering est utilisé avec en entrée les régions décrites par les histogrammes. Les développements effectués lors de cette phase sont présentés dans la section 6.2.4. Cette méthode a donné de très bons résultats et a été validée par une publication dans une revue internationale [Wemmert et al., 2009]. Une thèse commencée en octobre 2010 a pris la suite de ces travaux [Kurtz et al., 2010].

6.3 Données multisources pour la simulation de capteurs

6.3.1 La simulation de capteurs

6.3.1.1 Introduction

Comme nous l'avons vu jusqu'ici, un nombre important de capteurs satellitaires sont disponibles pour aquérir des images de la surface terrestre. Ces données de télédétection sont utilisées intensivement pour étudier la Terre et collectent des données dans de nombreux domaines.

Chaque satellite a ses propres caractéristiques, l'une des plus importantes étant sa résolution spectrale. La résolution spectrale d'un capteur peut être caractérisée par le nombre de bandes spectrales, leurs largeurs, et leurs positions sur le spectre [Herold et al., 2003].

Plusieurs études ont montré [Heidena et al., 2007; Herold et al., 2003; Meyer et Chander, 2007] que la résolution spectrale des capteurs est un point critique, particulièrement pour discriminer différents types d'occupation du sol dans des environnements complexes tels que le milieu urbain. Malgré l'augmentation de la disponibilité des données hyperspectrales, les capteurs multispectraux embarqués à bord de plusieurs satellites acquièrent chaque jour une très grande masse de données avec une résolution spectrale relativement pauvre. La plupart des systèmes satellitaires possède 4 à 7 bandes spectrales allant du visible à l'infrarouge sur le spectre électromagnétique [Govender et al., 2007]. Il existe cependant quelques capteurs qui utilisent également des bandes thermiques. L'un des avantages de ces capteurs multispectraux, comparés aux capteurs hyperspectraux, est la couverture spatiale plus importante, qui permet de cartographier plus rapidement de grandes zones.

Le nombre de systèmes satellitaires disponibles augmentant, un challenge important est d'évaluer la complémentarité de ces capteurs pour une application donnée. En effet, les informations fournies par les différents capteurs peuvent être complémentaires, et un problème clé est de proposer des systèmes capables d'utiliser ces sources d'informations hétérogènes dans un processus unique. Cependant, acquérir des images satellites est toujours très coûteux, c'est pourquoi il serait intéressant d'évaluer a priori la complémentarité des capteurs. Pour résoudre ce problème, nous adoptons

dans cette thèse une approche par simulation. La simulation de capteurs, également appelée simulation de bande, consiste à générer des données multispectrales à partir de données acquises par un autre capteur existant ayant une meilleure résolution spatiale. Cette simulation consiste à combiner des bandes fines en bandes plus larges. Ce type d'approche a déjà été utilisé, particulièrement pour la calibration de capteur. L'étape de simulation utilise les RSR (*Relative Spectral Response*) des capteurs multispectraux, qui décrivent la réponse spectrale de chaque bande simulée.

Les spectres utilisés pour la simulation proviennent de librairies spectrales qui sont des dépôts de spectres de différents matériaux (par exemple des minéraux, de la végétation, etc.) généralement capturés sur le terrain en utilisant des spectromètres. Nous avons utilisé ces librairies ainsi que les caractéristiques techniques de plusieurs capteurs pour simuler différentes vues des spectres de ces librairies. À l'issue de cette étape de simulation nous disposons d'un ensemble d'objets tels qu'ils auraient été perçus par les différents capteurs. Chaque objet est décrit par un certain nombre d'attributs correspondant au nombre de bandes spectrales du capteur. Ce nombre ainsi que la nature des bandes sont différents pour chaque capteur. On dispose bien alors de vues différentes de la même donnée originale (la libraire spectrale). La seule différence entre les données sont les caractéristiques du capteur utilisé lors de la simulation.

Pour évaluer l'intérêt d'utiliser plusieurs vues conjointement, nous nous sommes intéressés à évaluer l'utilisation de couples de capteurs. L'objectif est d'étudier si l'utilisation de couples de vues provenant de satellites différents améliore la qualité des résultats.

6.3.1.2 Méthodes de simulation de bande

Comme indiqué précédemment, le principe de la simulation de capteurs est de générer un spectre multispectral simulé à partir de données acquises à partir de capteurs ayant une meilleure résolution spectrale (hyperspectrale). La simulation consiste à combiner plusieurs bandes hyperspectrales voisines pour former la bande plus large correspondante dans la simulation multispectrale. Elle est réalisée par l'utilisation des fonctions de réponse spectrale relative ou RSR (*Relative Spectral Response*) du capteur multispectral à simuler. Ces fonctions décrivent la réponse spectrale de chacune des bandes spectrales du capteur. La figure 6.13 présente la fonction de RSR de deux systèmes satellites bien connus : QUICKBIRD et SPOT5.

La simulation de capteurs a été utilisée dans plusieurs types d'applications. Par exemple, Salvatore et al. [1999] ont simulé la réponse d'un nouveau capteur à partir de données AVIRIS hyperspectrales. Cela a permis aux scientifiques d'évaluer le potentiel de leur nouveau capteur multispectral et de paramétrer au mieux les RSR afin d'obtenir de meilleurs résultats en fonction de leurs objectifs. Herold et al. [2003] ont étudié différentes résolutions spectrales pour l'analyse de tissu urbain. Pour cela, ils ont utilisé des données hyperspectrales AVIRIS et une librairie de spectres mesurés sur des matériaux (bitume, tuiles, végétation, etc.) afin de trouver quelles bandes spectrales permettaient de séparer au mieux les classes urbaines d'occupation du sol (bâtiments, routes, végétation, etc.). Les données AVIRIS ont aussi été utilisées pour simuler des données LANDSAT et IKONOS. Les résultats ont montré que IKONOS et LANDSAT manquaient de finesse spectrale pour séparer certaines classes urbaines.

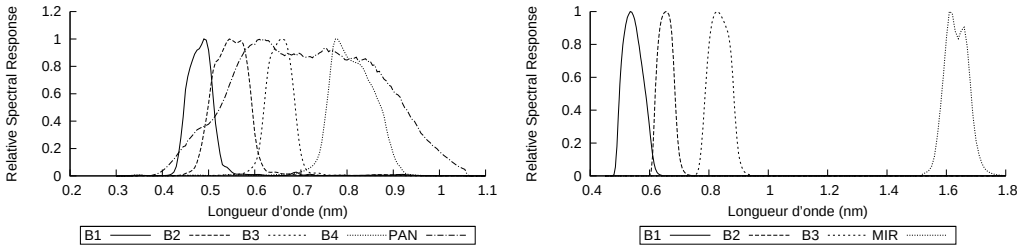
Afin de simuler des données multispectrales à partir de données hyperspectrales, il faut fusionner les bandes hyperspectrales voisines afin de simuler les bandes multispectrales. Cependant, les réflectances des bandes proches à fusionner ne peuvent pas être simplement additionnées. En fait, elles doivent être pondérées en fonction du RSR des bandes multispectrales. Cette sensibilité est décrite pour chaque capteur par sa fonction de RSR.

Comme évoqué par Clark et al. [2002], plusieurs stratégies différentes ont été proposées pour calculer les pondérations à appliquer à chaque bande hyperspectrale. Pour la simulation effectuée dans cette thèse, chaque longueur d'onde hypespectrale a été liée avec la moyenne de la RSR (dans l'intervalle de la largeur à mi-hauteur de chaque bande hyperspectrale simulée). Cette approche est similaire à celle proposée par Franke et al. [2006] et est décrite en détail dans [Forestier et al., 2009a].

Il est important de préciser que certains paramètres externes ne sont pas utilisés dans cette

Nom	# Bandes spectrales	Propriétaire
SPOT5	4	CNES
QUICKBIRD	4	Digital Globe
PLEIADES	5	CNES
LANDSAT	6	NASA
IKONOS	5	Satellite Imaging Corporation
FORMOSAT	5	Taiwan

Tab. 6.3: *Liste des capteurs étudiés lors des simulations.*



(a) QUICKBIRD (b) SPOT5

Fig. 6.13: *Exemple de deux réponses spectrales relatives (RSR).*

étude. Par exemple, d'autres approches [Kavzoglu, 2004] prennent en compte d'autres variables comme les paramètres atmosphériques ou les différences géométriques entre les différents capteurs. Dans nos travaux, nous nous intéressons à la capacité des différents capteurs à séparer les différentes classes en fonction de leur RSR, c'est pourquoi nous nous concentrons sur les différences spectrales entre les capteurs uniquement. Cependant, d'autres aspects comme la résolution spatiale devraient aussi être étudiés afin de mieux appréhender les différences entre les capteurs. Les six capteurs étudiés ici sont : SPOT5, QUICKBIRD, PLEIADES, LANDSAT, IKONOS et FORMOSAT (voir le tableau 6.3).

6.3.2 Les librairies spectrales

6.3.2.1 Données disponibles

Une librairie spectrale est une base de données de spectres de plusieurs types de matériaux (minéraux, objets artificiels, végétation, etc.) mesurés *in situ* à partir d'un spectromètre.

Plusieurs librairies spectrales libres de droit existent. Dans ces travaux, nous avons utilisé la librairie ASTER [Baldridge et al., 2008] qui se compose de spectres provenant du JPL (*Jet Propulsion Laboratory*), de la JHU (*John Hopkins University*) et de l'USGS (*United States Geological Survey*). Cette librairie comporte des spectres de différentes roches, minéraux, sols lunaires, sols terrestres, matériaux artificiels, météorites, végétation, neige et glace, qui couvrent les longueurs d'onde du visible à l'infrarouge thermique (0,4-15,4μm). La première version date de juillet 1998 et la seconde est disponible depuis 2007 sur simple demande via le site web d'ASTER[2]. La librairie ASTER est, à notre avis, la plus simple et complète disponible librement.

Les spectres de cette librairie ont été convolués avec le RSR de chaque capteur afin de créer plusieurs points de vue de la librairie. En fait, le processus de simulation permet de construire la *vue* qu'aurait chaque capteur des données disponibles dans la librairie. Ceci est illustré sur la figure 6.14 qui représente le spectre complet d'un matériau rentrant dans la composition de toit extrait de la librairie ASTER, et la figure 6.15 qui présente le spectre simulé obtenu pour chacun des capteurs étudiés. Le but est d'utiliser ces différentes vues de la même donnée afin d'évaluer le gain apporté par l'utilisation conjointe de plusieurs satellites pour identifier les différentes occupations du sol.

[2] `http://speclib.jpl.nasa.gov`

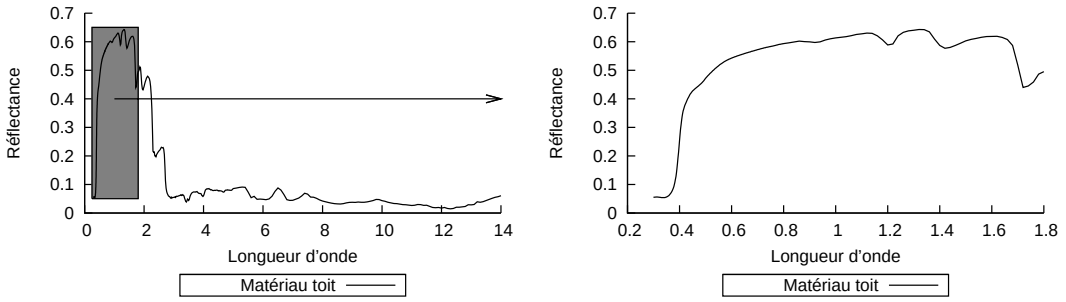


Fig. 6.14: *Exemple du spectre complet d'un matériau rentrant dans la composition de toit extrait de la librairie ASTER.*

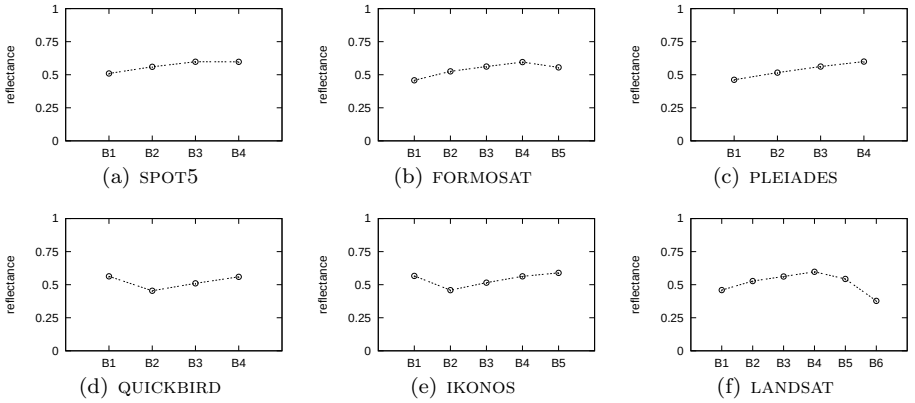


Fig. 6.15: *Exemple de spectres simulés à partir d'un spectre de la librairie ASTER (figure 6.14).*

6.3.3 Application en clustering collaboratif

6.3.3.1 Expérimentations

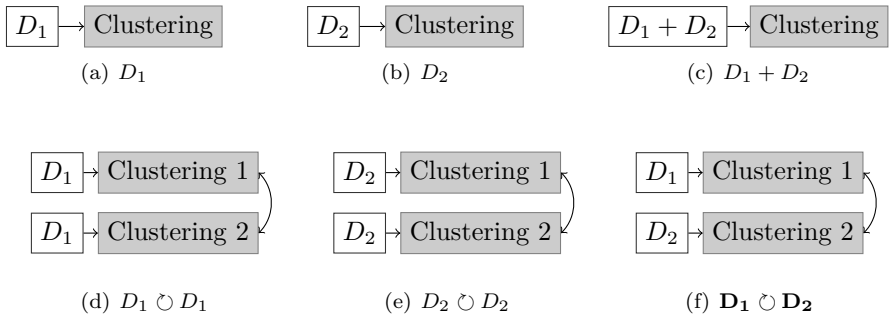
Pour vérifier l'intérêt d'utiliser des données multisources provenant de plusieurs capteurs différents, plusieurs configurations ont été évaluées pour chaque couple de capteurs. Soit D_1 les données issues du premier capteur et D_2 les données issues du second capteur. Les configurations évaluées sont les suivantes :

- D_1 : seulement la première vue
- D_2 : seulement la seconde vue
- $D_1 + D_2$: fusion de D_1 et de D_2 (c'est-à-dire aggregation des attributs)
- $D_1 \circlearrowleft D_1$: collaboration utilisant deux fois D_1
- $D_2 \circlearrowleft D_2$: collaboration utilisant deux fois D_2
- $D_1 \circlearrowleft D_2$: collaboration utilisant les deux vues D_1 et D_2

La figure 6.16 illustre ces différentes configurations. Dans chacune des expériences, nous avons utilisé l'algorithme KMEANS comme méthode de base. Pour chaque configuration ne contenant qu'une seule vue (les trois premières), l'algorithme a été initialisé pour trouver un nombre de clusters égal au nombre de classes. Pour les configurations en mode collaboratif (les trois autres), chaque méthode a été initialisée aléatoirement avec un nombre de clusters choisi dans $[5; 10]$. Ce choix a été fait pour assurer une certaine diversité entre les deux résultats impliqués dans la collaboration. Les résultats de clustering obtenus avec et sans collaboration ont été évalués à l'aide d'indices d'évaluation de partition (par exemple *Rand*, *Jaccard*). Le tableau 6.4 illustre les résultats sous forme d'une matrice pour chaque couple de capteurs. Le symbole • indique que la configuration

D_1 \ D_2	FORMOSAT	PLEIADES	QUICKBIRD	SPOT5	LANDSAT	IKONOS
FORMOSAT	-	○	○	●	●	○
PLEIADES		-	○	●	●	●
QUICKBIRD			-	●	●	○
SPOT5				-	○	●
LANDSAT					-	●
IKONOS						-

Tab. 6.4: *Évaluation de la collaboration de couples de capteurs*



(a) D_1 (b) D_2 (c) $D_1 + D_2$

(d) $D_1 \circlearrowleft D_1$ (e) $D_2 \circlearrowleft D_2$ (f) $\mathbf{D_1 \circlearrowleft D_2}$

Fig. 6.16: *Les différentes configurations évaluées.*

$D_1 \circlearrowleft D_2$ a fourni de meilleurs résultats que toutes les autres configurations, le symbole ∘ indique qu'au moins une autre configuration a donné un meilleur résultat que $D_1 \circlearrowleft D_2$.

Dans ce tableau, il apparait que les capteurs ayant des fonctions RSR proches ne tirent pas parti de la collaboration. Par exemple, la collaboration des capteurs QUICKBIRD et PLEIADES dont les RSR sont très proches n'est pas bénéfique. A contrario, la collaboration de SPOT5 avec QUICKBIRD ou PLEIADES semble bénéfique, SPOT5 possédant une bande dans le moyen infra-rouge que ces deux autres capteurs ne possèdent pas. La conclusion que l'on peut tirer de ces premiers résultats est qu'il semble que la collaboration soit bénéfique quand les capteurs ne portent pas exactement la même information.

6.3.3.2 Évaluation par classifieurs supervisés

Les résultats obtenus dans les expériences menées dans la section précédente montrent qu'il semble intéressant de faire collaborer plusieurs vues si les données ne sont pas similaires. En effet, d'après les résultats obtenus, plus les capteurs sont différents, et plus ceux-ci apportent des informations différentes et potentiellement complémentaires. Pour valider cette hypothèse, nous avons appris un modèle prédictif à l'aide d'une méthode de classification supervisée (Bayésien Naïf) pour chacune des vues. Nous avons ensuite comparé les prédictions des différents modèles appris. Soit p_1 et p_2 deux modèles prédictifs et $p_1(o_i)$ et $p_2(o_i)$ les prédictions pour ces deux modèles pour l'objet o_i (même objet mais représenté de manière différente par les deux vues). Nous avons calculé un coefficient d'accord qui correspond au pourcentage d'accord sur l'ensemble des prédictions des n objets :

$$pr = \sum_{i=0}^{n} \frac{(p_1(o_i) = p_2(o_i))}{n} \tag{6.10}$$

Le tableau 6.5 présente les résultats pour chaque couple de capteurs. En mettant ces résultats en rapport avec ceux obtenus en clustering collaboratif, on peut y observer un lien fort (voir valeurs en gras). On observe que pour que l'utilisation de différentes vues soit bénéfique il est nécessaire

D_1 \ D_2	FORMOSAT	PLEIADES	QUICKBIRD	SPOT5	LANDSAT	IKONOS
FORMOSAT	100	91, 85	97, 78	**85, 93**	**80, 74**	99, 26
PLEIADES		100	94, 07	**92, 59**	**83, 7**	**92, 59**
QUICKBIRD			100	**88, 15**	**82, 96**	98, 52
SPOT5				100	91, 11	**86, 67**
LANDSAT					100	**81, 48**
IKONOS						100

Tab. 6.5: *Pourcentage du nombre de fois où les deux classifieurs sont en accord.*

D_1 \ D_2	FORMOSAT	PLEIADES	QUICKBIRD	SPOT5	LANDSAT	IKONOS
FORMOSAT	-	0, 630	0, 642	**0, 592**	**0, 632**	0, 643
PLEIADES		-	0, 640	**0, 590**	**0, 631**	**0, 641**
QUICKBIRD			-	**0, 602**	**0, 628**	0, 632
SPOT5				-	0, 676	**0, 654**
LANDSAT					-	**0, 653**
IKONOS						-

Tab. 6.6: *Corrélation moyenne entre les différentes vues.*

que celles-ci portent des informations différentes. En effet, on peut imaginer qu'utiliser des données fortement similaires risque de ne pas améliorer le processus. Il cependant nécessaire que ces données partagent une certaine cohérence commune, et ne diffèrent que sur certains objets spécifiques. Cela pose le problème de pouvoir évaluer ces différences entre les vues et savoir *à quel point* ces données doivent diverger, et également de connaître l'impact de données *trop* dissimilaires.

6.3.3.3 Évaluation par critère statistique

Dans cette section, nous évaluons les similarités entre les différentes vues en utilisant un coefficient de corrélation dans le but de confirmer les résultats obtenus dans les deux sections précédentes. Ce coefficient de corrélation a été calculé comme la moyenne entre les corrélations des différents attributs des différentes vues. Comme chaque satellite possède un nombre de bandes différent et que celles-ci ne sont pas similaires ni ordonnées, il est nécessaire de calculer ce coefficient pour tous les couples de bandes pour un couple de capteurs donné. Soit $D_1 = (a_1, \ldots, a_{n_1}), D_2 = (a_1, \ldots, a_{n_2})$ les attributs (ou bandes) pour deux capteurs. La corrélation moyenne est évaluée telle que :

$$\mu_{corr} = \sum_{i}^{n_1} \sum_{j}^{n_2} \frac{r(a_i, a_j)}{(n_1 * n_2)} \tag{6.11}$$

avec $r(a_i, a_j)$ la coefficient de corrélation linéaire de Bravais-Pearson [Pearson, 1900] entre les valeurs des bandes considérées comme deux variables aléatoires :

$$r(a_i, a_j) = \frac{\rho_{a_i a_j}}{\rho_{a_i} \rho_{a_j}} \tag{6.12}$$

avec $\rho_{a_i a_j}$ la covariance de (a_i, a_j) et ρ_{a_i} et ρ_{a_j}, respectivement l'écart-type de a_i et a_j.

Le tableau 6.6 présente les résultats du calcul de cette corrélation pour les données issues des différents capteurs. La tendance identifiée précédemment se retrouve dans ces résultats.

6.3.4 Conclusion et perspectives

Dans cette section nous avons présenté des travaux sur l'utilisation de données multisources provenant de la simulation de capteurs. Une étape de simulation utilisant une librairie spectrale

permet de générer des vues de cette librairie à la résolution de différents capteurs. Ces données ont ensuite été utilisées dans un processus de clustering collaboratif pour étudier l'intérêt de faire collaborer des données issues de couples de capteurs. Une étude utilisant des modèles prédictifs supervisés ainsi que le calcul de coefficients de corrélation entre différentes vues ont permis d'identifier qu'il est intéressant de faire collaborer les vues si celle-ci sont légèrement dissimilaires.

La question maintenant posée est de pouvoir mieux évaluer ces dissimilarités, les quantifier, et évaluer quel niveau de dissimilarité est nécessaire pour obtenir une amélioration significative des résultats. Il est également nécessaire de prendre en compte plus de paramètre au niveau de la simulation afin de la rendre plus réaliste.

6.4 Bilan

Dans ce chapitre nous avons présenté nos recherches dans le cadre du clustering collaboratif multisource. La première partie du chapitre s'est intéressée à l'étude de l'utilisation conjointe d'images de télédétection hétérogènes (différents capteurs, différentes résolutions, etc.). Deux approches ont été proposées en fonction de la différence de résolution entre les images en entrée du système. Quand la différence de résolution n'est pas trop importante, le même nombre de clusters peut être recherché dans les deux images. La méthode de clustering collaboratif vue au Chapitre 3 peut alors être utilisée. Il est cependant nécessaire de modifier celle-ci pour être capable de comparer des images à résolutions différentes. Nous avons présenté comment ce verrou peut être levé à l'aide du géoréférencement des images. Quand la différence de résolution est trop importante, nous avons proposé un mécanisme de collaboration binaire qui utilise une approche orientée objet pour effectuer le clustering conjoint des images. Un calcul de cohérence a été proposé, qui est évalué en étudiant les histogrammes de clusters dans les régions construites. L'approche proposée a été évaluée sur une zone de l'agglomération de Strasbourg et a montré de très bons résultats.

Dans la deuxième partie du chapitre, nous avons présenté nos travaux sur l'utilisation de librairies spectrales pour la simulation de capteurs. Nous avons proposé un mécanisme permettant de simuler la réponse spectrale de différent capteurs à partir de ces librairies spectrales. Les données ainsi générées ont été utilisées pour évaluer l'intérêt potentiel de l'utilisation conjointe de couples de capteurs.

Contributions et valorisations

Les travaux effectués dans la première partie de ce chapitre ont permis le développement d'une méthode de clustering collaboratif multisource pour l'interprétation d'images de télédétection. Les résultats obtenus ont été validés par la publication de deux articles en revues internationales. Le premier article [Forestier et al., 2008c] a été publié dans un numéro spécial de *EURASIP Journal on Advances in Signal Processing*. Il présente les travaux sur la modification de la méthode collaborative et son adaptation au cas multisource. La second article [Wemmert et al., 2009] publié dans *IEEE Geoscience and Remote Sensing Letters* présente la méthode utilisant la cohérence lorsque les différences de résolution sont trop importantes. Ces travaux sont actuellement poursuivis dans le cadre d'une thèse débutée en octobre 2010 [Kurtz et al., 2010] à Strasbourg.

Les travaux effectués dans la seconde partie de ce chapitre ont permis la mise en place d'un processus complet de simulation de capteurs. Ce système permet de générer très facilement des données multisources pouvant servir à étudier l'intérêt de leur utilisation conjointe. Les propositions ont été validées par plusieurs publications en conférence internationale [Forestier et al., 2009a,b] ainsi que nationale [Forestier et al., 2010e]. Ces travaux sont actuellement poursuivis au CNES dans le cadre de l'intégration d'une chaîne de simulation de capteurs dans la librairie ORFEO ToolBox[3].

[3] `http://www.orfeo-toolbox.org/otb/`

Chapitre 7

Conclusion

Dans cette thèse, plusieurs aspects du processus de fouille de données ont été abordés dans le but de lever d'importants verrous concernant le traitement de données complexes. Dans une première partie, nous avons étudié la combinaison de résultats de clustering ainsi que la collaboration de méthodes de clustering. Ces approches ont connu un essor important ces dernières années dans le but d'améliorer la qualité des résultats des classifications produites et de limiter les biais liés à la sélection d'un unique algorithme. Nous nous sommes particulièrement intéressés au clustering collaboratif et avons proposé d'un nouvel algorithme optimisant la collaboration entre les résultats.

La deuxième partie de cette thèse s'est consacrée à l'étude de l'intégration de connaissances dans les algorithmes de clustering, et plus particulièrement en clustering collaboratif. En effet, avec des données de plus en plus complexes, il devient crucial de proposer des approches permettant d'intégrer des connaissances du domaine ou fournies par les experts. Ces connaissances doivent permettre de guider les algorithmes vers des solutions plus pertinentes. Nous avons étudié les différents types de connaissances disponibles ainsi que leur intégration dans le processus de clustering collaboratif. Les avantages des différentes solutions ont été présentés et comparés.

Enfin, la dernière partie de cette thèse a présenté des applications en observation de la Terre. Dans ce domaine, un verrou important concerne la représentation des connaissances et ses utilisations en interprétation automatique d'images de télédétection. Nous avons proposé un formalisme et un mécanisme de construction d'une base de connaissances ainsi que sa mise en œuvre dans différents processus d'identification (segmentation, classification automatique, etc.). Une méthode de clustering multisource a également été introduite, permettant de traiter des données hétérogènes multisources. Des applications en interprétation d'images de télédétection à différentes résolutions ont été proposées ainsi qu'un mécanisme de simulation de capteurs permettant d'étudier la complémentarité des données produites par ces capteurs.

7.1 Contributions

Le point de départ de notre analyse sur la combinaison de méthodes de clustering a été la méthode de clustering collaboratif proposée par Cédric Wemmert [Wemmert, 2000]. Une étude complémentaire de cette méthode a été menée afin de mieux comprendre son fonctionnement et ses propriétés. Nous avons par la suite proposé une étude sur la résolution de conflits en clustering collaboratif [Forestier et al., 2010d]. Cette étude nous a amené à la définition d'une nouvelle méthode de clustering collaboratif basé sur un algorithme génétique. Ce nouvel algorithme s'est révélé très efficace pour guider la collaboration entre les méthodes. Plus particulièrement, nous avons montré comment ce nouveau processus collaboratif explorait mieux l'espace des solutions. Enfin, un ensemble d'expériences a montré que cette méthode proposait de bons résultats tant sur des données artificielles que réelles. Cette approche se démarque des approches proposées dans la littérature en proposant une réelle collaboration entre les méthodes. En effet, la majorité des

méthodes proposées jusqu'alors se contente de fusionner des résultats existants.

Ces travaux ont ensuite été poursuivis par l'étude de l'intégration de connaissances dans les algorithmes de clustering. Un état de l'art a été présenté sur les différentes formes de connaissances ainsi que les avantages apportés par l'utilisation de celles-ci. Une étude plus précise a été menée pour le cas des connaissances sous la forme d'objets étiquetés, ces connaissances étant utilisées pour évaluer la pureté des clusters d'un résultat [Forestier et al., 2010c]. Une attention plus générale a été portée sur la présentation les différents niveaux d'intégration des connaissances en clustering collaboratif. Différents niveaux ont été identifiés, et la méthode collaborative SAMARAH a été modifiée pour pouvoir prendre en compte les connaissances aux différents niveaux [Forestier et al., 2010a, 2008e].

Les géosciences et plus particulièrement celles liées à l'observation de la Terre via les images de télédétection ont été le domaine privilégié d'application des propositions faites lors de cette thèse. Dans ce cadre nous avons proposé un formalisme de représentation des connaissances sur les objets géographiques [Durand et al., 2007]. Nous avons montré comment ces connaissances peuvent être utilisées pour la segmentation [Forestier et al., 2008a,b] ainsi que pour l'étiquetage de clusters [Forestier et al., 2008d]. Ces approches innovantes ont permis de mettre en place un processus automatique d'interprétation d'image guidé par les connaissances de l'expert. Loin de vouloir fournir un système totalement automatique, le but est d'aider l'expert dans le processus d'interprétation en lui fournissant des outils lui permettant d'accélérer les traitements.

Enfin, nous avons également abordé le thème de l'intégration de données multisources qui consiste à prendre en compte différentes sources d'informations dans un processus unique de fouille de données. Nous avons étudié l'utilisation de deux types de données hétérogènes. La première sous forme d'images de télédétection à différentes résolutions [Forestier et al., 2008c; Wemmert et al., 2009]. La seconde, à partir de librairies spectrales et de simulations de capteurs [Forestier et al., 2009a, 2010e, 2009b]. Une méthode de clustering collaboratif multisource a été mise en place et a permis de montrer l'intérêt et le potentiel de l'utilisation de multiples sources d'information.

7.2 Développement logiciel

Les développements effectués lors de cette thèse ont été intégrés dans les logiciels développés au sein de l'équipe FDBT. La plate-forme de traitement et d'interprétation d'images Mustic [1] a servi de base pour les développements. Elle a été enrichie de nombreuses fonctionnalités (segmentation, classification, traitements) permettant d'effectuer toutes les étapes de l'interprétation d'une image de télédétection proposées dans cette thèse. Le logiciel a également été modifié pour prendre en compte plusieurs images et effectuer de la classification multisource.

Les développements logiciels concernant le clustering ont été intégrés dans la librairie de classification JCL développée au sein du laboratoire. Des développements ont également été effectués avec la librairie de fouille de données Weka [Hall et al., 2009]. Une nouvelle implémentation de l'approche SAMARAH a notamment été développée à partir de cette librairie. Ceci nous permet de bénéficier des algorithmes de clustering présents et futurs ajoutés à Weka par la communauté.

7.3 Perspectives

Les travaux de recherche présentés dans cette thèse ont ouvert de nombreuses pistes de recherche.

En premier lieu, les travaux effectués sur la résolution de conflits apportent de nouvelles perspectives en terme de stratégie de résolution des désaccords de classification. De nombreuses autres méthodes peuvent être envisagées et certaines d'entre-elles sont d'ores et déjà à l'étude (ordonnancement des conflits, nouveaux opérateurs de résolution, etc.). Un deuxième axe de recherche est l'étude plus précise de l'influence des différents types de méthodes utilisées en clustering collabo-

[1] `https://lsiit.u-strasbg.fr/fdbt-fr/index.php/Logiciels`

ratif. En effet, les travaux effectués dans cette thèse se sont principalement intéressés à l'algorithme KMEANS. Cependant, comme présenté dans le Chapitre 3, d'autres algorithmes peuvent être utilisés. Ainsi, des travaux en cours étudient plus en détail l'influence du choix des méthodes et des paramètres du processus collaboratif. À plus long terme, il serait intéressant d'envisager le portage de l'approche collaborative sous une forme distribuée (*cloud computing*). En effet, l'architecture de la méthode collaborative étant bien adaptée au calcul distribué, une telle approche permettrait un passage à l'échelle sur des données volumineuses.

Les travaux sur l'intégration de connaissances ont permis de montrer que l'utilisation de données étiquetées permettait d'améliorer la qualité de la collaboration en clustering collaboratif. Cependant, d'autres types de connaissances (par exemple des contraintes) seraient à étudier plus en détail. De plus, une réflexion sur la qualité des connaissances a également été engagée. L'objectif est de pouvoir estimer si des connaissances sont de bonne qualité et sont potentiellement intéressantes. Le verrou scientifique important qui reste à lever est la définition de critères d'évaluation de la qualité de ces connaissances. Une réflexion plus générale serait ainsi à envisager sur le rôle des connaissances du domaine dans le processus global d'extraction des connaissances. Ce type d'intégration est de plus en plus courant, mais peu d'études présentent des avancées facilement généralisables. Cette thèse s'est consacrée à l'intégration de connaissances lors de l'étape de clustering mais d'autres étapes du processus de fouille de données pourraient tirer parti de ces connaissances (nettoyage des données, sélection d'attributs, etc.).

D'autres pistes de recherche concernent également les approches d'acquisition active (*active learning*) des connaissances, qui seraient également à étudier pour permettre une interaction plus importante avec l'expert. Il est en effet intéressant d'envisager le développement d'une plate-forme de clustering collaboratif faisant intervenir l'expert pendant le processus de collaboration. Celui-ci pourrait alors être sollicité pour guider le processus et donner des retours sur les modifications entreprise par la méthode.

Parallèlement, la modélisation et l'utilisation de connaissances dans les processus d'interprétation des images de télédétection ont également ouvert de nombreux axes de recherche. Ce domaine est actuellement très actif et trop peu de résultats sont diffusés à l'heure actuelle. L'utilisation de connaissances spatiales est notamment à l'étude pour améliorer la détection et l'identification des objets géographiques. Des travaux ont débuté récemment au sein de notre équipe de recherche pour représenter ces connaissances sous la forme de graphes. Il serait intéressant, à plus long terme, d'étudier comment ces contraintes pourraient être intégrées dans un processus de clustering collaboratif.

L'utilisation de données multisources est un domaine en plein développement et de nombreuses approches ont été proposées récemment. Les méthodes proposées lors de cette thèse ont illustré l'utilisation conjointe de plusieurs données en clustering collaboratif. La méthode de clustering collaboratif utilisant des données multisources sous la forme d'images a été poursuivie par une thèse débutée en octobre 2009 [Kurtz et al., 2010]. La problématique de la segmentation conjointe des images est à l'étude. Enfin, l'utilisation de données issues de simulation à partir de librairies spectrales est actuellement poursuivie au sein du CNES en vue d'une intégration dans l'ORFEO ToolBox [2].

En conclusion, si cette thèse a levé un certain nombre de verrous et a ouvert plusieurs pistes de recherche, il reste à notre avis de nombreux travaux avant d'arriver à une véritable méthode de classification multisource capable à la fois de traiter toutes les données disponibles (quelque soit leur format) et à la fois d'utiliser à bon escient toutes les connaissances disponibles via les experts, et donc d'envisager une méthode de classification réellement multisource enrichie par des connaissances du domaine.

[2] `http://www.orfeo-toolbox.org/otb/`

Annexes

Annexe A

Principales méthodes de clustering

Dans cette Annexe, nous présentons les méthodes les plus classiques en clustering et qui ont été les plus utilisées au cours de cette thèse.

A.1 L'algorithme KMeans

La méthode KMEANS [Macqueen, 1967] cherche un partitionnement des données en se basant sur les barycentres des clusters en minimisant la somme de l'erreur au carré (SSE). L'algorithme est itératif et consiste en la répétition de deux étapes :

1. redéfinition des centres des clusters : le centre μ_i est défini comme l'isobarycentre des objets du i-ième cluster ;
2. affectation des objets aux clusters : un objet x est affecté au cluster dont le centre μ est le plus proche selon une distance donnée.

À l'issue de ces deux étapes la fonction objective est localement optimisée. Les centres sont généralement initialisés aléatoirement, par exemple en prenant des objets aléatoirement dans X. La méthode KMEANS est très populaire par le fait de sa simplicité et son coût algorithmique relativement faible. La méthode souffre cependant de quelques limitations, comme par exemple le nombre de clusters à fournir en entrée, ou la forme hypersphérique des clusters en sortie. De plus, l'algorithme est très sensible aux choix initiaux des centres. En effet, l'algorithme ne converge que vers un minimum local, et en fonction de l'initialisation les résultats peuvent être très différents.

La version floue de KMEANS est appelée FUZZY-C-MEANS [Dunn, 1974]. Elle consiste à considérer une appartenance floue à chaque cluster plutôt qu'une appartenance dure comme dans KMEANS. De cette manière, un degré d'appartenance de chaque objet à chaque cluster est calculé en fonction de la distance de cet objet aux prototypes des différents clusters. Le degré d'appartenance μ d'un objet x au cluster C_i est calculé comme :

$$\mu_i(x) = \frac{1}{\sum_{j=1}^{K} \left(\frac{d(x, \mu_i)}{d(x, \mu_j)} \right)^{\frac{2}{f-1}}}$$

où f est un paramètre qui règle le degré de flou, et d une distance entre objets.

Ce type d'approche floue peut être intéressante notamment dans le cas de données bruitées ou lors de la présence de clusters peu séparables dans l'espace des données. Il est toujours possible d'obtenir une partition dure à l'issue de l'algorithme FUZZY-C-MEANS en considérant pour chaque objet le cluster dont l'appartenance est maximale ($\arg\max_i \mu_i(x)$).

A.2 L'algorithme SOM

La méthode des cartes auto-organisatrices de Kohonen (SOM : *Self-Organizing Maps*) [Kohonen, 1984] est un algorithme de clustering non supervisé basé sur la simulation du fonctionnement des neurones. Un ensemble de neurones virtuels sont reliés entre eux et forme un réseau. Chaque neurone possède des coordonnées dans l'espace des données et représente un prototype d'un cluster. Deux neurones connectés vont s'influencer au cours de l'apprentissage de la topologie du réseau. Une grille est généralement utilisée pour déterminer l'inter-influence des neurones $N_h (h = 1 \leq h \leq H)$ dans le réseau.

Les objets sont présentés un par un aux neurones. À chaque objet présenté x, le neurone vainqueur N_v est déterminé par $N_v = \arg\min_{N_h} d(x, N_h)$. Chaque neurone N_h est alors modifié :

$$N_h = N_h + \gamma(t)\,\Gamma(N_h, N_v, t)\,(x - N_h)$$

où la fonction $\gamma(t)$ est appelée taux d'apprentissage, définie par $\gamma(t) = \gamma_{max}\left(\frac{\gamma_{min}}{\gamma_{max}}\right)^{\frac{t}{t_{max}}}$, $\Gamma(N_h, N_v, t) = e^{\frac{-d^G(N_h, N_v)^2}{2\sigma(t)^2}}$, $\sigma(t) = \sigma_{max}\left(\frac{\sigma_{min}}{\sigma_{max}}\right)^{\frac{t}{t_{max}}}$, $d^G(N_h, N_{h'})$ est la distance de Manhattan sur la grille entre les neurones N_h et $N_{h'}$, t est initialisé à 0 et est incrémenté de 1 à chaque objet présenté.

L'algorithme SOM permet de découvrir la topologie des données, les neurones se stabilisant là où la densité des objets est la plus importante. Plusieurs neurones proches les uns des autres dans la topologie peuvent correspondre à un même cluster. De nombreuses recherches ont été menées et sont en cours pour déterminer la meilleure topologie et comment la choisir.

A.3 L'algorithme EM

L'algorithme EM [Dempster et al., 1977] est un un algorithme de classification probabiliste, c'est-à-dire que l'algorithme cherche les paramètres $\Phi = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$ d'une loi de mélange. L'algorithme EM cherche à maximiser la log-vraisemblance. Chaque itération de l'algorithme se décompose en deux phases, une phase E (*Expectation*) et une phase M (*Maximization*).

La phase E consiste à évaluer $P(x, \Phi)$ pour chaque objet x en fonction des paramètres du modèle, ainsi que la probabilité d'appartenance de x à chacune des classes (assimilable à un degré d'appartenance) définie par :

$$t_i(x) = \frac{\pi_i \times P(x, \theta_i)}{\sum_{j=1}^{K} \pi_j \times P(x, \theta_j)}$$

La phase M consiste à estimer, en fonction des probabilités d'appartenance des objets aux classes, les paramètres du modèle qui maximisent la log-vraisemblance des objets. La définition des paramètres θ_i va dépendre du type de distribution de la classe C_i. La définition des paramètres π_i se fait par :

$$\pi_i = \frac{1}{N} \sum_{x \in X} t_i(x)$$

On suppose généralement que chaque classe suit une loi normale multidimensionnelle. Dans [Candillier et al., 2005], les auteurs utilisent un modèle simplifié en considérant les attributs indépendants les uns des autres. Dans ce cas, on peut définir $P(x, \theta_i) = \prod_{j=1}^{n} P(x_j, \theta_{i,j})$. Les

paramètres du modèle sont alors simplement les moyennes et écarts-types de chacun des attributs numériques et la fréquence des différentes modalités des attributs catégoriels.

A.4 L'algorithme COBWEB

L'algorithme COBWEB [Fisher, 1987] est un algorithme de classification incrémental, hiérarchique, de formation de concepts. Les objets sont présentés un par un à la hiérarchie de concepts. Un objet x présenté à la hiérarchie sera inséré dans le concept racine, puis récursivement dans le concept le plus adapté, en modifiant l'arbre par différents opérateurs. Si le concept courant c est une feuille, l'objet est intégré au concept s'il correspond suffisamment. S'il ne correspond pas, un nouveau concept est créé. Si c n'est pas une feuille, différents opérateurs sont alors testés sur le concept c pour voir si un nouveau concept est crée. L'algorithme utilise l'indice de prédictivité pour faire ce choix. Le meilleur résultat est appliqué et l'algorithme se poursuit récursivement avec le concept où x a été intégré.

Annexe B

Évaluation de clustering par critères externes

Il est possible d'évaluer la qualité d'un résultat de clustering pour lequel la classe réelle de chaque objet est connue, en comparant le partitionnement obtenu avec la classification réelle. Ce type d'évaluation est appelé évaluation par critères externes. Nous présentons dans cette annexe les critères classiques utilisés lors de cette thèse pour évaluer la pertinence des algorithmes proposés. Les critères consisteront en la comparaison de deux partitions $\mathcal{C}^{(1)}$ et $\mathcal{C}^{(2)}$.

B.1 Critères d'évaluations

Comme il n'est pas possible de faire une correspondance entre les clusters/classes des deux résultats (voir Chapitre 4), les critères ne sont pas évalués directement sur les classes des objets, mais sur des paires d'objets : dans une classification dure, deux objets peuvent appartenir à la même classe ou à deux classes différentes.

On notera $\mathcal{P}_2(X)$ l'ensemble des paires de X, c'est-à-dire l'ensemble des sous-ensembles de deux éléments de X. Soit $M = |\mathcal{P}_2(X)| = \frac{1}{2} \times N \times (N-1)$ le nombre de paires d'objets.

On note alors mm le nombre de paires d'objets qui sont dans la même classe dans $\mathcal{C}^{(1)}$ et dans $\mathcal{C}^{(2)}$, dd le nombre de paires d'objets qui sont dans deux classes différentes dans $\mathcal{C}^{(1)}$ et dans $\mathcal{C}^{(2)}$, md le nombre de paires d'objets qui sont dans la même classe dans $\mathcal{C}^{(1)}$ mais dans deux classes différentes dans $\mathcal{C}^{(2)}$ et dm le nombre de paires d'objets qui sont dans deux classes différentes dans $\mathcal{C}^{(1)}$ mais dans la même classe dans $\mathcal{C}^{(2)}$. On a $mm + dd + md + dm = M$.

Plus mm et dd sont élevés et md et dm sont bas, plus on considère que les partitions sont similaires.

Plusieurs critères ont été définis comme des combinaisons de ces différentes valeurs. Sur chacun de ces critères une valeur forte indique une forte ressemblance entre les partitions, et une valeur faible indique une faible ressemblance.

Précision :

La *précision* est la probabilité pour que deux objets soient dans la même classe dans $\mathcal{C}^{(2)}$ s'il le sont dans $\mathcal{C}^{(1)}$:

$$\text{Précision} = \frac{mm}{mm + md} \tag{B.1}$$

La précision prend ses valeurs sur $[0; 1]$. Mais une valeur de 1 ne garantit pas que les deux

classifications sont identiques. Ce critère suppose que $\mathcal{C}^{(2)}$ représente la classification réelle des données.

Rappel :

Le *rappel* est la probabilité pour que deux objets soient dans la même classe dans $\mathcal{C}^{(1)}$ s'ils le sont dans $\mathcal{C}^{(2)}$:

$$\text{Rappel} = \frac{mm}{mm + dm} \tag{B.2}$$

Le rappel prend ses valeurs sur $[0; 1]$. Mais une valeur de 1 ne garantie pas que les deux classifications sont identiques. Ce critère suppose que $\mathcal{C}^{(2)}$ représente la classification réelle des données.

Indice de Rand :

L'*indice de Rand* est défini par :

$$\text{Rand} = \frac{mm + dd}{M} \tag{B.3}$$

L'indice de Rand prend ses valeurs sur $[0; 1]$. Il vaut 1 si et seulement si les deux classifications sont identiques.

Indice de Jaccard :

L'*indice de Jaccard* est défini par :

$$\text{Jaccard} = \frac{mm}{mm + md + dm} \tag{B.4}$$

L'indice de Jaccard prend ses valeurs sur $[0; 1]$. Il vaut 1 si et seulement si les deux classifications sont identiques.

Indice de Folkes et Mallows :

L'*indice de Folkes et Mallows* est défini par la moyenne géométrique entre la précision et le rappel :

$$\text{Folkes-Mallows} = \sqrt{prec \times rapp} \tag{B.5}$$

L'indice de Folkes et Mallows prend ses valeurs sur $[0; 1]$. Il vaut 1 si et seulement si la précision et le rappel valent 1 tous les deux, et donc si et seulement si les deux classifications sont identiques.

F-mesure :

La *F-mesure* est définie par :

$$\text{F-mesure} = \frac{2 \times prec \times rapp}{prec + rapp} \tag{B.6}$$

La F-mesure prend ses valeurs sur $[0; 1]$. Elle vaut 1 si et seulement si la précision et le rappel valent 1 tous les deux, et donc si et seulement si les deux classifications sont identiques.

Indice Kappa (κ) :

L'indice Kappa (κ) est défini par :

$$\text{Kappa} = \frac{P_0 - P_e}{1 - P_e} \tag{B.7}$$

avec $P_0 = \frac{mm+dd}{M}$ et$P_e = \frac{1}{M^2} \times (mm + md) \times (mm + dm) + (md + dd) \times (dm + dd)$.

L'indice Kappa (κ) prend ses valeurs sur $[-1; 1]$. Une forte valeur sur le coefficient κ indique une forte similarité entre les classification.

Bibliographie

[**Adryan et Schuh, 2004**] B. Adryan et R. Schuh. Gene-ontology-based clustering of gene expression data. *Bioinformatics*, 20(16) :2851–2852, 2004. ISSN 1367-4803.

[**Aggarwal et al., 1999**] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, et J. S. Park. Fast algorithms for projected clustering. *SIGMOD Record*, 28(2) :61–72, 1999.

[**Aggarwal et Yu, 2008**] C. C. Aggarwal et P. S. Yu. *Privacy-Preserving Data Mining Models and Algorithms*, volume 34 of *Advances in Database Systems*. Springer, 2008.

[**Agrawal et Srikant, 2000**] R. Agrawal et R. Srikant. Privacy-preserving data mining. *SIGMOD Record*, 29(2) :439–450, 2000. ISSN 0163-5808.

[**Ajmera et al., 2002**] J. Ajmera, H. Bourlard, I. Lapidot, et I. McCowan. Unknown-multiple speaker clustering using hmm. In *International Conference on Spoken Language Processing*, pages 573–576, September 2002.

[**Alboody et al., 2009**] A. Alboody, F. Sedes, et J. Inglada. Multi-level topological relations of the spatial reasoning system rcc-8. In *International Conference on Advances in Databases, Knowledge, and Data Applications*, pages 13 –21, 1-6 2009.

[**Anand et al., 1995**] S. S. Anand, D. A. Bell, et J. G. Hughes. The role of domain knowledge in data mining. In *CIKM '95 : Proceedings of the fourth international conference on Information and knowledge management*, pages 37–43, New York, NY, USA, 1995. ACM. ISBN 0-89791-812-6.

[**Asuncion et Newman, 2007**] A. Asuncion et D. Newman. Uci machine learning repository, 2007.

[**Athanasiadias et al., 2007**] T. Athanasiadias, P. Mylonas, et Y. Avrithis. Semantic image segmentation and object labeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(3) :298–312, 2007.

[**Ayad et Kamel, 2008**] H. Ayad et M. S. Kamel. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1) :160–173, 2008.

[**Bae et Bailey, 2006**] E. Bae et J. Bailey. Coala : A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. *Data Mining, IEEE International Conference on*, 0 :53–62, 2006. ISSN 1550-4786.

[**Baldridge et al., 2008**] A. M. Baldridge, S. J. Hook, C. I. Grove, et R. g. The aster spectral library version 2.0. *Remote Sensing of Environment*, 2008.

[**Ball et Hall, 1965**] G. Ball et D. Hall. Isodata, a novel method of data anlysis and pattern classification. Technical report, Stanford Research Institute, Stanford, CA, 1965.

[**Basu et al., 2002**] S. Basu, A. Banerjee, et R. J. Mooney. Semi-supervised clustering by seeding. In *International Conference on Machine Learning*, pages 19–26, 2002.

[**Basu et al., 2004a**] S. Basu, A. Banerjee, et R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *SIAM International Conference on Data Mining*, pages 333–344, 2004a.

[**Basu et al., 2004b**] S. Basu, M. Bilenko, et R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *International Conference on Knowledge Discovery and Data Mining*, pages 59–68, 2004b.

[**Bellman, 1961**] R. Bellman. *Adaptive Control Processes*. 1961.

[**Benediktsson et Kanellopoulos, 1999**] J. Benediktsson et I. Kanellopoulos. Classification of multisource and hyperspectral data based on decision fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3) :1367–1377, 1999.

[**Berkhin, 2002**] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[**Bilenko et al., 2004**] M. Bilenko, S. Basu, et R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *International Conference on Machine Learning*, pages 81–88, 2004.

[**Blansché, 2006**] A. Blansché. *Sélection automatique d'attributs et combinaison de classifieurs pour des données complexes*. PhD thesis, Université Louis Pasteur, Strasbourg, 2006.

[**Bohm et Plant, 2008**] C. Bohm et C. Plant. Hissclu : A hierarchical density-based method for semi-supervised clustering. 2008.

[**Bonn et Rochon, 1992**] F. Bonn et G. Rochon. *Précis de télédétection*, volume 1. Presses de l'Université du Québec, 1992.

[**Bouchachia et Pedrycz, 2006**] A. Bouchachia et W. Pedrycz. Data clustering with partial supervision. *Data Mining and Knowledge Discovery*, 12(1) :47–78, 2006.

[**Bradley et al., 2000**] P. S. Bradley, K. P. Bennett, et A. Demiriz. Constrained k-means clustering. Technical report, MSR-TR-2000-65, Microsoft Research, 2000.

[**Breaux et Reed, 2005**] T. D. Breaux et J. W. Reed. Using ontology in hierarchical information clustering. *Hawaii International Conference on System Sciences*, 4 :111b, 2005. ISSN 1530-1605.

[**Breiman, 1996**] L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2) :123–140, 1996. ISSN 0885-6125.

[**Brut et al., 2008**] M. Brut, F. Sedes, R. Grigoras, et V. Charvillat. An ontology based approach for providing multimedia personalised recommendations. *International Journal Web Grid Services*, 4(3) :314–329, 2008. ISSN 1741-1106.

[**Bruzzone et al., 2002**] L. Bruzzone, R. Cossu, et G. Vernazza. Combining parametric and non-parametric algorithms for a partially unsupervised classification of multitemporal remote-sensing images. *Image Fusion*, 3 :289–297, 2002.

[**Candillier et al., 2005**] L. Candillier, I. Tellier, F. Torre, et O. Bousquet. SSC : Statistical Subspace Clustering. In S. Pinson et N. Vincent, editors, *Extraction et Gestion des Connaissances (EGC)*, volume 1, pages 177–182, 2005.

[**Candillier et al., 2006**] L. Candillier, I. Tellier, F. Torre, et O. Bousquet. Cascade evaluation of clustering algorithms. In *European Conference on Machine Learning*, volume 4212/2006, pages 574–581. Springer Berlin / Heidelberg, 2006.

[**Cardoso et Corte-Real, 2005**] J. S. Cardoso et L. Corte-Real. Toward a generic evaluation of image segmentation. *IEEE Transactions on Image Processing*, 14(11) :1773–1782, 2005.

[**Chabrier et al., 2006**] S. Chabrier, B. Emile, C. Rosenberger, et H. Laurent. Unsupervised performance evaluation of image segmentation. *EURASIP Journal on Applied Signal Processing*, pages 217–217, 2006. ISSN 1110-8657.

[**Chang et al., 2007**] Y.-L. Chang, L.-S. Liang, C.-C. Han, J.-P. Fang, L. W.-Y., et C. K.-S. Multisource data fusion for landslide classification using generalized positive boolean functions. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6), 2007.

[**Chibani, 2005**] Y. Chibani. Selective synthetic aperture radar and panchromatic image fusion by using the a wavelet decomposition. *EURASIP Journal on Applied Signal Processing*, (14) : 2207–2214, 2005.

[**Clark et al., 2002**] R. Clark, G. A. Swayze, K. Livo, R. F. Kokaly, T. V. V. King, J. B. Dalton, J. S. Vance, B. W. Rockwell, T. Hoefen, et R. R. McDougal. Synthesis of multispectral bands from hyperspectral data : Validation based on images acquired by aviris, hyperion, ali, and etm+. 2002.

[**Cleuziou G., 2009**] M. L. . S. J.-H. Cleuziou G., Exbrayat M. Cofkm : A centralized method for multiple-view clustering. In *IEEE International Conference on Data Mining*, pages 752–757, 2009.

[**Clifton, 2001**] C. Clifton. Privacy preserving distributed data mining. *ACM SIGKDD Explorations*, 4, 2001.

[**Cohn et al., 2003**] D. Cohn, R. Caruana, et A. Mccallum. Semi-supervised clustering with user feedback. Technical report, Cornell University, 2003.

[**Congalton, 1991**] R. G. Congalton. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environnnment*, 37 :35–46, 1991.

[**Cover et Thomas, 2006**] T. M. Cover et J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2 edition, 2006.

[**Crevier et Lepage, 1997**] D. Crevier et R. Lepage. Knowledge-based image understanding systems : a survey. *Computer Vision and Image Understanding*, 67(2) :161–185, 1997.

[**Davidson et Ravi, 2005**] I. Davidson et S. Ravi. Clustering under constraints : Feasibility issues and the k-means algorithm. In *SIAM Data Mining Conference*, 2005.

[**Davidson et al., 2006**] I. Davidson, K. L. Wagstaff, et S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 115–126, 2006.

[**Davies et Bouldin, 1979**] D. L. Davies et D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2) :224–227, 1979.

[**Demiriz et al., 1999**] A. Demiriz, K. Bennett, et M. Embrechts. Semi-supervised clustering using genetic algorithms. pages 809–814, 1999.

[**Dempster et al., 1977**] A. P. Dempster, N. M. Laird, et D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1) :1–38, 1977.

[**Derivaux, 2009**] S. Derivaux. *Construction et classification d'objets à partir d'images de télédétection par une approche itérative guidée par des connaissances du domaine*. PhD thesis, Université de Strasbourg, 2009.

[**Derivaux et al., 2006**] S. Derivaux, S. Lefevre, C. Wemmert, et J. Korczak. Watershed segmentation of remotely sensed images based on a supervised fuzzy pixel classification. In *IEEE International Geosciences And Remote Sensing Symposium (IGARSS)*, pages 3712–3715, 2006.

[**Desjardins, 2000**] R. Desjardins. *La Télédétection - Perspective Analytique*. Editions Estem, 2000.

[**Dhillon, 2001**] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274, 2001.

[**Dimitriadou et al., 2002**] E. Dimitriadou, A. Weingessel, et K. Hornik. A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 16 (7) :901–912, 2002.

[**Dou et al., 2007**] W. Dou, Y. Chen, X. Li, et D. Z. Sui. A general framework for component substitution image fusion : An implementation using the fast image fusion method. *Computers & Geosciences*, 33(2) :219–228, 2007.

[**Draper et al., 1989**] B. Draper, A. Collins, J. Brolio, A. Hanson, et E. Riseman. The schema system. *International Journal of Computer Vision*, 2(3) :209–250, 1989.

[**Duarte et al., 2005**] F. Duarte, A. Fred, A. Lourenco, et M. Rodrigues. Weighting cluster ensembles in evidence accumulation clustering. *Portuguese Conference on Artificial Intelligence*, pages 159–167, 2005.

[**Dubuisson et Jain, 1995**] M. Dubuisson et A. Jain. Contour extraction of moving objects in complex outdoor scenes. *International Journal of Computer Vision*, (14) :83–105, 1995.

[**Dunn, 1974**] J. C. Dunn. Well separated clusters and optimal fuzzy-partitions. *Journal of Cybernetics*, 4 :95–104, 1974.

[**Durand et al., 2007**] N. Durand, S. Derivaux, G. Forestier, C. Wemmert, P. Gancarski, O. Boussïd, et A. Puissant. Ontology-based object recognition for remote sensing image interpretation. In *IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 472–479, Patras, Greece, october 2007. IEEE Computer Society.

[**Eick et al., 2004**] C. F. Eick, N. Zeidat, , et Z. Zhao. Supervised clustering - algorithms and benefits. In *International Conference on Tools with Artificial Intelligence*, pages 774–776, 2004.

[**Ester et al., 1996**] M. Ester, H. Kriegel, J. Sander, et X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Information Retrieval*, pages 226–231, 1996.

[**Faceli et al., 2006**] K. Faceli, A. de Carvalho, et M. de Souto. Multi-objective clustering ensemble. *International Conference on Hybrid Intelligent Systems*, pages 51–51, 2006.

[**Faceli et al., 2007**] K. Faceli, A. C. P. F. de Carvalho, et M. C. P. de Souto. Multi-objective clustering ensemble with prior knowledge. volume 4643, pages 34–45. Springer, 2007.

[**Fern et Brodley, 2004**] X. Z. Fern et C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML '04 : Proceedings of the twenty-first international conference on Machine learning*, page 36, New York, NY, USA, 2004. ACM. ISBN 1-58113-828-5.

[**Figueiredo et Jain, 2000**] M. A. T. Figueiredo et A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 :381–396, 2000.

[**Fisher, 1987**] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2 :139–172, 1987.

[**Forestier et al., 2008a**] G. Forestier, S. Derivaux, C. Wemmert, et P. Gancarski. An evolutionary approach for ontology driven image interpretation. In *Tenth European Workshop on Evolutionary Computation in Image Analysis and Signal Processing*, volume 4974 of *Lecture Notes in Computer Sciences*, pages 295–304, Napoli, Italy, march 2008a. Springer.

[**Forestier et al., 2008b**] G. Forestier, S. Derivaux, C. Wemmert, et P. Gancarski. Interprétation d'images basée sur une approche évolutive guidée par une ontologie. In *Journées Francophones Extraction et Gestion des Connaissances (EGC 2008)*, volume 2, pages 469–474, Sophia Antipolis, France, january 2008b.

[**Forestier et al., 2010a**] G. Forestier, P. Gancarski, et C. Wemmert. Collaborative clustering with background knowledge. *Data & Knowledge Engineering*, 69(2) :211–228, 2010a.

[**Forestier et al., 2009a**] G. Forestier, J. Inglada, C. Wemmert, et P. Gancarski. Mining spectral libraries to study sensors' discrimination ability. In *SPIE Europe Remote Sensing*, volume 7478, page 9 pages, september 2009a.

[**Forestier et al., 2008c**] G. Forestier, C. Wemmert, et P. Gancarski. Multi-source images analysis using collaborative clustering. *EURASIP Journal on Advances in Signal Processing - Special issue on Machine Learning in Image Processing*, 2008, 2008c.

[**Forestier et al., 2008d**] G. Forestier, C. Wemmert, et P. Gancarski. On combining unsupervised classification and ontology knowledge. In *IEEE Geoscience and Remote Sensing Symposium*, volume 4, pages 395–398, Boston, Massachusetts, july 2008d.

[**Forestier et al., 2008e**] G. Forestier, C. Wemmert, et P. Gancarski. Semi-supervised collaborative clustering with partial background knowledge. In *Workshop on Mining Complex Data*, pages 211–217, Pisa, Italy, december 2008e. IEEE International Conference on Data Mining.

[**Forestier et al., 2010b**] G. Forestier, C. Wemmert, et P. Gancarski. Background knowledge integration in clustering using purity indexes. In *International Conference on Knowledge Science, Engineering & Management*, Belfast, UK, 2010b.

[**Forestier et al., 2010c**] G. Forestier, C. Wemmert, et P. Gancarski. Comparaison de critères de pureté pour l'intégration de connaissances en clustering semi-supervisé. In *Journées Francophones Extraction et Gestion des Connaissances (EGC 2010)*, pages 127–132, january 2010c.

[**Forestier et al., 2010d**] G. Forestier, C. Wemmert, et P. Gancarski. Towards conflict resolution in collaborative clustering. In *IEEE International Conference on Intelligent Systems*, London, UK, 2010d.

[**Forestier et al., 2010e**] G. Forestier, C. Wemmert, et P. Gancarski. Étude de données multisources par simulation de capteurs et clustering collaboratif. In *Atelier Fouille de données complexes, Journées Francophones Extraction et Gestion des Connaissances (EGC 2010)*, pages A143–A152, Hammamet, Tunisie, 2010e.

[**Forestier et al., 2009b**] G. Forestier, C. Wemmert, P. Gancarski, et J. Inglada. Mining multiple satellite sensor data using collaborative clustering. In *Workshop on Mining Multiple Information Sources*, pages 501–506. IEEE International Conference on Data Mining, 2009b.

[**Franke et al., 2006**] J. Franke, V. Heinzel, et G. Menz. Assessment of ndvi- differences caused by sensor specific relative spectral response functions. *IEEE International Geoscience and Remote Sensing Symposium*, pages 1138–1141, 2006.

[**Fred et Jain, 2005**] A. Fred et A. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6) :835–850, 2005.

[**Fred, 2001**] A. L. N. Fred. Finding consistent clusters in data partitions. In J. Kittler et F. Roli, editors, *Multiple Classifier Systems*, Lecture Notes in Computer Science, pages 309–318. Springer, 2001.

[**Fritzke, 1995**] B. Fritzke. A growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems*, pages 625–632. MIT Press, 1995.

[**Gahegan et Pike, 2006**] M. Gahegan et W. Pike. A situated representation of geographical information. *Transactions in GIS*, 10(5) :727–749, 2006.

[**Gao et al., 2006**] J. Gao, P. Tan, et H. Cheng. Semi-supervised clustering with partial background information. In *SIAM International Conference on Data Mining*, 2006.

[**Germain et al., 2004**] M. Germain, M. Voorons, J.-M. Boucher, G. B. Bénié, et E. Beaudry. Multisource image fusion algorithm based on a new evidential reasoning approach. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 2004.

[**Goldberg, 1989**] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[**Govender et al., 2007**] M. Govender, K. Chetty, et H. Bulcock. A review of hyperspectral remote sensing and its application in vegetation and water resource studies. *Water SA*, 33(2) : 145–152, 2007.

[**Grira et al., 2008**] N. Grira, M. Crucianu, et N. Boujemaa. Active semi-supervised fuzzy clustering. *Pattern Recognition*, 41(5) :1851–1861, 2008. ISSN 0031-3203.

[**Grozavu et Bennani, 2010**] N. Grozavu et Y. Bennani. Classification collaborative non supervisée. In *Conférence francophone sur l'apprentissage automatique (CAP)*, 2010.

[**Gruber, 1995**] T. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43(5/6) :907–928, 1995.

[**Hadjitodorov et Kuncheva, 2007**] S. T. Hadjitodorov et L. I. Kuncheva. Selecting diversifying heuristics for cluster ensembles. In M. Haindl, J. Kittler, et F. Roli, editors, *Multiple Classifier Systems*, volume 4472 of *Lecture Notes in Computer Science*, pages 200–209. Springer, 2007.

[**Hadjitodorov et al., 2006**] S. T. Hadjitodorov, L. I. Kuncheva, et L. P. Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3) :264–275, 2006.

[**Halkidi et al., **] M. Halkidi, Y. Batistakis, et M. Vazirgiannis. Clustering validity checking methods. *SIGMOD Record*, 31(3) :19–27.

[**Hall et al., 2009**] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, et I. H. Witten. The weka data mining software : an update. *SIGKDD Explorations Newsletter*, 11(1) :10–18, 2009.

[**Han, 2005**] J. Han. *Data Mining : Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[**Han et al., 1999**] J. Han, L. Lakshmanan, et R. Ng. Constraint-based, multidimensional data mining. *Computer*, 32(8) :46–50, Aug 1999.

[**Handl et Knowles, 2007**] J. Handl et J. Knowles. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 11(1) :56–76, 2007.

[**Handl et Knowles, 2006**] J. Handl et J. D. Knowles. On semi-supervised clustering via multiobjective optimization. In *Genetic and Evolutionary Computation Conference*, pages 1465–1472, 2006.

[**Hansen et Yu, 1998**] M. H. Hansen et B. Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96 :746–774, 1998.

[**He et al., 2005**] Z. He, X. Xu, et S. Deng. A cluster ensemble method for clustering categorical data. *Information Fusion*, 6 :143–151, 2005.

[**Heidena et al., 2007**] U. Heidena, K. Segl, S. Roessner, et H. Kaufmann. Determination of robust spectral features for identification of urban surface materials in hyperspectral remote sensing data. *Remote Sensing of Environment*, 111 :537–552, 2007.

[**Herold et al., 2003**] M. Herold, M. Gardner, et D. Roberts. Spectral resolution requirements for mapping urban areas. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(9) :1907–1919, 2003.

[**Hotho et al., 2001**] A. Hotho, A. Maedche, , et S. Staab. Ontology-based text clustering. 2001.

[**Hu et al., 2006**] T. Hu, Y. Yu, J. Xiong, et S. Y. Sung. Maximum likelihood combination of multiple clusterings. *Pattern Recognition Letters*, 27(13) :1457–1464, 2006.

[**Huang et Lam, 2009**] R. Huang et W. Lam. An active learning framework for semi-supervised document clustering with language modeling. *Data & Knowledge Engineering*, 68(1) :49–67, 2009.

[**Hughes, 1968**] G. F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Informations Theory*, 14(1) :55–63, 1968.

[**Ioannidou, 2007**] V. K. P. K. S. Ioannidou. A comparison study on fusion methods using evaluation indicators. *International Journal of Remote Sensing*, 28 :2309–2341, 2007.

[**Jain et al., 2004**] A. Jain, A. Topchy, M. Law, et J. Buhmann. Landscape of clustering algorithms. In *International Conference on Pattern Recognition*, volume 1, pages 260–263, 2004.

[**Jain, 2009**] A. K. Jain. Data clustering : 50 years beyond k-means. *Pattern Recognition Letters*, 31(8) :651–666, 2009.

[**Jain et Dubes, 1988**] A. K. Jain et R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X.

[**Jain et al., 1999**] A. K. Jain, M. N. Murty, et P. J. Flynn. Data clustering : a review. *ACM Computing Surveys*, 31(3) :264–323, 1999.

[**Janssen et Molenaar, 1995**] L. Janssen et M. Molenaar. Terrain objects, their dynamics and their monitoring by the integration of gis and remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 33 :749–758, 1995.

[**Jia et Richards, 2005**] X. Jia et J. Richards. Fast k-nn classification using the cluster-space approach. *Geoscience and Remote Sensing Letters, IEEE*, 2(2) :225–228, 2005.

[**Karypis et al., 1997**] G. Karypis, R. Aggarwal, V. Kumar, et S. Shekhar. Multilevel hypergraph partitioning : Application in vlsi domain. In *ACM IEEE Design Automation Conference*, pages 526–529, 1997.

[**Kaufman et Rousseeuw, 1990**] L. Kaufman et P. Rousseeuw. *Finding Groups in Data An Introduction to Cluster Analysis*. Wiley Interscience, New York, 1990.

[**Kavzoglu, 2004**] T. Kavzoglu. Simulating landsat etm+ imagery using dais 7915 hyperspectral scanner data. *International journal of remote sensing*, 25(22) :5049–5067, 2004.

[**Kittler et al., 1998**] J. Kittler, M. Hatef, R. P. W. Duin, et J. Matas. On combining classifiers. *IEEE Transaction Pattern Analysis Machine Intelligence*, 20(3) :226–239, 1998.

[**Klein et al., 2002**] D. Klein, S. Kamvar, et C. Manning. From instance-level constraints to space-level constraints : Making the most of prior knowledge in data clustering. In *In The Nineteenth International Conference on Machine Learning*, pages 307–314, 2002.

[**Kleinberg, 2002**] J. Kleinberg. An impossibility theorem for clustering. In *Neural Information Processing Systems*, pages 446–453. MIT Press, 2002.

[**Kohonen, 1984**] T. Kohonen. *Self-Organization and Associative Memory*. Springer Series in Information Sciences. Springer, 1984.

[**Konak et al., 2006**] A. Konak, D. Coit, et A. Smith. Multi-objective optimization using genetic algorithms : A tutorial. *Reliability Engineering & System Safety*, 91(9) :992–1007, 2006.

[**Kopanas et al., 2002**] I. Kopanas, N. M. Avouris, et S. Daskalaki. The role of domain knowledge in a large scale data mining project. In *Methods and Applications of Artificial Intelligence : Lecture Notes in Artificial Intelligence*, pages 288–299. Springer-Verlag, 2002.

[**Kreiger et Green, 1999**] A. Kreiger et P. Green. A generalized rand-index method for consensus clustering of separate partitions of the same data base. *Journal of Classification*, 16 :63–89, 1999.

[**Kropatsch, 1990**] W. Kropatsch. Integration of sar and dem data - geometrical considerations. *Proc. Workshop Multisource Data Integration in Remote Sensing*, pages 27–38, 1990.

[**Kumar et Kummamuru, 2008**] N. Kumar et K. Kummamuru. Semisupervised clustering with metric learning using relative comparisons. *IEEE Transactions on Knowledge and Data Engineering*, 20(4) :496–503, 2008. ISSN 1041-4347.

[**Kuncheva, 2008**] L. I. Kuncheva. Classifier ensembles : Facts, fiction, faults and future, 2008.

[**Kurtz et al., 2010**] C. Kurtz, N. Passat, P. Gancarski, et A. Puissant. Multiresolution region-based clustering for urban analysis. *International Journal of Remote Sensing*, 2010.

[**Lange et al., 2004**] T. Lange, V. Roth, M. L. Braun, et J. M. Buhmann. Stability-based validation of clustering solutions. *Neural Computing*, 16(6) :1299–1323, 2004. ISSN 0899-7667.

[**Law et al., 2004**] M. Law, A. Topchy, et A. Jain. Multiobjective data clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 424–430, 2004.

[**Liu et al., 1994**] S. Liu, M. Thonnat, et M. Berthod. Automatic classification of planktonic foraminifera by a knowledge-based system. In *Conference on Artificial Intelligence for Applications*, pages 358–364. IEEE Computer Society, 1994.

[**Loia et al., 2007**] V. Loia, W. Pedrycz, et S. Senatore. Semantic web content analysis : A study in proximity-based collaborative clustering. *Fuzzy Systems, IEEE Transactions on*, 15(6) : 1294–1312, 2007.

[**Long et al., 2005**] B. Long, Z. M. Zhang, et P. S. Yu. Combining multiple clusterings by soft correspondence. In *International Conference on Data Mining*, pages 282–289. IEEE Computer Society, 2005.

[**Léo Provencher, 2007**] J.-M. M. D. Léo Provencher. *Précis de télédetection : Méthodes de photo-interprétation et d'interprétation d'image*. Presses de l'Université du Québec, 2007.

[**Macqueen, 1967**] J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[**Maillot et Thonnat, 2008**] N. Maillot et M. Thonnat. Ontology based complex object recognition. *Image and Vision Computing*, 26(1) :102–113, 2008.

[**Manning et al., 2008**] C. D. Manning, P. Raghavan, et H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[**Marino Drews, 1993**] O. Marino Drews. *Raisonnement classificatoire dans une représentation à objets multi-points de vue*. Thèse de doctorat, Université J. Fourier, Grenoble, 1993.

[**Mark et al., 1999**] D. M. Mark, B. Smith, et B. Tversky. Ontology and geographic objects : An empirical study of cognitive categorization. In *International Conference on Spatial Information Theory : Cognitive and Computational Foundations of Geographic Information Science*, pages 283–298, London, UK, 1999. Springer-Verlag.

[**Martin et al., 2010**] L. Martin, G. Cleuziou, M. Exbrayat, et F. Moal. Intégration interactive de contraintes pour la réduction de dimensions et la visualisation. In *Journées Francophones Extraction et Gestion des Connaissances (EGC 2010)*, pages 369–380, 2010.

[**Matsuyama et Hwang, 1990**] T. Matsuyama et V.-S. Hwang. *SIGMA - A Knowledge-Based Aerial Image Understanding System*. Plenum Press New York USA, 1990.

[**Megiddo et Supowit, 1984**] N. Megiddo et K. J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1) :182–196, 1984.

[**Merugu et Ghosh, 2003**] S. Merugu et J. Ghosh. Privacy-preserving distributed clustering using generative models. In *IEEE International Conference on Data Mining*, pages 211–218. IEEE Computer Society, 2003.

[**Meyer et Chander, 2007**] D. Meyer et G. Chander. The effect of variations in relative spectral response on the retrieval of land surface parameters from multiple sources of remotely sensed imagery. *IEEE International Geoscience and Remote Sensing Symposium*, pages 5150–5153, 2007.

[**Minaei-Bidgoli et al., 2004a**] B. Minaei-Bidgoli, A. Topchy, et W. Punch. Ensembles of partitions via data resampling. *International Conference on Information Technology : Coding and Computing*, 2 :188–192, April 2004a.

[**Minaei-Bidgoli et al., 2004b**] B. Minaei-Bidgoli, A. P. Topchy, et W. F. Punch. A comparison of resampling methods for clustering ensembles. In *International Conference on Artificial Intelligence*, pages 939–945, 2004b.

[**Minsky, 1975**] M. Minsky. A framework for representing knowledge. In *The Psychology of Computer Vision*, pages 211–281. McGraw-Hill, 1975.

[**Mitra et al., 2006**] H. Mitra, Banka, et W. Pedrycz. Rough-fuzzy collaborative clustering. *IEEE Transactions on Systems, Man, and Cybernetics*, 36 :795–805, 2006.

[**Nagesh et al., 2000**] H. S. Nagesh, S. Goil, et A. Choudhary. A scalable parallel subspace clustering algorithm for massive data sets. In *International Conference on Parallel Processing*, pages 477–484, 2000.

[**Nguyen et Caruana, 2007**] N. Nguyen et R. Caruana. Consensus clusterings. In *International Conference on Data Mining*, pages 607–612. IEEE Computer Society, 2007.

[**Ogiela et Tadeusiewicz, 2008**] M. R. Ogiela et R. Tadeusiewicz. *Modern Computational Intelligence Methods for the Interpretation of Medical Images*. Springer, 2008.

[**Oliveira et Pedrycz, 2007**] J. V. d. Oliveira et W. Pedrycz. *Advances in Fuzzy Clustering and its Applications*. John Wiley & Sons, Inc., New York, NY, USA, 2007.

[**Pakhira et al., 2004**] M. K. Pakhira, S. Bandyopadhyay, et U. Maulik. Validity index for crisp and fuzzy clusters. *Pattern Recognition*, 37(3) :487 – 501, 2004.

[**Panagi et al., 2006**] P. Panagi, S. Dasiopoulou, G. T. Papadopoulos, I. Kompatsiaris, et M. G. Strintzis. A genetic algorithm approach to ontology-driven semantic image analysis. In *IEEE International Conference of Visual Information Engineering (VIE 2006)*, pages 132–137. IEEE Computer Society, 2006.

[**Pearson, 1900**] K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, 1900.

[**Pedrycz, 2002**] W. Pedrycz. Collaborative fuzzy clustering. *Pattern Recognition Letters*, 23 : 1675–1686, 2002.

[**Pedrycz, 2004**] W. Pedrycz. Fuzzy clustering with a knowledge-based guidance. *Pattern Recognition Letters*, 25(4) :469–480, 2004.

[**Pedrycz et Rai, 2008**] W. Pedrycz et P. Rai. A multifaceted perspective at data analysis : A study in collaborative intelligent agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 38(4) :1062–1072, 2008.

[**Pohl et Genderen, 1998**] C. Pohl et J. L. V. Genderen. Multisensor image fusion in remote sensing : concepts, methods and applications. *International Journal on Remote Sensing*, 19(5) : 823–854, 1998.

[**Puissant et al., 2003**] A. Puissant, T. Ranchin, C. Weber, et A. Serradj. Fusion of quickbird ms and pan data for urban studies. In *European Association of Remote Sensing Laboratories Symposium (EARSeL)*, pages 77–83, Gent, Belgium, June 2003.

[**Punera et Ghosh, 2007**] K. Punera et J. Ghosh. *Advances in Fuzzy Clustering and its Applications*. Wiley-Interscience, 2007.

[**Qi et Davidson, 2009**] Z. Qi et I. Davidson. A principled and flexible framework for finding alternative clusterings. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 717–726, 2009.

[**Quinlan, 1996**] J. Quinlan. Improved use of continuous attributes in {C4.5}. *Journal of Artificial Intelligence Research*, 4 :77–90, 1996.

[**Rand, 1971**] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66 :622–626, 1971.

[**Rauber et al., 2000**] A. Rauber, E. Pampalk, et J. Paralic. Empirical evaluation of clustering algorithms. *Journal of Information and Organizational Sciences*, 24(2) :195–209, 2000.

[**Rodriguez et Egenhofer, 2003**] M. A. Rodriguez et M. J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15(2) :442–456, 2003.

[**Roussey, 2001**] C. Roussey. *Une méthode d'indexation sémantique adaptée au corpus multilingues*. Thèse de doctorat, Insa de Lyon, 2001.

[**Ruiz et al., 2009**] C. Ruiz, M. Spiliopoulou, et E. Menasalvas. Density-based semi-supervised clustering. *Data Mining and Knowledge Discovery*, 2009.

[**Salvador et Chan, 2004**] S. Salvador et P. Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *IEEE International Conference on Tools with Artificial Intelligence*, pages 576–584, 2004.

[**Salvatore et al., 1999**] E. Salvatore, C. Esposito, T. Krug, et R. Green. Simulation of the spectral bands of the ccd and wfi cameras of the cbers satellite using aviris data, 1999.

[**Schapire, 1990**] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2) : 197–227, 1990. ISSN 0885-6125.

[**Schikuta et Erhart, 1997**] E. Schikuta et M. Erhart. The bang-clustering system : Grid-based data analysis. In *International Symposium on Advances in Intelligent Data Analysis, Reasoning about Data*, pages 513–524, 1997.

[**Schwering et Raubal, 2005**] A. Schwering et M. Raubal. Measuring semantic similarity between geospatial conceptual regions. In *International Conference on GeoSpatial Semantics (GeoS)*, volume 3799 of *Lecture Notes in Computer Science*, pages 90–106, Mexico City, Mexico, November 2005.

[**Shafer, 1978**] G. Shafer. A mathematical theory of evidence. *Journal of the American Statistical Association*, 73(363) :677–678, 1978.

[**Sheeren et al., 2006a**] D. Sheeren, A. Puissant, C. Weber, P. Gancarski, et C. Wemmert. Deriving classification rules from multiple sensed urban data with data mining. In *Workshop of the EARSel Special Interest Group Urban Remote Sensing*, 2006a.

[**Sheeren et al., 2006b**] D. Sheeren, A. Quirin, A. Puissant, P. Gancarski, et C. Weber. Discovering rules with genetic algorithm to classify remotely sensed data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS'06)*. IEEE, 2006b.

[**Shi, 2000**] J. Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing*, pages 470–477. MIT Press, 2000.

[**Shi et Malik, 2000**] J. Shi et J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :888–905, 2000.

[**Smeulders et al., 2000**] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, et R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 22(12) :1349–1380, 2000. ISSN 0162-8828.

[**Smith et Mark, 2000**] B. Smith et D. M. Mark. Geographic categories : An ontological investigation. *International Journal of Geographical Information Science*, 15 :591–612, 2000.

[**Soille, 2003**] P. Soille. *Morphological Image Analysis : Principles and Applications*. Springer-Verlag New York, Inc., 2003.

[**Solomonoff et al., 1998**] A. Solomonoff, A. Mielke, M. Schmidt, et H. Gish. Clustering speakers by their voices. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 757–760, May 1998.

[**Strehl et Ghosh, 2002**] A. Strehl et J. Ghosh. *Journal on Machine Learning Research*, 3 : 583–617, 2002.

[**Tabachnick et Fidell, 2006**] B. G. Tabachnick et L. S. Fidell. *Using Multivariate Statistics*. Allyn & Bacon, Inc., Needham Heights, MA, USA, 2006.

[**Tan et al., 2005**] P. Tan, M. Steinbach, et V. Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[**Topchy et al., 2005**] A. Topchy, A. Jain, et W. Punch. Clustering ensembles : models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12) : 1866–1881, 2005.

[**Topchy et al., 2003**] A. P. Topchy, A. K. Jain, et W. F. Punch. Combining multiple weak clusterings. In *IEEE International Conference on Data Mining*, pages 331–338, 2003.

[**Tversky, 1977**] A. Tversky. Features of similarity. *Psychological Review*, 84 :327–352, 1977.

[**Van Ness, 1973**] J. W. Van Ness. Admissible clustering procedures. *Biometrika*, pages 422–424, 1973.

[**van Rijsbergen, 1979**] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[**Vanegas et al., 2009**] M. C. Vanegas, I. Bloch, et J. Inglada. Fuzzy spatial relations for high resolution remote sensing image analysis : The case of to go across. volume 4, pages 773–776, 2009.

[**Verykios et al., 2004**] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, et Y. Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 33(1) : 50–57, 2004. ISSN 0163-5808.

[**Vincent et Soille, 1991**] L. Vincent et P. Soille. Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(6) :583–598, 1991.

[**Wagstaff et al., 2001**] K. Wagstaff, C. Cardie, S. Rogers, et S. Schroedl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning*, pages 557–584, 2001.

[**Wagstaff, 2007**] K. L. Wagstaff. Value, cost, and sharing : Open issues in constrained clustering. In *International Workshop on Knowledge Discovery in Inductive Databases*, pages 1–10, 2007.

[**Wang et al., 2007**] C. Wang, W. Chen, P. Yin, et J. Wang. Semi-supervised clustering using incomplete prior knowledge. In *Computational Science – ICCS 2007*, pages 192–195, 2007.

[**Wemmert, 2000**] C. Wemmert. *Classification hybride distribuée par collaboration de méthodes non supervisées*. Thèse de doctorat, Université de Strasbourg, 2000.

[**Wemmert et Gancarski, 2002a**] C. Wemmert et P. Gancarski. A multi-view voting method to combine unsupervised classifications. In *Artificial Intelligence and Applications*, pages 447–452, Malaga, Spain, 2002a.

[**Wemmert et Gancarski, 2002b**] C. Wemmert et P. Gancarski. A multi-view voting method to combine unsupervised classifications. In *Artificial Intelligence and Applications*, pages 447–452, Malaga, Spain, september 2002b.

[**Wemmert et al., 2009**] C. Wemmert, A. Puissant, G. Forestier, et P. Gancarski. Multiresolution remote sensing image clustering. *IEEE Geoscience and Remote Sensing Letters*, 6 :533 – 537, july 2009. ISSN 1545-598X.

[**Wolpert, 1992**] D. H. Wolpert. Stacked generalization. *Neural Network*, 5(2) :241–259, 1992. ISSN 0893-6080.

[**Wolpert et Macready, 1997**] D. H. Wolpert et W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1) :67–82, 1997.

[**Xu et Wunsch, 2005**] R. Xu et D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3) :645–678, 2005.

[**Zhang, 1996**] Y. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8) :1335–1346, 1996.

[**Zhou et Tang, 2006**] Z.-H. Zhou et W. Tang. Clusterer ensemble. *Knowledge-Based Systems*, 19(1) :77–83, 2006.

[**Zhu, 2008**] X. Zhu. Semi-supervised learning literature survey. 2008.

[**Zlatoff et al., 2004**] N. Zlatoff, B. Tellez, et A. Baskurt. Image understanding using domain knowledge. In *Proceedings of Recherche d'information Assistée par Ordinateur (RIAO)*, pages 277–290, 2004.

Résumé

Les nouveaux défis apportés par les données complexes imposent le développement de nouvelles méthodes de fouille de données. Dans la première partie de cette thèse, nous nous intéressons plus particulièrement à l'étape du processus de fouille qu'est la classification non supervisée (*clustering*). Celle-ci consiste à créer, de manière automatique, des groupes au sein d'un ensemble d'objets. Ces groupes sont créés de telle sorte que les objets au sein d'un même groupe soient les plus similaires possibles, et que des objets de groupes différents soient les plus distincts possible. Il existe de nombreux algorithmes de clustering différents qui peuvent, à partir des mêmes données, fournir des résultats très différents. En effet, en fonction de leurs paramètres ou de leur initialisation, le résultat final de l'algorithme peut diverger. Le clustering collaboratif consiste à utiliser plusieurs algorithmes différents, ou le même algorithme avec des paramètres différents, au sein d'un processus collaboratif. Notre point de départ est la méthode collaborative SAMARAH, proposée par Cédric Wemmert. Une étude détaillée de cette méthode collaborative est proposée ainsi qu'une description des améliorations apportées à celle-ci lors de ce travail de thèse. Une nouvelle méthode de clustering collaborative est présentée, celle-ci menant à une amélioration de la qualité de la collaboration entre les méthodes.

Ces recherches sur l'utilisation simultanée de plusieurs résultats de clustering nous ont poussés à étudier plus en détail comment prendre en compte toutes les informations disponibles sur un problème. Dans ce cadre, l'intégration de connaissances dans les algorithmes de clustering a connu un fort intérêt ces dernières années. En effet, les approches totalement non supervisées tendent à montrer leurs limites face à des données de plus en plus complexes. De plus, des connaissances du domaine, contextuelles ou complémentaires sont souvent disponibles. Il convient alors de développer des méthodes permettant la prise en compte de ces connaissances afin d'améliorer la qualité des résultats et les performances des algorithmes. Nous étudions dans cette thèse les différentes méthodes proposées dans la littérature pour intégrer des connaissances dans les algorithmes de clustering. Nous présentons en détail différents critères de pureté permettant l'intégration de connaissance sous forme d'objets étiquetés. Enfin, nous proposons une approche innovante d'intégration de connaissances dans le cadre du clustering collaboratif.

Les géosciences et plus particulièrement l'observation de la Terre via les images de télédétection ont été le domaine privilégié d'application des propositions faites lors de cette thèse. Dans cette thèse, nous nous intéressons principalement à l'utilisation d'images issues de capteurs embarqués à bord de satellites ou aéroportés. L'image de télédétection est le prototype même d'une donnée complexe de par sa structure physique mais aussi par le fossé sémantique entre les informations bas niveau (les valeurs radiométriques des pixels) et les informations à extraire (par exemple l'occupation du sol). Ce fossé sémantique est défini comme le manque de concordance entre l'information bas niveau et l'interprétation faite par un expert. Dans cette thèse, nous proposons un formalisme de représentation des connaissances de l'expert pouvant être utilisé pour l'identification automatique d'objets géographiques dans les images de télédétection. Ces approches innovantes ont permis de mettre en place un processus automatique d'interprétation d'image guidé par les connaissances de l'expert. Loin de vouloir fournir un système totalement automatique, le but est d'aider l'expert dans le processus d'interprétation en lui fournissant des outils lui permettant d'améliorer les traitements.Enfin, nous étudions également l'utilisation du clustering collaboratif pour le traitement de données multisources hétérogènes en géosciences. Une approche permettant l'utilisation conjointe d'images hétérogènes à différentes résolutions est proposée et évaluée.