Classification spectrale semi-supervisée : Application à la supervision de l'écosystème marin

Guillaume Wacquet

CGR de Calais **École Doctorale Sciences pour l'Ingénieur**
Numéro d'ordre :

THÈSE

présentée et soutenue publiquement le 08 décembre 2011

pour l'obtention du grade de

Docteur de l'Université du Littoral Côte d'Opale

(Discipline : Traitement du Signal et des Images)

par

Guillaume WACQUET

Titre :

Classification spectrale semi-supervisée. Application à la surveillance de l'écosystème marin

Composition du jury

Rapporteurs :	**Stéphane Canu** Professeur à l'Institut National des Sciences Appliquées de Rouen **Fadi Dornaika** Ikerbasque Research Professor, Universidad del Pais Vasco, San Sebastian, Espagne
Examinateurs :	**Sylvie Thiria** Professeur à l'Université de Versailles Saint-Quentin-en-Yvelines **Gérard Govaert** Professeur à l'Université de Technologie de Compiègne **Luis Felipe Artigas** Maître de Conférences à l'Université du Littoral Côte d'Opale
Encadrante :	**Émilie Caillault Poisson** Maître de Conférences à l'Université du Littoral Côte d'Opale
Directeur de Thèse :	**Denis Hamad** Professeur à l'Université du Littoral Côte d'Opale

Laboratoire d'Informatique, Signal et Image de la Côte d'Opale – EA 4491
Maison de la Recherche Blaise Pascal – Centre Universitaire de la Mi-Voix
50 rue Ferdinand Buisson – B.P. 719, 62228 Calais Cedex, France

Remerciements

Ce travail de thèse a été réalisé au sein du Laboratoire d'Informatique, Signal et Image de la Côte d'Opale (LISIC) de l'Université du Littoral Côte d'Opale. Je tiens donc à remercier le Professeur Christophe Renaud, directeur du laboratoire LISIC, pour son accueil et le dynamisme qu'il a impulsé au laboratoire.

Je profite de ces quelques lignes pour remercier également, très chaleureusement, mon directeur de thèse, le Professeur Denis Hamad, pour son implication tant au niveau scientifique que rédactionnel, mais également pour son soutien sans failles durant ces trois années de thèse.

Je remercie également Madame Émilie Poisson-Caillault et Monsieur Pierre-Alexandre Hébert de m'avoir fait découvrir le monde de la recherche et de m'avoir initié au domaine de la classification de par l'apport de leurs compétences et remarques scientifiques.

Les membres du jury trouvent ici toute ma gratitude pour avoir accepté de juger ce travail. Je remercie très sincérement Monsieur Stéphane Canu, Professeur à l'INSA de Rouen et Directeur du laboratoire LITIS, et Monsieur Fadi Dornaïka, Ikerbasque Research Professor en Espagne, d'avoir accepté d'assurer la tâche de rapporteur. Je remercie également le Professeur Sylvie Thiria, le Professeur Gérard Govaert et Monsieur Luis Felipe Artigas d'avoir accepté d'être membre du jury et d'examiner ce travail.

Je souhaiterai saluer l'ensemble des membres du laboratoire LISIC avec qui j'ai eu la chance de cohabiter pendant ces trois années, mais également les membres participants au projet INTERREG DYMAPHY (et plus particulièrement Monsieur Xavier Mériaux et Madame Natacha Guiselin du LOG) pour leur aide, leur sympathie et leurs encouragements.

Ces remerciements vont évidemment aussi à mes proches. Je tiens à remercier mes parents et mes frères, Jean-Pierre mais surtout Caroline, de m'avoir soutenu et encouragé tout au long de cette thèse.

Cette thèse a été co-financée par une allocation de recherche du ministère de l'enseignement supérieur et par le Fonds Européen de DÉveloppement Régional (FEDER) via le projet DYMAPHY (INTERREG IV A "2 Mers").

Guillaume WACQUET

Table des matières

Glossaire

N	:	Nombre d'objets
M	:	Nombre d'attributs
$\mathcal{X}$	:	Ensemble des N objets
$\mathcal{F}$	:	Ensemble des D attributs
X	:	Matrice de données (de dimensions $N \times D$)
d_{ij}	:	Fonction de distance entre l'objet x_i et l'objet x_j
Δ	:	Matrice de distances (de dimensions $N \times N$)
w_{ij}	:	Fonction de similarité entre l'objet x_i et l'objet x_j
W	:	Matrice de similarités (de dimensions $N \times N$)
$\tilde{W}$	:	Matrice de similarités augmentée (de dimensions $(N+K) \times (N+K)$)
p_{ij}	:	Fonction de proximité entre l'objet x_i et l'objet x_j
P	:	Matrice de proximités (de dimensions $N \times N$)
σ	:	Paramètre de dispersion
G	:	Graphe des données
$\tilde{G}$	:	Graphe augmenté des données
V	:	Ensemble des noeuds du graphe G
E	:	Ensemble des arcs inter-noeuds du graphe G
d_i	:	Degré de l'objet x_i
D	:	Matrice de degré
L	:	Matrice laplacienne non normalisée
$\overline{L}_1$	:	Matrice laplacienne normalisée asymétrique
$\overline{L}_2$	:	Matrice laplacienne normalisée symétrique
$\overline{L}_3$	:	Matrice Laplacienne normalisée
$hflr-ls$	:	Hauteur du signal de fluorescence rouge en basse sensibilité
$iflr-ls$	:	Intégrale du signal de fluorescence rouge en basse sensibilité
$lflr-ls$	:	Longueur du signal de fluorescence rouge en basse sensibilité
$nbflr-ls$	:	Nombre de pics du signal de fluorescence rouge en basse sensibilité
λ_{em}	:	Longueur d'onde d'émission

λ_i	:	i^{me} valeur propre
z_i	:	i^{me} vecteur propre
K	:	Nombre total de groupes
$\mathcal{K}$	:	Nombre de voisins
C	:	Ensemble des groupes obtenus
C_k	:	k^{ime} groupe d'objets
Y	:	Vecteur d'étiquettes de groupes
F	:	Vecteur (ou matrice) indicateur (indicatrice) optimal(e) d'appartenance aux groupes
ACP	:	Analyse en Composantes Principales
LPP	:	Locality Preserving Projection
$FLO-LS$	:	Signal de fluorescence orange en basse sensibilité
$FLR-HS$	:	Signal de fluorescence rouge en haute sensibilité
$FLR-LS$	:	Signal de fluorescence rouge en basse sensibilité
$FLY-HS$	:	Signal de fluorescence jaune en haute sensibilité
$FLY-LS$	:	Signal de fluorescence jaune en basse sensibilité
FWS	:	Signal de diffusion à petits angles (ForWard Scatter)
$SWS-HS$	:	Signal de diffusion à 90° en haute sensibilité (SideWard Scatter)
$SWS-LS$	:	Signal de diffusion à 90° en basse sensibilité (SideWard Scatter)

Introduction générale

1 Contexte

Le travail de thèse concerne la classification spectrale avec intégration de connaissances contextuelles simples à formaliser. L'application est relative à la surveillance de l'écosystème marin par analyse cytométrique d'échantillons de culture et d'échantillons prélevés du milieu marin. Le travail s'est déroulé dans le cadre du projet INTERREG IV A "2 Mers Seas Zeeën" DYMAPHY (2008-2013) : "Développement d'un système d'observation dynamique pour la détermination de la qualité des eaux marines basé sur l'analyse du phytoplancton". Dans ce cadre, nous disposons de données numériques abondantes issues de dispositifs de type cytométres en flux. En effet, un tel type d'appareils génère huit séquences de signaux formant un profil cytométrique caractéristique de la cellule. L'analyse automatique d'un échantillon d'eau conduit naturellement à la génération d'une base de signaux conséquente. Pour ces signaux, les biologistes sont capables de comparer certains profils de cellules, voire de reconnaître quelques espèces à partir de l'observation de leurs profils. Ce type d'information sera introduit dans la conception de nos algorithmes de classification automatique.

Nous plaçons le travail dans un contexte plus général de conception de systèmes d'aide à la décision. En effet, le développement technologique (moyens de calcul et de communication) et la disponibilité de capteurs et d'instruments de mesures de faibles coûts, conduisent de plus en plus à la génération de bases de données de grande taille : signaux, images, documents, etc. Si dans ces grandes bases, nous sommes sûrs d'avoir une information complète et utile, celle-ci risque d'être noyée dans la masse. Dans ce sens, il est naturel d'avoir recours à la classification automatique pour l'exploration et la structuration des données pour en extraire des informations utiles.

La représentation des données sous forme de graphe, dont les noeuds sont interconnectés par des poids représentant les similarités entre données, conduit à considérer le problème de recherche de groupes naturels dans les données comme étant un problème de coupes de graphes.

C'est l'idée de base de la classification spectrale.

Les algorithmes de classification spectrale traitent donc le problème de partitionnement des données sous l'angle de coupes de graphe, par optimisation d'un critère de coupe normalisé. Ils consistent en une étape de génération d'un espace spectral à partir des vecteurs propres de la matrice Laplacienne associée au graphe, suivie d'une étape de partitionnement. L'espace spectral ainsi construit devrait mieux révéler la structuration en groupes naturels pour que la partition soit réalisée par un simple algorithme de classification non supervisée.

Les performances des algorithmes de classification spectrale dépendent de la qualité des données à disposition mais aussi de l'intégration de connaissances contextuelles de plus haut niveau dans leur conception. Dans ce travail, nous nous intéressons, en particulier, aux algorithmes intégrant des connaissances faciles à formaliser par l'analyste, dites contraintes de comparaison. Les contraintes sont de type "Must-Link" si deux données sont similaires et doivent donc être groupées ensemble, ou "Cannot-Link" s'ils sont non similaires et doivent appartenir à des groupes distincts. Les contraintes interviennent dans le critère d'optimisation de coupe de graphes sous formes de pénalités. L'espace spectral généré par l'algorithme spectral contraint doit, dans ce cas, révéler la structuration en groupes naturels tout en respectant autant que possible ces contraintes.

Récemment, dans la littérature, la classification spectrale contrainte a attiré une attention particulière [Wagstaff, 2002] [Davidson et al., 2006]. Globalement, nous distinguons deux modalités d'intégration des contraintes : soit explicitement dans la matrice de similarité en imposant des poids (valeurs binaires), soit implicitement comme pénalité dans le critère de coupe normalisé. Ces algorithmes ont été développés soit dans un but de réduction de la dimension (voire de visualisation), soit dans un objectif de classification.

La plupart des algorithmes de classification spectrale contrainte sont basés sur la bi-coupe ou bi-partition du graphe. Dans ce travail, nous proposons un nouvel algorithme qui permet de générer un espace spectral par optimisation d'un critère de multi-coupe normalisé avec ajustement des coefficients de pénalité dus aux contraintes. Nous étudions également le problème de la détermination automatique du nombre de classes. Les performances de l'algorithme sont mises en évidence sur différentes bases de données par comparaison à d'autres algorithmes de la littérature, au moyen de l'indice de Rand et du taux de contraintes respectées.

Pour la surveillance de l'écosystème marin, nous illustrons nos propos par l'emploi de deux

bases de données : une base issue de cellules de culture et une issue du milieu naturel. Dans ce cadre, les espèces phytoplanctoniques sont identifiées par analyse des signaux profils caractéristiques des cellules. Il est d'usage de générer, à partir des signaux profils, des attributs caractéristiques. Nous proposons d'employer les signaux profils bruts comme attributs. Pour ce faire, nous appliquons l'approche d'appariement élastique pour quantifier la similarité entre signaux. Le problème de classification est abordé avec des approches issues de différents domaines : perceptron multi-couches, séparateur à vaste marge, classifieur des k-plus proches voisins. Nous appliquons notre algorithme sur les deux bases de données et nous comparons ses performances avec les autres algorithmes de la littérature.

2 Organisation de la thèse

Le manuscrit est structuré en six chapitres. Les quatre premiers sont à caractère fondamental et traitent les différents aspects de la classification spectrale. Les deux derniers sont applicatifs et sont dédiés à la surveillance de l'écosystème marin par recherche d'espèces de cellules phytoplanctoniques. La démarche est illustrée sur deux exemples : un échantillon de culture et un du milieu marin.

Le premier chapitre est relatif à la représentation des connaissances et des contextes de classification. En général, les connaissances disponibles à l'analyste sont des données numériques de bas niveau et éventuellement des informations contextuelles de niveau supérieur. Les données numériques se présentent sous forme de matrice rectangulaire (objets $\times$ attributs) ou sous forme de matrice carrée (objets $\times$ objets). Les données contextuelles correspondent aux étiquettes de classes, aux contraintes de comparaison entre objets, etc. Selon que ces connaissances sont plus ou moins complètes, nous nous trouvons dans des contextes de classification supervisée, semi-supervisée ou non supervisée. La représentation des données en graphes, auxquels sont associées des matrices de similarités, pose le problème de classification sous l'angle de coupe de graphes, qui constitue la base de la classification spectrale.

Le deuxième chapitre présente un état de l'art sur les algorithmes de classification spectrale basés sur une bi-partition ou une K-partition des données. Le principe est de définir un espace de projection spectral à partir des vecteurs propres de la matrice Laplacienne et de rechercher dans cet espace des groupes représentatifs des données. Les algorithmes spectraux dépendent des paramètres de la fonction similarité et du nombre de groupes recherchés. Différentes techniques d'ajustement de ces paramètres sont présentées.

Dans le troisième chapitre, nous passons en revue les différents algorithmes de la littérature pour la projection et la classification spectrale semi-supervisées, utilisant les contraintes de comparaison par paires d'objets, sous la forme de deux ensembles de contraintes de type "Must-Link" et "Cannot-Link". Les contraintes sont soit fournies par l'analyste, soit souvent générées aléatoirement à partir d'étiquettes de classes. Elles se distinguent par leur cohérence et l'information qu'elles peuvent apporter à l'algorithme de classification. L'espace spectral généré par les algorithmes de classification spectrale contrainte, est censé révéler la présence de groupes naturels tout en respectant les contraintes de comparaison.

Nous réalisons, dans le quatrième chapitre, l'étude et la mise en oeuvre d'un algorithme de classification spectrale semi-supervisée, pour les problèmes multi-groupes. En effet, les méthodes traditionnelles possédant certaines limites (pondération des contraintes, application au cas $K = 2$, etc.), il nous paraît intéressant de proposer une technique capable de pallier ces problèmes. La méthode s'appuie donc principalement sur l'optimisation d'un critère de coupe de graphe pénalisé par le non respect des contraintes de comparaison définies. L'algorithme est comparé à plusieurs algorithmes de la littérature sur des bases de données UCI, en faisant varier les pourcentages de contraintes de comparaison. Pour ce faire, nous utilisons comme scores de performance, le critère de coupe normalisée, l'indice de Rand, la F-mesure et le pourcentage de contraintes respectées.

Le cinquième chapitre présente le principe de la caractérisation du phytoplancton par cytométrie en flux. Une cellule phytoplanctonique est caractérisée par huit séquences de signaux, constituant le profil cytométrique de la cellule. Nous disposons d'une base de signaux profils de 700 cellules cultivées en laboratoire et appartenant à sept espèces (100 cellules par espèce). Il est d'usage de générer à partir des profils cytométriques, des attributs caractéristiques des cellules et construire une base de données "attributs", dans un objectif de classification. Pour éviter la perte d'information due à l'extraction d'attributs, nous proposons une alternative s'appuyant sur l'appariement élastique entre profils cytométriques qui a l'avantage de tolérer des déformations temporelles locales des séquences. L'ensemble des mesures d'appariements constitue une base de données "similarités". Pour évaluer les performances de l'appariement élastique des signaux par rapport à l'extraction d'attributs, nous appliquons différents types de classifieurs dans des contextes supervisés et semi-supervisés.

Enfin, dans le sixième et dernier chapitre, nous nous intéressons à l'analyse d'une base de signaux de profils cytométriques de cellules provenant d'un échantillon du milieu naturel. Nous avons extrait une base "attributs" et une base "similarités". La base de profils des cellules

étant partiellement étiquetée face à la diversité du milieu naturel, nous sommes donc confrontés à un problème de classification non supervisée. Dans ce cadre, il est nécessaire d'étudier le problème de la détermination automatique du nombre de groupes et d'analyser les résultats de la classification. Sur cette base, nous appliquons également les algorithmes spectraux semi-supervisés du chapitre 4 et évaluons leurs performances en présence de faibles pourcentages de contraintes de comparaison.

3 Contributions de la thèse

Les contributions de la thèse se situent au niveau des points suivants :

- État de l'art sur les algorithmes de classification spectrale non supervisée et semi-supervisée, en particulier ceux faisant intervenir les contraintes de comparaison par paires d'objets.
- Proposition d'un nouvel algorithme de classification spectrale multi-groupes intégrant les contraintes de comparaison par paires. Les performances de cet algorithme sont comparées avec les algorithmes de la littérature sur des bases de données extraites des archives UCI.
- Proposition de l'appariement élastique comme mesure de similarité, pour comparer des signaux de longueurs différentes.
- Application à la surveillance de l'écosystème marin par analyse des profils cytométriques des cellules phytoplanctoniques, dans des cultures de laboratoire et dans le milieu naturel.

Chapitre 1

Représentation des connaissances et contextes de classification

1.1 Introduction

Pour les systèmes d'aide à la décision, nous disposons, en général, de données numériques abondantes (données quantitatives) issues des capteurs et des instruments de mesure, et éventuellement de connaissances contextuelles *a priori* ou *a posteriori* par retour d'expérience (données qualitatives). Les données numériques se présentent sous forme de matrice rectangulaire (objets $\times$ attributs) où chaque objet est caractérisé par un vecteur attribut, ou sous forme de matrice carrée (objets $\times$ objets) exprimant la similarité entre les objets deux à deux.

Les données contextuelles correspondent, par exemple, aux étiquettes de classes des objets, aux contraintes de comparaison entre objets, etc. Selon que les connaissances contextuelles sont plus moins complètes sur les objets, nous appliquons des méthodes de classification supervisée, semi-supervisée ou non supervisée. La représentation des données sous forme de graphe et la possibilité d'intégrer les connaissances contextuelles dans celui-ci, conduisent à considérer le problème de la classification sous l'angle de la coupe de graphe qui est résolue par des algorithmes de classification spectrale.

1.2 Représentation numérique des données

Les données (numériques ou qualitatives) disponibles à l'analyse, sont soit sous forme de matrice rectangulaire (objets $\times$ attributs), soit sous forme de matrice carré (objets $\times$ objets). La similarité entre données est à la base de la classification.

1.2.1 Représentation des données sous forme de matrices

Soit $\mathcal{X} = \{x_1, \ldots, x_i, \ldots, x_N\}$ un ensemble de N objets. Chaque objet x_i est une entité décrite par un ensemble $\mathcal{F} = \{f_1, \ldots, f_r, \ldots, f_M\}$ composé de M attributs caractéristiques pouvant avoir des valeurs quantitatives (à valeurs réelles, binaires, etc.) ou qualitatives (nominales, ordinales, catégorielles, etc.). Le vecteur observation est indifféremment appelé donnée, objet, voire point. Dans le cadre de notre application de cytométrie, les objets sont les cellules phytoplanctoniques et les attributs caractéristiques sont la longueur, la hauteur, l'intégrale et le nombre de pics extraits des signaux cytométriques.

L'ensemble des données peut se mettre sous deux formes : la matrice de données objets $\times$ attributs, et la matrice de proximité objets $\times$ objets, permettant de représenter les proximités entre objets. La matrice de données (notée X) est construite à partir de la mise en correspondance de l'ensemble des objets $\mathcal{X}$ avec l'ensemble des attributs $\mathcal{F}$, alors que la matrice de proximités P résulte du calcul des ressemblances ou des dissemblances entre ces différents objets. Ces deux types de matrices sont représentés comme suit :

$$X = \begin{pmatrix} x_{11} & \ldots & x_{1r} & \ldots & x_{1M} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{i1} & \ldots & x_{ir} & \ldots & x_{iM} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{N1} & \ldots & x_{Nr} & \ldots & x_{NM} \end{pmatrix} \tag{1.1}$$

$$P = \begin{pmatrix} p(x_1,x_1) & \ldots & p(x_1,x_i) & \ldots & p(x_1,x_N) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ p(x_i,x_1) & \ldots & p(x_i,x_i) & \ldots & p(x_i,x_N) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ p(x_N,x_1) & \ldots & p(x_N,x_i) & \ldots & p(x_N,x_N) \end{pmatrix} \tag{1.2}$$

La matrice de données X est composée de l'ensemble des valeurs des attributs (en colonnes) pour chacun des objets (en lignes). Un objet est alors défini par le vecteur d'attributs dans $\mathbb{R}^M$, noté $x_i = (x_{i1}, \ldots, x_{ir}, \ldots, x_{iM})^T$, et peut être représenté par un point dans un espace à M dimensions. Un attribut est, quant à lui, défini par un vecteur dans $\mathbb{R}^N$, composé de valeurs pour l'ensemble des objets, et noté $f_j = (x_{1j}, \ldots, x_{ij}, \ldots, x_{Nj})^T$. Cette matrice X est donc définie dans $\mathbb{R}^{N \times M}$.

En ce qui concerne la matrice de proximités P, chaque ligne est composée des mesures de proximités entre un objet et tous les autres. Le terme $p(x_i, x_j)$ désigne la proximité entre les objets x_i et x_j. Celle-ci peut être calculée soit à l'aide d'une fonction de distance, soit à l'aide d'une fonction de similarité. La matrice P est donc définie dans $\mathbb{R}^{N \times N}$.

Pour la suite, nous nous intéressons principalement aux attributs prenant des valeurs quantitatives (et plus particulièrement aux valeurs numériques réelles ou discrètes). Cependant, il est possible d'adapter les méthodes présentées dans ce mémoire, aux attributs ayant des valeurs qualitatives [Wan and Eick, 1998].

1.2.2 Mesures de comparaison entre paires d'objets

Les premières approches proposées en classification reposaient essentiellement sur la proximité entre les objets à classer, c'est-à-dire la distance qui sépare les objets les uns des autres. Cette mesure de proximité peut alors être vue sous deux aspects :

- mesure de dissemblance où plus deux objets sont proches, plus la valeur de la mesure de dissemblance entre ces objets est faible (notion de distance) ;
- mesure de ressemblance où plus deux objets sont proches, plus la valeur de la mesure de ressemblance entre ces objets est élevée (notion de similarité).

Les fonctions distance

Il est souvent utile d'avoir une représentation géométrique afin de fournir une première vue des données dans le but de repérer des groupes d'objets naturels ou des objets isolés appelés "outliers". Chaque attribut représente alors un axe de coordonnées. L'espace des données est donc un espace euclidien à M dimensions dans lequel chaque objet est représenté par un point. La distance $d(x_i, x_j)$ (notée $d(i, j)$) entre les points x_i et x_j, est ensuite mesurée dans cet espace grâce à l'utilisation d'une fonction de distance $d : \mathbb{R}^M \times \mathbb{R}^M$ dans $\mathbb{R}^+$, respectant les propriétés suivantes [Losee, 1998] :

1. $d(i, j) \geq 0$ (contrainte de positivité),
2. $d(i, j) = 0$ si et seulement si $x_i = x_j$,
3. $\forall x_i, x_j,\ d(i, j) = d(j, i)$ (contrainte de symétrie),
4. $\forall x_i, x_j, x_l,\ d(i, j) \leq d(i, l) + d(l, j)$ (contrainte d'inégalité triangulaire).

Dans la littérature, il est aisé de trouver plusieurs définitions de métrique de distance. Cependant, toutes ces mesures ont une interprétation identique : il est très probable que des points proches dans l'espace de projection représentent des données d'un même groupe (la distance est alors proche de 0), alors que des points éloignés représentent, quant à eux, des données appartenant à des groupes différents (la distance tend vers l'infini). La liste suivante non exhaustive, reprend les plus classiques :

– **La distance de Minkowski** :

$$d(i,j) = \sqrt[p]{\sum_{r=1}^{M} |x_{ir} - x_{jr}|^p} \tag{1.3}$$

avec p un entier positif non nul.

– **La distance euclidienne** est la plus connue et la plus utilisée. Elle peut être vue comme un cas particulier de la distance de Minkowski pour $p = 2$:

$$d(i,j) = \sqrt{\sum_{r=1}^{M} |x_{ir} - x_{jr}|^2} \tag{1.4}$$

– **La distance de Manhattan** est également un cas particulier de la distance de Minkowski pour $p = 1$ (également appelée "métrique absolue") :

$$d(i,j) = \sum_{r=1}^{M} |x_{ir} - x_{jr}| \tag{1.5}$$

– **La distance de Tchebychev**, correspondant au cas $p = \infty$ de la formule de Minkowski, est aussi appelée "métrique maximum" :

$$\begin{aligned} d(i,j) &= \lim_{p\to\infty} \sqrt[p]{\sum_{r=1}^{M} |x_{ir} - x_{jr}|^p} \\ &= \max_r |x_{ir} - x_{jr}| \end{aligned} \tag{1.6}$$

La matrice de distances Δ est alors représentée comme suit :

$$\Delta = \begin{pmatrix} 0 & \dots & d(1,i) & \dots & d(1,N) \\ \dots & \dots & \dots & \dots & \dots \\ d(i,1) & \dots & 0 & \dots & d(i,N) \\ \dots & \dots & \dots & \dots & \dots \\ d(N,1) & \dots & d(N,i) & \dots & 0 \end{pmatrix} \tag{1.7}$$

Les fonctions similarités

De nombreux algorithmes conventionnels de classification utilisent la seule notion de distance afin d'obtenir des classes (ou des groupes) de points. En effet, il semble cohérent de regrouper les points dont la distance les uns par rapport aux autres est faible, et de séparer les points les plus éloignés. Cependant, d'autres algorithmes, plus récents, lui préfèrent la notion de similarité entre points, qui est plus générale. De plus, contrairement à la mesure de distance qui prend des valeurs entre 0 et l'infini, la mesure de similarité permet de fournir des valeurs quantifiables entre 0 et 1, plus aisées à interpréter, comme illustré sur un exemple composé de trois points notés x_1, x_2 et x_3.

$$X = \begin{pmatrix} 1 & 6 \\ 1.5 & 7 \\ 1.5 & 4.5 \end{pmatrix} \tag{1.8}$$

Le tableau 1.1 reprend les valeurs de distances et de similarités entre les points x_1, x_2 et x_3. Les points x_1 et x_2 sont considérés similaires avec une similarité $w_{12} > 0.5$.

	$\{x_1, x_2\}$	$\{x_1, x_3\}$	$\{x_2, x_3\}$
Distance	$d(1,2) = 1.12$	$d(1,3) = 1.58$	$d(2,3) = 2.50$
Similarité	$w_{12} = 0.54$	$w_{13} = 0.29$	$w_{23} = 0.04$

TABLE 1.1 – Mesures de distances (euclidienne) et de similarités (noyau gaussien avec $\sigma = 1$).

La similarité $w(x_i, x_j)$ (notée w_{ij}) entre deux points x_i et x_j est calculée en fonction de la distance entre ces points. La fonction similarité utilisée $w : \mathbb{R}^N \times \mathbb{R}^N$ dans $[0,1]$ respecte les propriétés suivantes :

1. $w_{ij} \in [0,1]$ (contraintes de normalisation),
2. $w_{ij} = 1$ si et seulement si $x_i = x_j$,
3. $w_{ij} = w_{ji}$ (contraintes de symétrie).

Contrairement à la métrique de distance, deux points sont considérés comme similaires si la valeur de la similarité w_{ij} (ou w_{ji}) est proche de 1. En revanche, ils sont considérés comme très différents l'un de l'autre si la valeur de la similarité w_{ij} (ou w_{ji}) est proche de 0. Un exemple simple de similarité est la fonction binaire : $w_{ij} \in \{0,1\}$.

De la même façon que pour la fonction distance, il existe, dans la littérature, plusieurs définitions différentes de métrique de similarités entre objets décrits par des attributs. Parmi celles-ci,

nous choisissons de présenter respectivement les trois formules se distinguant en terme de domaine d'utilisation, puis une alternative d'évaluation de similarité entre objets décrits par des signaux temporels :

– **L'inverse de la distance euclidienne** est une fonction de similarité, définie par : $w_{ij} = \frac{1}{1+d^2(i,j)}$. Kunegis et al [Kunegis et al., 2008] introduisent un paramètre de dispersion σ permettant de prendre en compte la dispersion locale des données, et noté σ^2 :

$$w_{ij} = \frac{1}{1+\frac{d^2(i,j)}{\sigma^2}} \tag{1.9}$$

Cette mesure de similarité permet alors d'obtenir un maximum global borné à 1 pour une distance nulle. Elle est fréquemment utilisée dans le domaine de la classification de documents.

– **Le noyau gaussien** (ou Gaussian RBF pour "Gaussian Radial Basis Function") est la fonction de similarité la plus couramment utilisée dans le domaine de la classification. L'importance relative des valeurs des attributs, mais également les relations de voisinage entre objets sont des facteurs essentiels à la mesure de similarités entre objets. Cette fonction non linéaire est définie par :

$$w_{ij} = \exp(-\frac{d^2(i,j)}{2\sigma^2}) \tag{1.10}$$

avec σ étant un paramètre de dispersion (ou paramètre d'échelle) permettant de prendre en compte la dispersion locale des données, et $d(i,j)$ une fonction de distance (souvent, la distance euclidienne) entre les deux points définis par x_i et x_j. Plus cette distance est importante dans l'espace de projection des données, plus la similarité entre les points est faible. En revanche, une distance faible engendre une forte similarité. Contrairement à la similarité basée sur l'inverse de la distance euclidienne au carré, cette fonction diminue de manière exponentielle lorsque les distances sont importantes, ce qui permet d'obtenir un poids élevé pour des objets très similaires [Kunegis et al., 2008]. Le paramètre de dispersion σ est généralement fixé par l'utilisateur. Cependant, il existe des méthodes de calcul automatique de σ telle que celle proposée dans [Zelnik-Manor and Perona, 2004]. Cette dernière est décrite dans le chapitre 2.

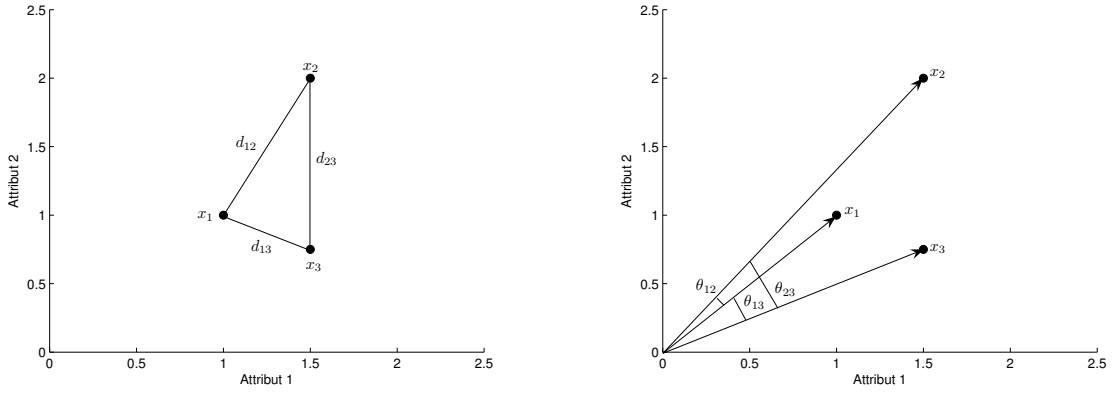
– **La fonction cosinus** permet de mesurer la similarité entre deux vecteurs normalisés (en fixant leur norme à 1), en calculant l'angle entre ces derniers [Salton, 1989]. Plus l'angle entre ces vecteurs est faible, plus les objets associés sont similaires. Cette fonction de similarité est utilisée essentiellement en exploration et analyse de contenu de documents

[Salton and Buckley, 1988] :

$$w_{ij} = |\cos(x_i, x_j)| = \frac{|{x_i}^T x_j|}{||x_i||.||x_j||} \tag{1.11}$$

L'inconvénient majeur de ce type de mesure (illustré sur la figure 1.1) réside dans le fait que la direction dans laquelle est projetée chaque objet prend beaucoup plus d'importance que l'amplitude des valeurs de chacun de ses attributs. En effet, la figure 1.1(a), représentant trois objets projetés dans un espace à deux dimensions (cf. matrice X ci-dessous), montre que l'objet x_1 est plus proche de l'objet x_3 que de l'objet x_2 au sens d'une mesure de distance. Sur la figure 1.1(b), même si visuellement, les points x_1 et x_3 semblent les plus proches, le tableau 1.2 (présentant les valeurs de distances et de similarités entre les objets x_1, x_2 et x_3) montre que le point x_1 est le plus proche de x_2 au sens de la similarité cosinus ($0.95 < 0.99$).

$$X = \begin{pmatrix} 1 & 1 \\ 1.5 & 2 \\ 1.5 & 0.75 \end{pmatrix} \tag{1.12}$$



(a) Distances inter-objets (b) Fonction cosinus

FIGURE 1.1 – Illustration de l'utilisation de la fonction cosinus pour le calcul de la similarité entre trois objets.

Ce tableau permet de mettre en évidence les problèmes liés à la fonction cosinus et montre l'intérêt du noyau gaussien dans le calcul de la similarité. En effet, les objets les plus similaires dans un espace de représentation donné, ne sont pas les mêmes selon la méthode utilisée (x_1 et x_3 avec $w_{13} = 0.86$ pour le noyau gaussien et x_1 et x_2 avec $\theta_{12} = 0.99$ pour la fonction cosinus).

	$\{x_1, x_2\}$	$\{x_1, x_3\}$	$\{x_2, x_3\}$
Distance euclidienne	$d(1,2) = 1.12$	$d(1,3) = 0.56$	$d(2,3) = 1.25$
Similarité (fonction inverse)	$w_{12} = 0.31$	$w_{13} = 0.94$	$w_{23} = 0.40$
Similarité (noyau gaussien)	$w_{12} = 0.54$	$w_{13} = 0.86$	$w_{23} = 0.46$
Similarité (fonction cosinus)	$\theta_{12} = 0.99$	$\theta_{13} = 0.95$	$\theta_{23} = 0.89$

TABLE 1.2 – Mesures de distances et de similarités.

– **L'appariement élastique** (noté DTW pour "Dynamic Time Warping") est une méthode de comparaison de courbes [Sakoe and Chiba, 1978]. La proximité entre signaux temporels numériques de tailles variables peut être évaluée en se basant sur la quantification des déformations à apparier ces deux signaux entre eux [Caillault et al., 2009]. Il s'agit donc d'une mesure de comparaison conjointe d'ensembles de signaux permettant de rendre compte de la similarité de deux courbes, en tolérant des déformations temporelles contrôlées, comme montré sur la figure 1.3. En effet, à partir des signaux de tailles différentes (représentés sur la figure 1.2), et après dilatation de la courbe possédant la longueur la plus faible, il est possible d'apparier les points de cette dernière avec les points de l'autre courbe (de manière élastique), comme montré sur la figure 1.3(a). Grâce à cette technique, il est possible d'obtenir des courbes comparables (cf. figure 1.3(b)) et donc d'obtenir une valeur de similarité. Les détails mathématiques liés à cette méthode sont présentés dans le chapitre 5.

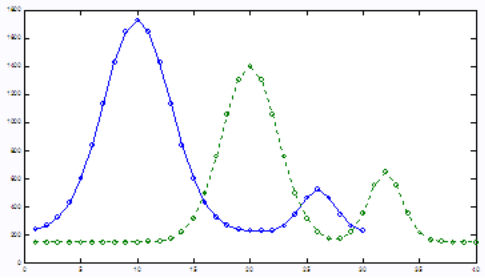
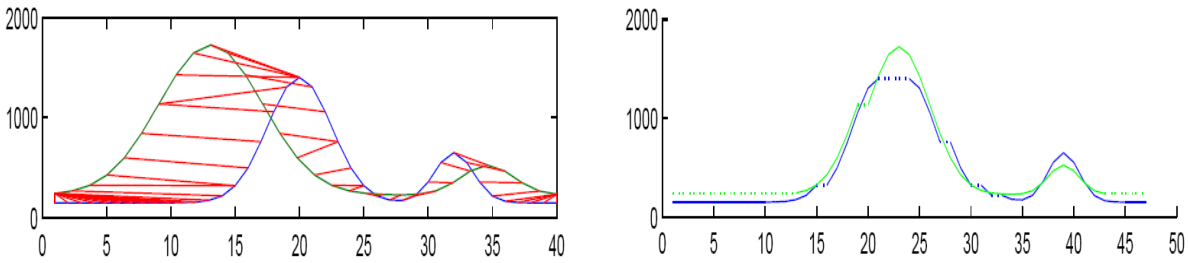


FIGURE 1.2 – Exemple de deux courbes à apparier

En respectant les propriétés et les définitions des fonctions de similarités précédemment citées, il est possible de construire la matrice W (de dimensions $N \times N$) dont les composantes w_{ij} représentent les mesures de similarités entre chaque paire d'objets (ou de signaux) (x_i, x_j) :



(a) Appariemment élastique (b) Courbes obtenues après appariement

FIGURE 1.3 – Appariemment élastique de deux courbes.

$$W = \begin{pmatrix} 1 & \dots & w_{1i} & \dots & w_{1N} \\ \dots & \dots & \dots & \dots & \dots \\ w_{i1} & \dots & 1 & \dots & w_{iN} \\ \dots & \dots & \dots & \dots & \dots \\ w_{N1} & \dots & w_{Ni} & \dots & 1 \end{pmatrix} \tag{1.13}$$

Cette matrice de similarités entre les N objets de l'ensemble $\mathcal{X}$ est symétrique et semi-définie positive, c'est à dire que toutes ses valeurs propres sont non négatives.

En résumé, les données se présentent sous forme d'une matrice X rectangulaire (objets $\times$ attributs) ou matrice W carrée (objets $\times$ objets). Il est possible d'obtenir une matrice de similarités W à partir de la matrice de données X. Cependant, il est important de noter que la réciproque est fausse. En effet, à partir de la matrice de similarités W, il n'est pas possible de retrouver les valeurs des élements de la matrice de données X.

1.2.3 Graphes et caractéristiques

Les graphes sont un mode de représentation abstraite des connaissances, utilisés dans le domaine de la classification. Ils permettent une description simple et synthétique des données, mais également des relations inter-données.

Présentation générale des graphes

Les données présentes dans l'ensemble $\mathcal{X} = \{x_1, \dots, x_N\}$ peuvent être mises sous la forme d'un graphe $G(V,E)$ [Luxburg, 2007] [Hamad and Biela, 2008], avec :

- V l'ensemble des noeuds, où le noeud v_i est associé à l'objet x_i ($V = \{v_1, \dots, v_N\}$) ;
- E l'ensemble des arcs inter-noeuds ($E \subseteq V \times V$).

Celui-ci peut alors se présenter sous deux formes :

- **Graphe non-orienté**, pour lequel les liens sont symétriques : $\forall v_i, v_j \in V, (v_i, v_j) \in E \Leftrightarrow (v_j, v_i) \in E$
- **Graphe orienté**, pour lequel un lien entre deux noeuds v_i et v_j relie soit v_i vers v_j, soit v_j vers v_i.

Dans la littérature, les données sont fréquemment représentées sous la forme d'un graphe non-orienté et pondéré. Cette pondération permet de modéliser les relations inter-objets à l'aide de leurs similarités. Le graphe $G(V,E)$ s'écrit alors $G(V,E,W)$ avec W étant une matrice de similarités ($w_{ij} = w_{ji} \geq 0$). Chaque arc inter-noeuds (pour deux noeuds différents : $v_i \neq v_j$) prend donc une valeur numérique, appelée poids (ou coût de l'arc), qui peut être un entier ou un réel et qui représente un concept de proximité tel que la distance ou la similarité.

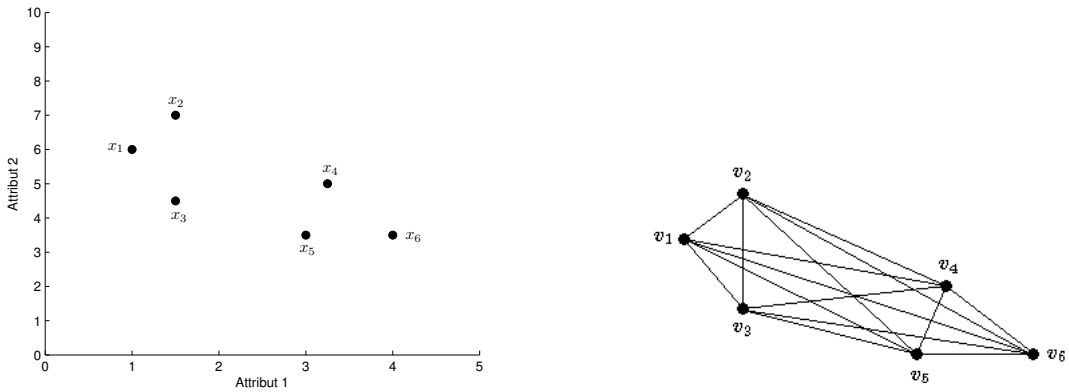
Afin d'illustrer les différentes types de graphes $G(V,E,W)$, nous choisissons un exemple simple. Soit un ensemble $\mathcal{X}$ composé de six objets et définis par deux attributs f_1 et f_2. La matrice de données X, définie dans $\mathbb{R}^{6\times 2}$, est alors représentée comme suit :

$$X = \begin{pmatrix} 1 & 6 \\ 1.5 & 7 \\ 1.5 & 4.5 \\ 3.25 & 5 \\ 3 & 3.5 \\ 4 & 3.5 \end{pmatrix} \tag{1.14}$$

Les graphes pondérés peuvent être séparés en deux catégories, suivant le nombre de connexions inter-objets :

- **Graphes totalement connectés**, pour lesquels chaque paire de noeuds distincts est connectée par un arc unique (tous les noeuds sont donc connectés entre eux). Ceci implique donc une dimension importante de l'ensemble E puisque pour un nombre N de noeuds, il existe $(N(N-1))/2$ arcs inter-noeuds (cf. figure 1.4) ;

- **Graphes partiellement connectés**, pour lesquels la notion de voisinage est intégrée. Ils sont divisés en deux sous-catégories :
  - *les graphes de rayon de voisinage* ε (appelés graphes ε-voisinage), où les noeuds v_i et v_j sont connectés si et seulement si la distance $d(i,j)$ entre les points x_i et x_j correspondants dans l'espace des données, est inférieure à un seuil ε donné et fixé par l'utilisateur (cf. figure 1.5(b)). Nous pouvons noter que plus la valeur de ε est importante, plus le

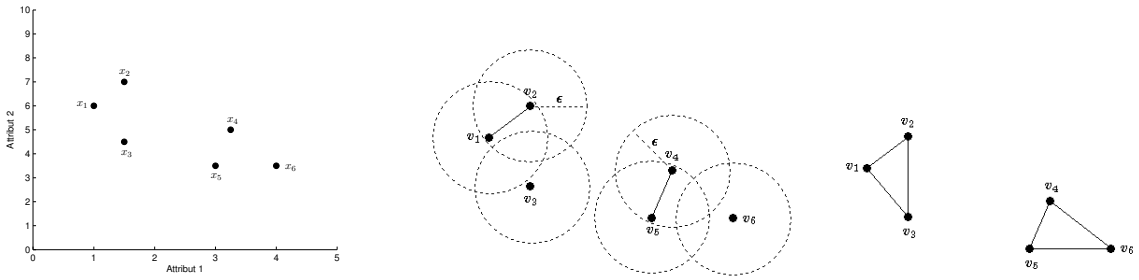


(a) Projection des données originales (b) Graphe totalement connecté

FIGURE 1.4 – Exemple de graphe totalement connecté.

nombre d'arcs de liaison inter-noeuds est élevé ;

– *les graphes des $\mathcal{K}$ plus proches voisins* (appelés graphes $\mathcal{K}$-PPV) où le noeud v_i est connecté au noeud v_j si et seulement si l'objet x_j appartient à l'ensemble des $\mathcal{K}$ plus proches voisins de l'objet x_i (en terme de distance dans l'espace des données) (cf. figure 1.5(c)). La variable $\mathcal{K}$ est un paramètre prenant une valeur entière fixée par l'utilisateur. Néanmoins, cette définition tend à s'approcher d'un graphe orienté, puisque les relations de voisinage ne sont pas toujours symétriques. De la même manière que pour le graphe ε, nous pouvons noter que plus la valeur du nombre de voisins $\mathcal{K}$ est importante, plus le nombre d'arcs de liaison inter-noeuds est élevé.



(a) Projection des données originales (b) Graphe ε-voisinage (c) Graphe $\mathcal{K}$-PPV ($\mathcal{K} = 2$)

FIGURE 1.5 – Exemple de graphe ε-voisinage et de graphe $\mathcal{K}$-PPV.

Les choix des paramètres ε et $\mathcal{K}$ dépendent de la structure des données et conditionnent les résultats de la classification. L'objectif de la classification est de diviser les données en groupes tels que les points d'un même groupe soient similaires et les points appartenant à des groupes

différents soient dissimilaires les uns des autres. Le graphe pondéré permet de reformuler ce problème de classification grâce, notamment, à la matrice de similarités W : il s'agit alors de trouver des coupes du graphe telle que les arcs inter-groupes aient des poids faibles (les points appartenant à des groupes différents sont dissimilaires), et les arcs intra-groupes aient des valeurs de poids élevées (les points d'un même groupe sont similaires).

Les caractéristiques d'un graphe

Dans le cas d'un graphe pondéré (orienté ou non), la matrice de similarités joue le rôle de matrice de poids des arcs inter-noeuds. En reprenant l'exemple 1.14, et en considérant disposer d'un graphe totalement connecté (comme représenté sur la figure 1.6(a)), la matrice W définie dans $\mathbb{R}^{6\times 6}$ et utilisant un noyau gaussien (avec $\sigma = 1$), est définie telle que :

$$W = \begin{pmatrix} 1.00 & 0.54 & 0.29 & 0.05 & 0.01 & 0.01 \\ 0.54 & 1.00 & 0.05 & 0.03 & 0.01 & 0.01 \\ 0.29 & 0.05 & 1.00 & 0.19 & 0.20 & 0.03 \\ 0.05 & 0.03 & 0.19 & 1.00 & 0.31 & 0.25 \\ 0.01 & 0.01 & 0.20 & 0.31 & 1.00 & 0.61 \\ 0.01 & 0.01 & 0.03 & 0.25 & 0.61 & 1.00 \end{pmatrix} \tag{1.15}$$

Cette matrice de similarités permet également d'exprimer et de calculer plusieurs caractéristiques du graphe $G(V,E,W)$:

– **Le degré d'un noeud** est défini comme étant égal à la somme des éléments de la ligne de la matrice de similarités W correspondant au noeud. Il est noté d_{ii} :

$$d_{ii} = \sum_{j=1}^{N} w_{ij} \tag{1.16}$$

Cette mesure correspond au poids total des connexions du noeud v_i. Dans l'exemple donné sur la figure 1.6(a), le degré du noeud v_1 est donc égal à la somme des poids des arcs représentés en rouge sur la figure 1.6(b), c'est à dire 0.90.

– **La matrice** $D = \mathbf{diag}\,(d_{11},\ldots,d_{NN})$, définie dans $\mathbb{R}^{N\times N}$, est une matrice diagonale telle que $D_{v_i,v_i} = d_{ii}$, désigne la matrice des degrés du graphe $G(V,E,W)$. Les éléments hors

diagonale sont alors égaux à 0. Cette matrice est représentée comme suit :

$$D = \begin{pmatrix} 0.90 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.64 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.76 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.83 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.91 \end{pmatrix} \tag{1.17}$$

– **La coupe du graphe**, quant à elle, a pour objectif de scinder le graphe $G(V,E,W)$ en deux groupes disjoints de noeuds, tout en minimisant les poids des arcs entre ces deux groupes.

$$Coupe(C_1,C_2) = Cut(C_1,C_2) = \sum_{\{i|v_i \in C_1\},\{j|v_j \in C_2\}} d_{ij}. \tag{1.18}$$

La valeur optimale de la coupe, pour notre exemple, est calculée en sommant les poids des liaisons inter-groupes (représentées en bleu sur la figure 1.6(c)), et est donc égale à 0.54.

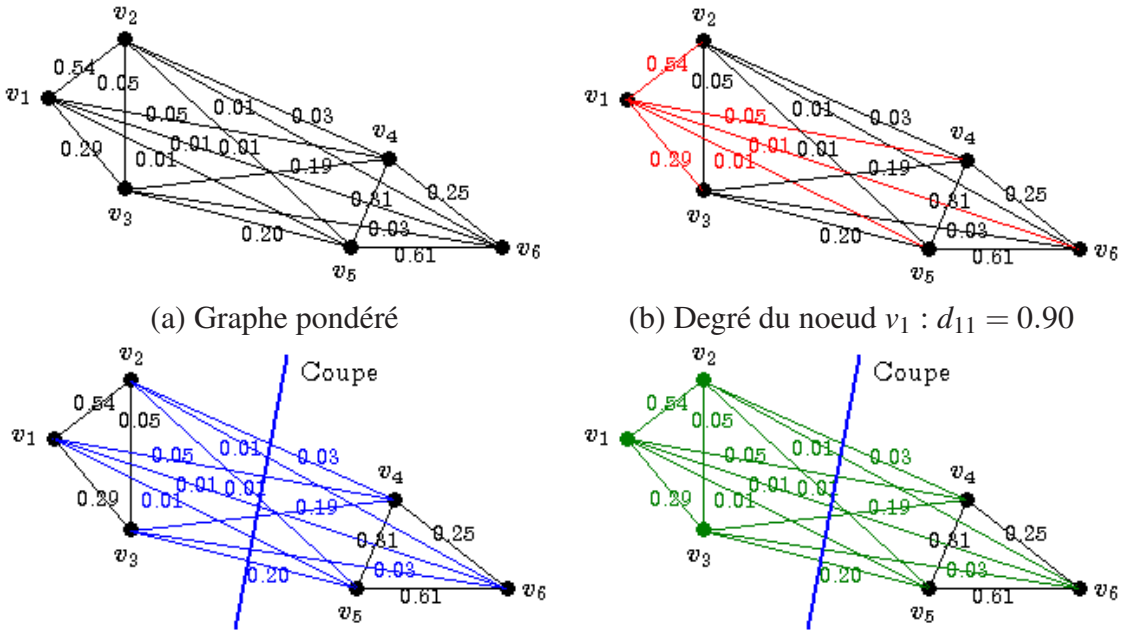
– **La structure interne** d'un sous-ensemble de noeuds, peut être définie de deux façons différentes :
  – le cardinal de C_1 : $|C_1|$,
  – le volume de C_1 :

$$vol(C_1) = \sum_{\{i|v_i \in C_1\}} \sum_{j=1}^{N} w_{ij} = \sum_{\{i|v_i \in C_1\}} d_{ii}, \tag{1.19}$$

Intuitivement, $|C_1|$ mesure la taille de C_1 en comptant le nombre de noeuds affectés à ce groupe (ici, $|C_1| = 3$ et $|C_2| = 3$), alors que $vol(C_1)$ mesure le volume de C_1 en sommant les poids de tous les arcs attachés aux noeuds affectés au groupe C_1 (représentés en vert sur la figure 1.6(d)). Les volumes des deux sous-ensembles C_1 et C_2 sont alors respectivement égaux à 2.30 et 2.88.

1.3 Représentation contextuelle des connaissances

Le contexte de classification est directement lié à la connaissance disponible [Cohn et al., 2003] [Wagstaff, 2002]. En effet, selon le type d'information à disposition, il est possible de distinguer trois contextes différents :



(a) Graphe pondéré

(b) Degré du noeud v_1 : $d_{11} = 0.90$

(c) Coupe du graphe (pour $k = 2$) = 0.54

(d) Volume du groupe C_1 : $vol(C_1) = 2.30$

FIGURE 1.6 – Illustration du calcul du degré des noeuds, de la coupe du graphe et du volume des sous-ensembles.

- un contexte non supervisé, où aucune information autre que la matrice de données originale X, n'est disponible ;
- un contexte supervisé, où l'ensemble des objets de la matrice de données X sont étiquetés. Les algorithmes de classification reçoivent en entrée les vecteurs données et forunissent, en sortie, les étiquettes correspondantes ;
- un contexte semi-supervisé occupant une place intermédiaire entre le contexte supervisé et le contexte non supervisé, comme illustrée sur la figure 1.7. Les données sont partiellement étiquetées.

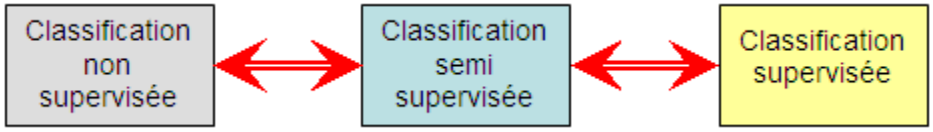


FIGURE 1.7 – Place de la classification semi-supervisée.

Par exemple, dans le cadre de notre application, l'objectif est de définir des ensembles de cellules phytoplanctoniques appartenant aux mêmes groupes fonctionnels. Ceux-ci peuvent alors être soumis à un expert dans le but d'apporter des connaissances additionnelles permettant l'identification des différentes cellules et/ou des différents groupes taxonomiques.

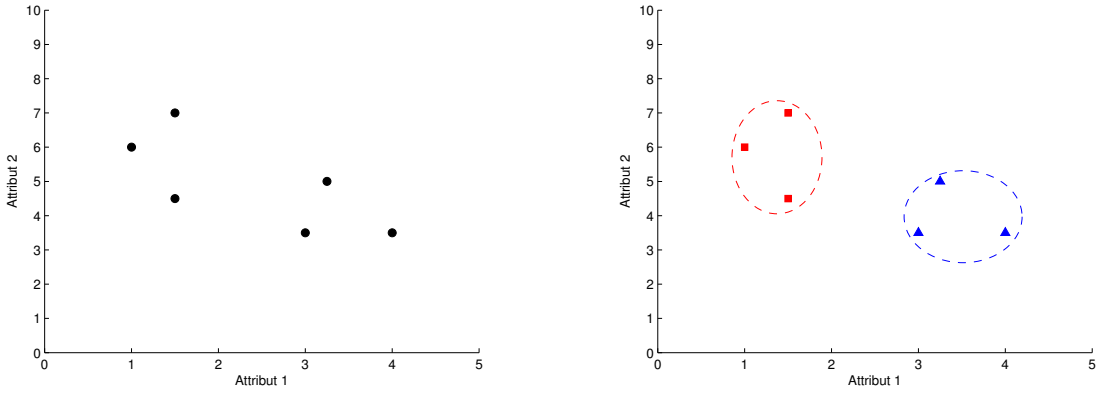
1.3.1 Contexte non supervisé

La classification non supervisée peut être définie comme une analyse exploratoire de la structure des données [Jain et al., 1999]. En effet, aucune information additionnelle autre que les données elles-mêmes, n'est fournie aux algorithmes de classification (cf. figure 1.8).

L'idée générale est donc d'organiser les données en groupes, telles que les objets d'un même groupe soient très similaires et que des objets de groupes différents ne le soient pas.

Sur la figure 1.8(a) sont représentés deux groupes d'objets bien séparés. Il est alors possible d'affecter visuellement les points de gauche à un groupe et ceux de droite à un deuxième groupe, comme montré sur la figure 1.8(b). Pour cela, l'algorithme de partitionnement des données cherche à satisfaire deux objectifs simultanément :

– *Objectif 1* : obtenir une cohérence importante au sein d'un même groupe ;
– *Objectif 2* : obtenir une séparation suffisamment discriminante des groupes.



(a) Données originales (b) Groupes obtenus

FIGURE 1.8 – Exemple de classification non supervisée (algorithme des K-moyennes).

Les algorithmes de classification non supervisée s'appuient sur des hypothèses implicites sur les classes des données :

– l'algorithme des K-moyennes : classes globulaires voire équiprobables, nombre de groupes K fixé ;
– la classification hiérarchique : les objets sont groupés de proche en proche ;
– les mélanges de densités : les classes sont des données de fortes densités de probabilités.

Malgré le succès d'utilisation des algorithmes de classification non supervisée dans différents domaines d'application, les résultats obtenus ne sont pas toujours interprétables. En effet,

pour les cellules phytoplanctoniques, il arrive que les classes fonctionnelles obtenues ne correspondent pas aux espèces réelles de phytoplancton.

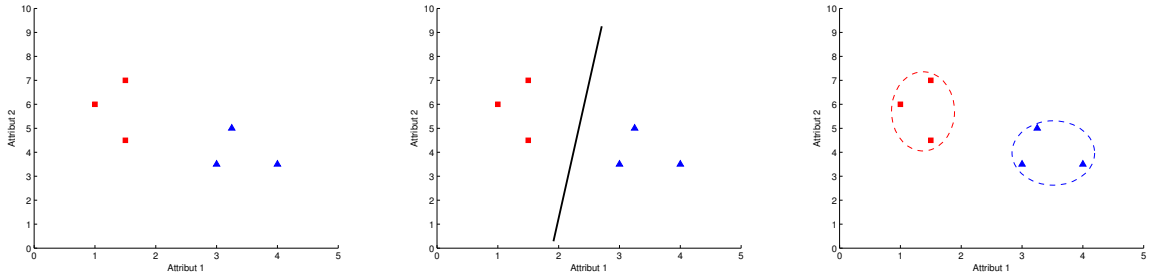
1.3.2 Contexte supervisé

Les algorithmes de classification supervisée diffèrent des algorithmes non supervisés de par le fait qu'ils disposent de connaissances *a priori* sous la forme d'ensembles d'objets étiquetés (au minimum, un objet étiqueté par classe), permettant de modéliser la relation entre les objets et les classes afin de pouvoir estimer la classe d'un nouvel objet.

Dans ce contexte, les classes sont donc prédéfinies et les étiquettes des classes sont disponibles (cf. figure 1.9). Cet ensemble d'étiquettes de classes est alors représenté par un vecteur $Y = [y_1, \ldots, y_i, \ldots, y_N]$, tel que $y_i \in \{1, \ldots, K\}$ représente l'étiquette de la classe de l'objet x_i (K étant le nombre total de classes).

L'ensemble d'apprentissage, composé de couples "(donnée, étiquette)", est utilisé afin de créer des règles de décisions. Ce sont ces règles qui sont utilisées afin de déterminer la classe d'appartenance d'un nouvel objet.

Par exemple, il est possible de définir une règle de classification pour l'ensemble d'apprentissage de la figure 1.9(a) permettant de classer correctement la totalité des objets étiquetés, comme illustré sur la figure 1.9(b). Celle-ci est représentée par la ligne noire (les points au dessus de la droite appartiennent à la classe 1, et les points en dessous appartiennent à la classe 2). Une fois cette règle établie, des objets sans étiquettes peuvent être étiquetés en suivant cette règle, comme montré sur la figure 1.9(c).



(a) Données originales (b) Règle de décision (c) Groupes obtenus

FIGURE 1.9 – Exemple de classification supervisée par extraction de règles de décision.

Dans la plupart des situations, ces règles de décision sont déterminées grâce à l'optimisation d'une fonction coût entre les étiquettes réelles de l'ensemble d'apprentissage et les étiquettes

estimées obtenues du classifieur.

Dans la littérature, certains auteurs proposent de considérer la classification supervisée sous l'angle des graphes. Ils proposent une méthode de construction originale de la matrice de similarités basée sur la connaissance des étiquettes de classes. Les noeuds v_i et v_j sont alors connectés (avec un poids de 1) si les objets x_i et x_j correspondants possèdent la même étiquette de classe, c'est à dire si $y_i = y_j$. La matrice W est donc définie telle que :

$$w_{ij} = \begin{cases} 1 & \text{si } y_i = y_j, \\ 0 & \text{sinon.} \end{cases} \tag{1.20}$$

Les classifieurs supervisés les plus fréquemment utilisés dans la littérature sont : l'analyse discriminante, l'algorithme des $\mathcal{K}$ plus proches voisins (noté $\mathcal{K}$-PPV), le perceptron multicouches (noté MLP), les séparateurs à vastes marges (notés SVM), etc.

Cependant, malgré le succès des classifieurs supervisés pour l'apprentissage, ils connaissent quelques lacunes.

1. La qualité des performances du classifieur est très liée à la qualité de l'étiquetage des données dans la base d'apprentissage. De plus, un nombre important de données étiquetées est nécessaire au bon fonctionnement des classifieurs.

2. Les méthodes de classification ne possèdant pas la même capacité de stockage d'informations, l'importance relative accordée à l'ensemble d'apprentissage et à l'ensemble des données nouvelles peut alors poser deux problèmes, selon la structure utilisée :
   – le "sur-apprentissage" : l'algorithme donne trop d'importance à la structure des données dans la base d'apprentissage et n'autorise que peu de degrés de liberté sur les règles de décision obtenues. Il perd alors ses pouvoirs de prédiction pour de nouveaux objets ;
   – le "sous-apprentissage" : l'algorithme ne se focalise pas (ou peu) sur la structure des données de l'ensemble d'apprentissage. Les degrés de liberté sur les règles de décision obtenues sont alors importantes.

3. Il est nécessaire de recourir à un expert du domaine, afin de créer manuellement une base d'objets étiquetés, de dimensions suffisamment importantes. En effet, l'étiquetage individuel des données est une tâche complexe, fastidieuse et parfois impossible à réaliser. Dans le cadre de notre application, la technique de référence pour déterminer la composition

phytoplanctonique d'un échantillon marin est le comptage en microscopie optique à inversion après sédimentation, dite technique d'Utermöhl [Lund et al., 1958]. Cette méthode nécessite l'élaboration d'un processus précis d'observation mais également des connaissances solides en taxonomie.

En résumé, les besoins en termes de connaissances *a priori* des méthodes de classification supervisée et non supervisée ne sont pas identiques. Si les algorithmes supervisés requièrent l'utilisation d'un ensemble d'apprentissage, pour lequel chaque objet est étiqueté, les algorithmes non supervisés travaillent uniquement sur la matrice de données originale et ne nécessitent aucune autre information additionnelle.

1.3.3 Contexte semi-supervisé

Le besoin croissant d'automatiser les tâches requérant une expertise humaine et la complexité de l'étiquetage manuel d'objets, ont conduit à l'utilisation de méthodes semi-supervisées pour lesquelles une faible quantité d'informations est fournie. En effet, dans les applications réelles comme celle présentée dans ce mémoire, il est fréquent d'avoir à disposition beaucoup de cellules sans étiquettes et peu de cellules étiquetées [Chapelle et al., 2006][Zapien, 2009].

Dans la littérature, la majorité des auteurs traitent ces problèmes en adaptant des procédures de groupement classiques et/ou des procédures d'optimisation grâce à l'introduction de connaissances. L'information contenue dans celles-ci peut alors prendre plusieurs formes :

- information structurelle sur les données (contraintes globales),
- capacité minimum ou maximum des groupes d'objets (contraintes de groupes),
- étiquetage partiel et contraintes de comparaisons (contraintes d'objets).

Les contraintes d'objets définissent des restrictions sur des paires d'objets. Ce type de connaissances *a priori* sur les données est généralement fourni sous trois formes :

- le retour d'informations,
- les étiquettes de classes,
- les contraintes de comparaison par paires d'objets.

Dans le cadre de l'application à l'identification de cellules phytoplanctoniques, nous nous focalisons sur les contraintes d'objets. En effet, les informations concernant la structure spatiale des données et la capacité (minimum ou maximum) des groupes souhaités ne peuvent être disponibles *a priori*, en raison de la forte variabilité des cellules phytoplanctoniques au sein d'une même espèce. Les contraintes de comparaison entre paires de cellules et l'analyse des

résultats obtenus par retour d'informations auprès de l'expert en biologie marine, semblent donc représenter les meilleures alternatives.

Retour d'informations

Les systèmes de classification intéractifs adoptent une approche itérative, dans laquelle le système produit une classification des données, puis la présente à un expert afin de l'évaluer et la valider [Cohn et al., 2003]. Cet expert peut alors indiquer clairement les erreurs de groupement (induits par le classifieur) grâce notamment à des outils graphiques de visualisation. Puis, cette information peut être utilisée lors de l'itération suivante de l'algorithme, comme illustré sur la figure 1.10.

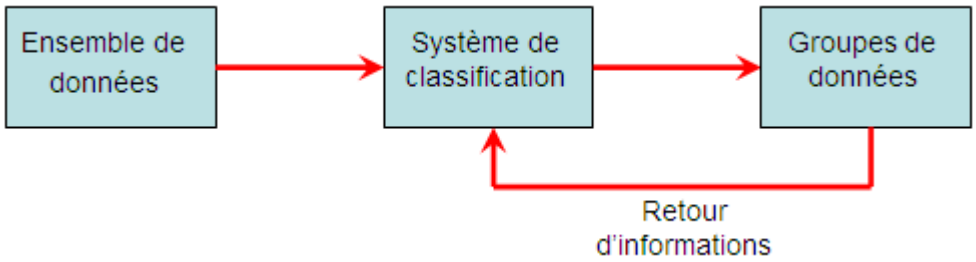


FIGURE 1.10 – Schéma du système de classification utilisant la méthode de retour d'informations.

Cependant, l'inconvénient lié à cette méthode réside dans le fait qu'un expert peut se retrouver en difficulté lors de la validation des résultats si l'ensemble des données est de dimension importante.

Étiquetage partiel

L'étiquetage des objets d'un ensemble de grande dimension représente une tâche difficile et coûteuse en temps ce qui la rend souvent infaisable. En revanche, il est possible d'étiqueter un sous-ensemble ne contenant que quelques objets, comme montré sur la figure 1.11(a). L'ensemble des données $\mathcal{X}$ peut alors être divisé en deux sous-ensembles, tels que $\mathcal{X} = \{\mathcal{X}_l \cup \mathcal{X}_u\}$ (avec $N = l + u$). $\mathcal{X}_l$ représente le sous-ensemble des l données étiquetées auquel est associé le vecteur $Y = [y_1, \ldots, y_i, \ldots, y_l]$, tel que $y_i \in \{1, \ldots, K\}$ représente l'étiquette affectée à l'objet x_i et K le nombre d'étiquettes différentes connues. $\mathcal{X}_u$ est, quant à lui, le sous-ensemble des u données non étiquetées.

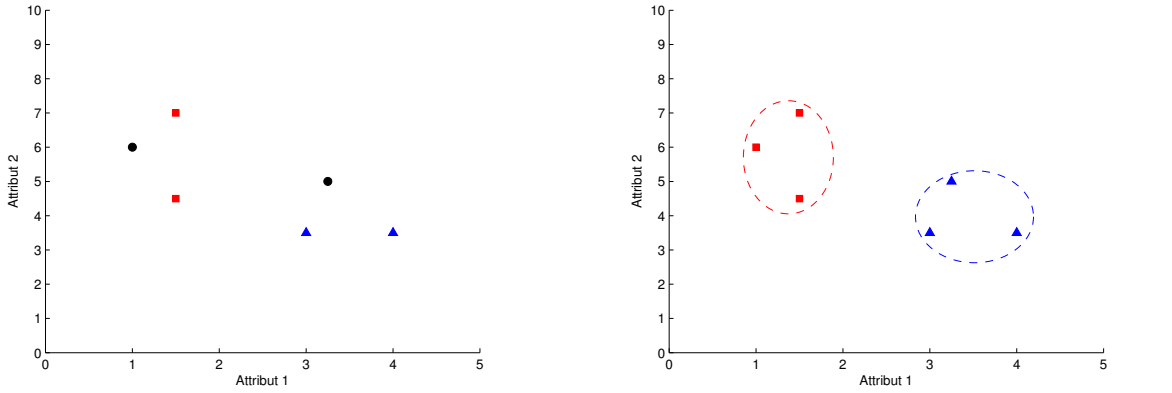
De même que pour le contexte supervisé, il est possible de fournir une alternative pour la construction de la matrice de similarités afin de prendre en considération les données non

étiquetées mais également les informations *a priori*. Les noeuds v_i et v_j sont alors connectés avec :

- un poids de 1, si les objets x_i et x_j correspondants appartiennent tous deux à $\mathcal{X}_l$ et s'ils possèdent la même étiquette de classe ($y_i = y_j$) ;
- un poids de $\exp(-\frac{d^2(i,j)}{2\sigma^2})$, si les objets x_i et/ou x_j correspondants n'appartiennent pas à $\mathcal{X}_l$.

La matrice W est donc définie telle que :

$$w_{ij} = \begin{cases} 1 & \text{si } (x_i, x_j) \in (\mathcal{X}_l \times \mathcal{X}_l) \text{ et } y_i = y_j, \\ 0 & \text{si } (x_i, x_j) \in (\mathcal{X}_l \times \mathcal{X}_l) \text{ et } y_i \neq y_j, \\ \exp(-\frac{d^2(i,j)}{2\sigma^2}) & \text{sinon.} \end{cases} \tag{1.21}$$



(a) Données originales (b) Groupes obtenus par l'algorithme du PPV

FIGURE 1.11 – Contexte semi-supervisé avec étiquetage partiel (algorithme du PPV).

Cet étiquetage partiel peut alors être utilisé de deux façons différentes :

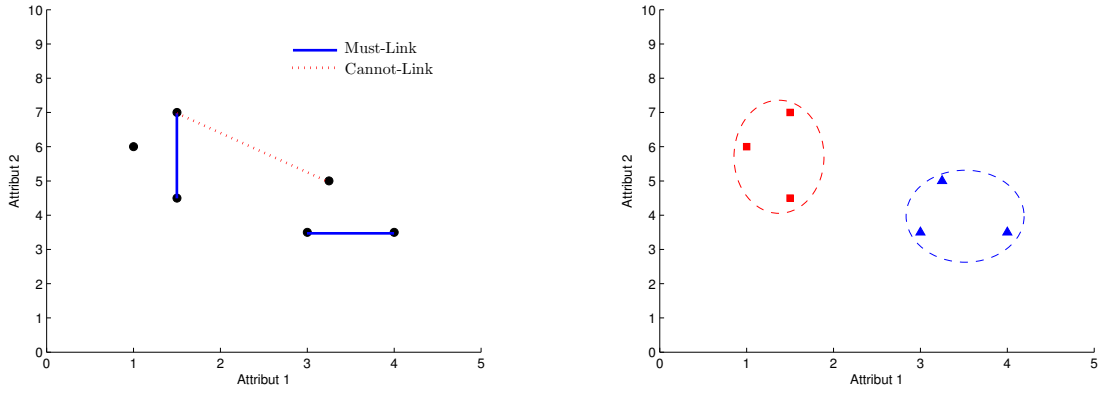
- identification des groupes obtenus (cf. figure 1.11(b)) : après avoir appliqué un algorithme de classification non-supervisée, les groupes d'objets obtenus peuvent être identifiés grâce au sous-ensemble d'objets étiquetés au préalable. Pour cela, l'utilisation de règles simples est requise. Parmi ces dernières, le vote majoritaire semble être une bonne alternative puisqu'il permet d'affecter une identité (ou une étiquette) à chaque groupe obtenu ;
- initialisation intelligente des algorithmes de partitionnement utilisés : l'initialisation de certains algorithmes de classification non-supervisée est une étape importante et peut se révéler cruciale dans la validation et l'interprétation des résultats de partitionnement obtenus. C'est pourquoi, il semble intéressant d'utiliser l'information contenue dans ces étiquettes de classes afin d'orienter le choix des paramètres initiaux des algorithmes de

classification.

Relation entre paires d'objets

Les contraintes de relation entre paires d'objets ont été récemment introduites par Wagstaff et al. [Wagstaff, 2002]. Si l'étiquetage des objets se révèle être une tâche longue et complexe, les contraintes de comparaison par paires, attestant simplement que deux objets doivent être dans la même classe (contraintes Must-Link) ou non (contraintes Cannot-Link), sont par contre plus aisées à recueillir auprès d'experts (cf. figure 1.12). Il s'agit alors, pour ces derniers, de construire deux sous-ensembles de paires d'objets :

- pour les contraintes de type "Must-Link" : un sous-ensemble $\mathcal{ML}$ composé de $|\mathcal{ML}|$ paires d'objets $\{x_i, x_j\}$ devant appartenir à un même groupe (avec $\{x_i, x_j\} \subseteq \mathcal{X}$) ;
- pour les contraintes de type "Cannot-Link" : un sous-ensemble $\mathcal{CL}$ composé de $|\mathcal{CL}|$ paires d'objets $\{x_i, x_j\}$ devant appartenir à des groupes différents (avec $\{x_i, x_j\} \subseteq \mathcal{X}$) ;



(a) Données originales (b) Groupes obtenus

FIGURE 1.12 – Contexte semi-supervisé avec contraintes de comparaison par paires d'objets ("Must-Link" : lignes bleues, "Cannot-Link" : pointillés rouge).

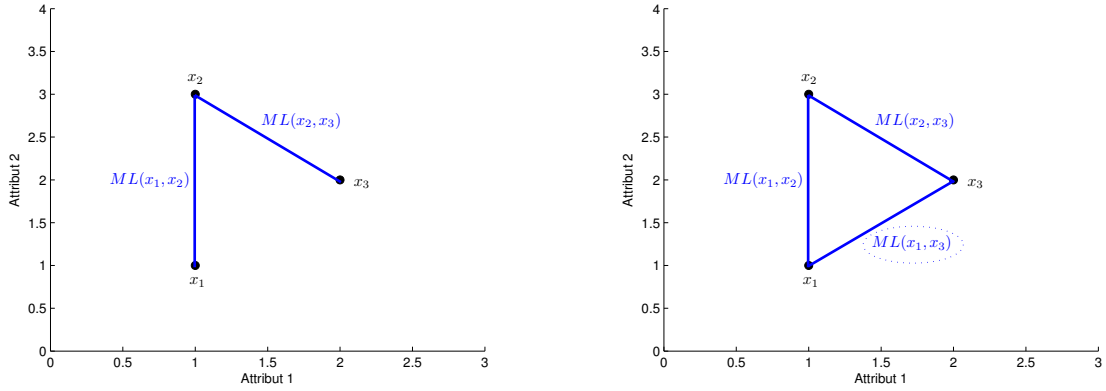
Cette manière de représenter l'information relationnelle est souvent considérée comme la plus générale, car elle permet de reprendre une grande partie des contraintes précédemment citées. Ces contraintes relationnelles peuvent être rangées dans deux catégories :

- les contraintes "dures" qui doivent être respectées dans la partition en sortie de l'algorithme : une contrainte de type "Must-Link" sur une paire d'objets indique que ces objets doivent être dans le même groupe dans la partition finale, alors qu'une contrainte de type "Cannot-Link" indique que ces objets ne doivent en aucun cas se trouver dans le même groupe ;

– les contraintes ”souples” qui ne doivent pas être obligatoirement respectées dans la partition finale des données : l'utilisateur peut alors indiquer une valeur correspondant à un degré de croyance sur chacune des contraintes. Ces dernières sont alors, généralement, considérées comme des ”préférences”.

Une fois les paires d'objets en ”Must-Link” et ”Cannot-Link” collectées, il est possible de propager ces contraintes : par transitivité pour les contraintes ”Must-Link” et par héritage pour les contraintes ”Cannot-Link”.

Transitivité des contraintes ”Must-Link”. Les contraintes de comparaison de type ”Must-Link” définissent une relation d'équivalence entre objets. Elles sont donc transitives. Comme montré sur la figure 1.13(a) avec trois points, si deux contraintes ”Must-Link” sont définies telles que $ML(x_1,x_2)$ et $ML(x_2,x_3)$, alors il est possible de propager ces informations par transitivité en dérivant la contrainte $ML(x_1,x_3)$ (cf. figure 1.13(b)).



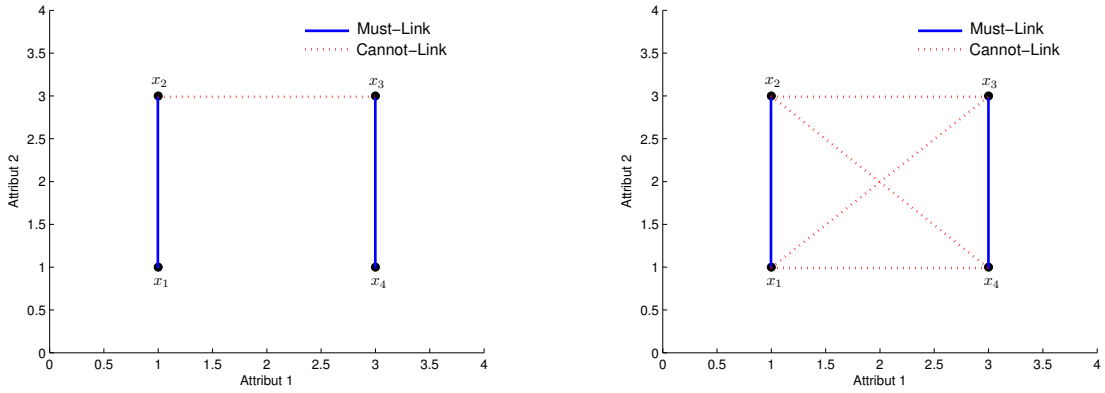
(a) Contraintes définies (b) Contrainte dérivée

FIGURE 1.13 – Propagation des contraintes ”Must-Link” par transitivité.).

Nous obtenons alors la propriété suivante : soient U et U' deux ensembles de contraintes ”Must-Link” (notés $ML(U)$ et $ML(U')$). Si $a \in U$ et $a \in U'$, alors il est possible d'obtenir $ML(U \bigcup U')$.

Héritage des contraintes ”Cannot-Link”. Les contraintes de type ”Cannot-Link” sont symétriques mais non transitives. En effet, il est impossible de prédire la relation liant deux objets x_i et x_l, sachant que les paires d'objets (x_i,x_j) et (x_j,x_l) sont liées par deux contraintes ”Cannot-Link” indépendantes. Cependant, la propagation de ces contraintes est rendue possible grâce à

la règle de l'héritage, comme illustré sur la figure 1.14 avec quatre points et trois contraintes de comparaison.



(a) Contraintes définies (b) Contraintes dérivées

FIGURE 1.14 – Propagation des contraintes "Cannot-Link" par héritage.).

La figure 1.14(a) montre que les objets x_1 et x_2, ainsi que x_3 et x_4 sont appariés en "Must-Link". Si une contrainte $CL(x_2, x_3)$ est définie, alors il est possible de dériver trois contraintes supplémentaires de type "Cannot-Link" par héritage ($CL(x_1, x_3)$, $CL(x_1, x_4)$ et $CL(x_2, x_4)$), comme montré sur la figure 1.14(b). La propriété de l'héritage est donc la suivante : soient U et U' deux groupes d'objets appariés en "Must-Link". Si une contrainte $CL(a \in U, b \in U')$ existe, alors $CL(x, y)$, $\forall x \in U$ et $\forall y \in U'$.

Les algorithmes de classification semi-supervisée fréquemment cités dans la littérature, sont basés sur des algorithmes non supervisés : l'algorithme des K-moyennes semi-supervisé (noté SS-KM) [Bradley et al., 2000], l'algorithme des COP K-moyennes (noté COP-KM) [Wagstaff et al., 2001], etc.

1.4 Conclusion

Dans ce chapitre, la représentation des données et les relations de similarités (ou de distances) ont été mises sous la forme de matrices et de graphes. Nous avons abordé la notion de représentation numérique des objets caractérisés par un ensemble d'attributs, mais également la notion de relation entre objets. Nous avons donc choisi de présenter les différentes métriques de comparaison entre objets afin de pouvoir caractériser et mesurer la proximité d'un objet par rapport aux autres. La fonction similarité gaussienne RBF basée sur la distance euclidienne, se distingue par le fait qu'elle utilise un paramètre de dispersion permettant d'intégrer la notion de

structure de voisinage. Nous avons introduit la similarité par appariement élastique pour quantifier la comparaison entre signaux de tailles différentes et présentant des distorsions temporelles.

Les données et les relations inter-objets peuvent également être représentées grâce à un graphe construit à partir de l'ensemble $\mathcal{X}$ et de la matrice de similarités W. Il existe plusieurs types de graphes pouvant prendre en considération la notion de voisinage et permettant de mettre en évidence la structure locale des données.

Nous avons présenté les différents contextes de classification suivant le type d'information *a priori* à disposition. Si la classification supervisée et non supervisée sont diamètralement opposées de par la quantité de connaissances *a priori* sur l'appartenance des objets à chaque classe, et sachant que l'étiquetage individuel des données peut s'avérer complexe et coûteux en temps, la classification semi-supervisée semble être plus réaliste en intégrant une faible quantité d'informations *a priori* telles que la structure, la forme, la capacité des groupes ou encore les relations entretenues par les objets.

A partir de la représentation des données en graphes auxquels sont associées des matrices de similarités, la recherche de classes devient un problème de coupe de graphes qui est traité par les algorithmes de classification spectrale, objet du chapitre suivant.

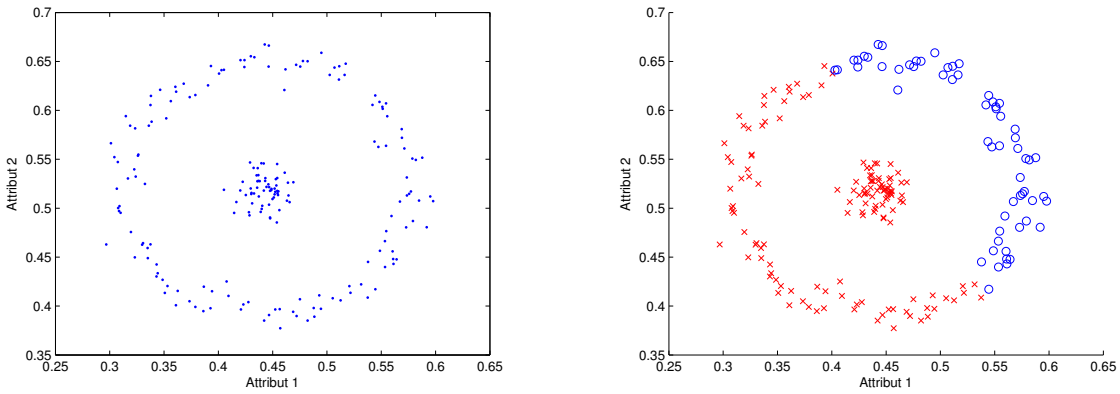
Chapitre 2

Graphes et algorithmes de classification spectrale

2.1 Introduction

La classification spectrale est une technique issue de la théorie des graphes et de l'analyse numérique. Elle présente un fort lien avec les machines à noyaux, de par le fait que ces deux méthodes s'appuient sur la définition d'un espace de projection non linéaire dans lequel s'effectue la classification [Dhillon, 2004]. Elle est de plus en plus usitée, à la fois en raison de son efficacité, mais également de sa simplicité relative d'implémentation qui se résume en l'extraction du spectre (valeurs et vecteurs propres) d'une matrice de similarités conçue à partir d'un ensemble de données [Ng et al., 2002]. Contrairement aux algorithmes traditionnels de classification non supervisée comme celui des K-moyennes, la méthode de classification spectrale offre l'avantage de traiter des ensembles de données de structures complexes "non globulaires" et non linéairement séparables, comme illustré sur la figure 2.1. Dans le cadre de l'identification de cellules phytoplanctoniques, cette technique semble donc appropriée puisque les groupes fonctionnels de cellules ne sont pas toujours linéairement séparables et permet de prendre en considération les relations entre les profils des cellules.

Par ailleurs, cette méthode permet de considérer l'angle de classification comme un problème de coupe de graphe, sans émettre d'hypothèses sur la forme globale des objets. Il existe plusieurs techniques de classification spectrale qui optimisent des fonctions coûts (ou fonctions objectifs) différentes à partir du graphe de données. Ces techniques sont, pour la majorité, basées sur la minimisation d'un critère de type coupe de graphe. Les fonctions les plus fréquemment utilisées dans la littérature sont : la coupe minimale, la coupe ratio, la coupe min-max ou encore



(a) Données originales (b) Groupes obtenus

FIGURE 2.1 – Illustration de la limite de la méthode des K-moyennes.

la coupe normalisée [Luxburg, 2007].

Dans ce chapitre, nous présentons différents algorithmes de classification spectrale. Nous donnons, tout d'abord, des détails sur la théorie spectrale des graphes et montrons que tout graphe peut se résumer en une matrice Laplacienne. Nous présentons alors les propriétés mathématiques de cette dernière mais également celles de ses variantes normalisées. Dans un second temps, nous montrons que le partitionnement des données peut s'apparenter à la recherche de la coupe optimale du graphe prédéfini. Nous distinguons ici deux cas : le cas où le nombre de groupes K recherchés est égal à deux et le cas où ce nombre est supérieur à deux. Dans la littérature, les problèmes les plus fréquemment traités sont ceux à $K = 2$. Il est possible de trouver deux types de méthodes permettant de traiter les cas où $K > 2$: les méthodes directes et celles récursives. Ces dernières ont pour objectif d'affiner récursivement les groupes d'objets obtenus à l'itération précédente. Il nous est alors possible de décrire quelques algorithmes permettant d'optimiser le critère de coupe afin de trouver la partition qui se rapproche le plus de celle optimale. Un exemple illustratif est fourni afin de mettre en évidence les caractéristiques de chaque méthode présentée.

2.2 Théorie spectrale des graphes

La classification spectrale est une méthode de classification en K groupes, basée sur le spectre de la matrice de similarités. Elle repose sur la minimisation d'un critère de type *Coupe Simple* à $K = 2$ [Shi and Malik, 2000], ou de type *Coupe Multiple* à $K \geq 2$ [Meila and Shi, 2000] [Ng et al., 2002]. Nous distinguons dans cette partie, les cas où le nombre de classes est égal à

deux ($K = 2$) et où le nombre de classes est strictement supérieur à deux ($K > 2$).

2.2.1 Matrices Laplaciennes des graphes

L'outil principal pour l'analyse spectrale de graphes associés aux données est la matrice Laplacienne [Chung, 1997]. Dans la littérature, plusieurs Laplaciens sont définis et regroupés dans le tableau 2.1. Nous supposons que le graphe de données $G(V,E,W)$ est un graphe non orienté et pondéré.

Matrices Laplaciennes	**Équations**
Non normalisée	$L = D - W$
Normalisée asymétrique	$\bar{L}_1 = I - D^{-1}W$
Normalisée symétrique	$\bar{L}_2 = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$
Normalisée par d_{max}	$\bar{L}_3 = \frac{(W+d_{max}I-D)}{d_{max}}$

TABLE 2.1 – Les différentes techniques de calcul de la matrice Laplacienne.

Matrice Laplacienne non normalisée.

La matrice Laplacienne non normalisée est une combinaison linéaire de la matrice de similarités W et de la matrice de degrés D (cf. chapitre 1, équation 1.16). Elle est définie comme suit :

$$L = D - W. \tag{2.1}$$

Mohar et Chung ont établi une liste de propriétés mathématiques de cette matrice, dont les principales sont [Mohar, 1997] :

1. Forme quadratique : soit f un vecteur dans $\mathbb{R}^N$; f_i et f_j deux composantes de f.

$$f^T L f = \frac{1}{2}\sum_{i,j=1}^{N}(f_i - f_j)^2 w_{ij}, \tag{2.2}$$

2. L est symétrique et semi-définie positive, c'est-à-dire que toutes ses valeurs propres sont positives ou nulles ($\lambda_i \geq 0$),
3. Solution triviale : la plus petite valeur propre de L est nulle, et le vecteur propre correspondant est le vecteur constant unitaire $\mathbb{1} = (1,\ldots,1)^T$,
4. L possède N valeurs propres réelles non-négatives, telles que $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$.

Nous pouvons remarquer que la matrice L non normalisée ne dépend pas des éléments sur la diagonale de la matrice de similarités W. En effet, les valeurs d'auto-similarité (c'est à dire la similarité d'un objet avec lui-même) ne modifient pas la matrice Laplacienne L. Ainsi, toute autre matrice de similarités composée de valeurs hors diagonales, égales à celles de W, permet d'obtenir la même matrice L.

Matrice Laplacienne normalisée.

Dans la littérature, la matrice Laplacienne normalisée est généralement définie de deux manières différentes [Meila and Shi, 2000] [Luxburg, 2007] :

– la matrice Laplacienne normalisée asymétrique (issue de la méthode de la marche aléatoire) [Meila and Shi, 2000] :

$$\bar{L}_1 = D^{-1}L = I - D^{-1}W. \tag{2.3}$$

– la matrice Laplacienne normalisée symétrique [Shi and Malik, 2000] :

$$\bar{L}_2 = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}, \tag{2.4}$$

Ces deux matrices Laplaciennes normalisées sont étroitement liées l'une à l'autre. Les principales propriétés que doivent respecter ces matrices sont :

1. Forme quadratique :

$$\forall f \in \mathbb{R}^N, f^T\bar{L}_2 f = \frac{1}{2}\sum_{i,j=1}^{N}(\frac{f_i}{\sqrt{d_{ii}}} - \frac{f_j}{\sqrt{d_{jj}}})^2 w_{ij}, \tag{2.5}$$

2. $\bar{L}_1$ et $\bar{L}_2$ sont semi-définies positives,
3. Solution triviale : 0 est une valeur propre de $\bar{L}_1$ associée au vecteur propre constant $\mathbb{1}$, 0 est une valeur propre de $\bar{L}_2$ associée au vecteur propre unitaire constant $D^{-\frac{1}{2}}\mathbb{1}$,
4. $\bar{L}_1$ et $\bar{L}_2$ possèdent N valeurs propres réelles non-négatives, telles que $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$.

Récemment, Kamvar et al. [Kamvar et al., 2003] ont utilisé la normalisation suivante :

$$\bar{L}_3 = \frac{(W + d_{max}I - D)}{d_{max}}. \tag{2.6}$$

avec d_{max} étant la valeur maximale des sommes pour chacune des lignes de la matrice de similarités W (autrement dit, le maximum de la diagonale de la matrice de degrés D). L'avantage

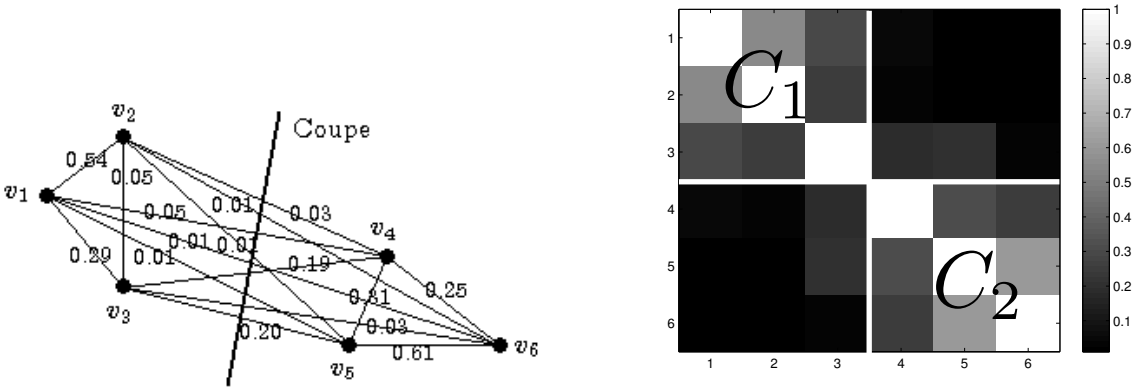
de ce type de normalisation réside dans le fait que les composantes de cette matrice $\bar{L}_3$ sont très sensibles aux valeurs absolues des similarités. Néanmoins, Xu et al. montrent l'inconvénient lié à cette normalisation [Xu et al., 2005] : les éléments non nuls de la diagonale de cette matrice tendent à générer des groupes ne contenant qu'un seul et unique objet lorsque des donnée aberrantes (''outliers'') sont présentes dans la base de données.

2.2.2 Bi-coupe de graphe ($K = 2$)

Dans la théorie des graphes, il existe plusieurs fonctions objectifs différentes : la coupe minimale (notée *MinCut*), la coupe ratio (notée *RatioCut*), la coupe normalisée (notée *NCut*), la coupe min-max (notée *MinMaxCut*), etc. L'objectif est de couper le graphe $G(V,E,W)$ en deux sous-ensembles disjoints de noeuds, C_1 et C_2, grâce à l'optimisation d'une de ces fonctions. Les groupes d'objets C_1 et C_2 sont définis tels que :

1. $C_1 \cup C_2 = \mathcal{X}$,
2. $C_1 \cap C_2 = \emptyset$.

La figure 2.2(b) représente la matrice de similarités ordonnée par groupes obtenus, issue du graphe 2.2(a). Les blocs diagonaux (désignés par C_1 et C_2) correspondent aux similarités intra-groupe pour l'ensemble C_1 et l'ensemble C_2, respectivement. Les zones hors blocs diagonaux (désignées par les couleurs grise et noire) correspondent, quant à elles, aux similarités inter-groupes.



(a) Coupe du graphe (b) Matrice de similarités ordonnée

FIGURE 2.2 – Exemple de coupe de graphe et matrice de similarités ordonnée pour deux groupes C_1 et C_2.

Etant donnés deux sous-ensembles de noeuds disjoints C_1 et C_2 du graphe $G(V,E,W)$, il est possible de définir les deux termes suivants :

– la coupe inter-groupes, définie par :

$$Cut(C_1,C_2) = \sum_{v_i \in C_1, v_j \in C_2} w_{ij}. \tag{2.7}$$

Elle est calculée en sommant les poids sur les arcs de connexions entre les deux sous-ensembles C_1 et C_2, ou autrement dit en sommant les similarités inter-groupes (zones bleues de la matrice de similarités représentée sur la figure 2.2(b)). Sur l'exemple de la figure 2.2(a), cette valeur est de 0.54 ;

– la coupe intra-groupe (également appelée "association"), définie par :

$$Cut(C_1,C_1) = \sum_{v_i \in C_1, v_j \in C_1} w_{ij}. \tag{2.8}$$

Elle est calculée en sommant les poids sur les arcs de connexions entre les objets du sous-ensemble C_1 (ou C_2), ou autrement dit en sommant les similarités intra-groupe (zones rouges de la matrice de similarités représentée sur la figure 2.2(b)). Les coupes intra-groupes pour l'exemple proposé sont : $Cut(C_1,C_1) = 0.88$ et $Cut(C_2,C_2) = 1.17$.

Comme montré dans le chapitre 1, le degré d_{ii} du noeud v_i est défini comme étant égal à $d_{ii} = \sum_{j=1}^{N} w_{ij}$. Afin de caractériser la structure interne d'un sous-ensemble, deux mesures sont alors définies :

– le cardinal de C_1 : $|C_1|$,

– le volume de C_1 :

$$vol(C_1) = \sum_{\{i|v_i \in C_1\}} \sum_{j=1}^{N} w_{ij} = \sum_{\{i|v_i \in C_1\}} d_{ii}, \tag{2.9}$$

Intuitivement, $|C_1|$ mesure la taille de C_1 en comptant le nombre de noeuds affectés à ce sous-ensemble (ici, $|C_1| = 3$ et $|C_2| = 3$), alors que $vol(C_1)$ mesure la taille de C_1 en sommant les poids de tous les arcs attachés aux noeuds affectés au sous-ensemble C_1 ($vol(C_1) = 2.30$ et $vol(C_2) = 2.88$).

La coupe minimale

La fonction objectif de la méthode de la coupe minimale est de trouver deux sous-graphes et, par suite, deux sous-ensembles d'objets C_1 et C_2 pour lesquels la somme des poids des connexions entre ces sous-ensembles est minimale.

La fonction objectif de cette méthode est alors relativement simple et est définie par :

$$J_{MinCut}(C_1, C_2) = Cut(C_1, C_2). \tag{2.10}$$

Cependant, la minimisation de la coupe ne produit pas toujours des partitions naturelles. En effet, dans cette méthode, la taille des sous-ensembles d'objets n'est pas prise en considération, ce qui peut conduire à des solutions aberrantes, comme montré sur la figure 2.3 (ligne pointillée). Le principal inconvénient de cette méthode, illustré par cet exemple, est le partitionnement des données en sous-ensembles déséquilibrés (bien que la coupe obtenue soit celle présentant une valeur minimale égale à 0.046) (cf. tableau 2.2). Cela s'explique principalement par le fait que les sous-ensembles ou les noeuds isolés ont un degré faible.

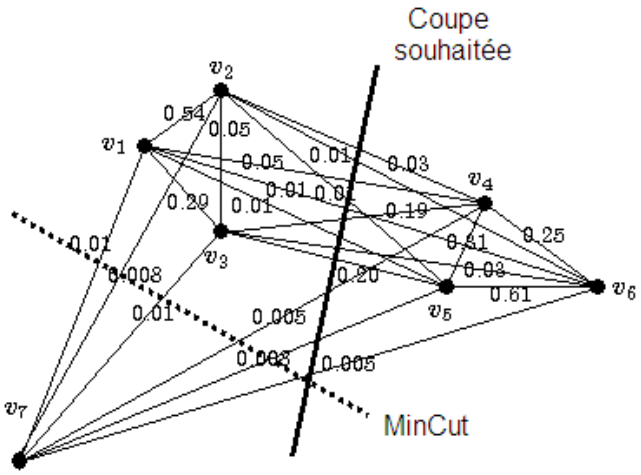


FIGURE 2.3 – Mise en évidence du problème lié au critère de MinCut (ligne pointillée : coupe obtenue, ligne pleine : coupe souhaitée).

Coupe souhaitée	Coupe obtenue
0.558	0.046

TABLE 2.2 – Valeurs de coupe souhaitée (ligne noire pleine) et coupe obtenue (ligne noire pointillée).

La minimisation de ce type de coupe est donc très sensible aux noeuds isolés. Une solution est de normaliser la valeur de la coupe par le nombre des éléments des sous-ensembles [Chen et al., 2005].

La coupe ratio

Hagen et Kahng [Hagen and Kahng, 1992] ont introduit un critère de coupe permettant de prendre en compte le cardinal des sous-ensembles. Celui-ci est égal au nombre de noeuds (ou

d'objets) associés à un sous-ensemble (ou groupe) donné. La fonction objectif est alors :

$$J_{RatioCut}(C_1,C_2) = Cut(C_1,C_2) \times (\frac{1}{|C_1|} + \frac{1}{|C_2|}). \tag{2.11}$$

Le critère introduit le cardinal des sous-ensembles dans le but d'obtenir des sous-ensembles équilibrés. Cependant, ce critère ne prend pas en considération les connexions intra-groupes. Pour pallier à ce problème, il est nécessaire d'utiliser un critère traitant à la fois les connexions inter-groupes et les connexions intra-groupes.

La coupe min-max

L'objectif de ce critère de coupe min-max est double :
- la minimisation des connexions inter-groupes, qui revient à minimiser le terme $Cut(C_1,C_2)$
- la maximisation des connexions intra-groupes, qui revient donc à maximiser les termes $Cut(C_1,C_1)$ et $Cut(C_2,C_2)$.

Afin de satisfaire ces deux objectifs simultanément, le critère de coupe min-max est défini comme suit [Ding et al., 2001] :

$$J_{MinMaxCut}(C_1,C_2) = Cut(C_1,C_2) \times (\frac{1}{Cut(C_1,C_1)} + \frac{1}{Cut(C_2,C_2)}). \tag{2.12}$$

Cette mesure exprime la cohésion d'un sous-ensemble de noeuds (représentée par la somme des poids des arcs de liaison entre noeuds d'un même sous-ensemble), relativement à sa séparation de l'autre sous-ensemble (la somme des poids des arcs liant les noeuds d'un sous-ensemble aux noeuds de l'autre sous-ensemble).

La coupe normalisée

L'objectif de ce critère de coupe normalisée est identique à celui de la coupe min-max. Cependant, les termes utilisés afin de maximiser les connexions intra-groupes ne sont pas rigoureusement les mêmes. En effet, dans ce critère, la notion de volume d'un sous-ensemble est introduite [Shi and Malik, 2000].

Afin de satisfaire simultanément les deux objectifs qui sont la minimisation de la coupe inter-groupes et la maximisation des volumes de chacun d'entre eux, le critère de coupe normalisée est défini comme suit :

$$J_{NCut}(C_1,C_2) = Cut(C_1,C_2) \times (\frac{1}{vol(C_1)} + \frac{1}{vol(C_2)}). \tag{2.13}$$

De la même manière que pour le critère précédent, cette mesure exprime la cohésion interne d'un sous-ensemble de noeuds, relativement à sa séparation de l'autre sous-ensemble. Nous pouvons remarquer que le critère est d'autant plus minimisé que les volumes de chaque sous-ensemble sont maximisés.

Le tableau 2.3 présente les groupes correspondants aux différents critères de coupe, pour l'exemple illustré sur la figure 2.3. Contrairement aux autres, le critère de coupe min-max ainsi que celui de coupe normalisée permettent d'obtenir la partition souhaitée des objets : $C_1 = \{v_1, v_2, v_3, v_7\}$ et $C_2 = \{x_4, x_5, x_6\}$.

	Groupe C_1	Groupe C_2
Coupe minimale	$\{v_7\}$	$\{v_1, v_2, v_3, v_4, v_5, v_6\}$
Coupe ratio	$\{v_1, v_7\}$	$\{v_2, v_3, v_4, v_5, v_6\}$
Coupe min-max	$\{v_1, v_2, v_3, v_7\}$	$\{v_4, v_5, v_6\}$
Coupe normalisée	$\{v_1, v_2, v_3, v_7\}$	$\{v_4, v_5, v_6\}$

TABLE 2.3 – Groupes obtenus pour les différents critères de coupe à $K = 2$.

Il est possible de normaliser ces critères de coupe en divisant par $\frac{1}{2}$, afin d'obtenir des valeurs comprises dans l'intervalle $[0, 1]$.

2.2.3 K-coupe de graphe ($K > 2$)

Lorsque le nombre de groupes recherchés est supérieur à 2, il est possible d'utiliser deux types d'approches :

- une approche récursive, pour laquelle l'idée principale est d'appliquer récursivement un algorithme de bi-partitionnement d'une manière hiérarchique : après avoir partitionné le graphe en deux sous-ensembles de noeuds, il est nécessaire de réappliquer la même procédure sur ces deux sous-ensembles. Le nombre de groupes recherchés est donc supposé connu ou directement contrôlé par un seuil (fixé par l'utilisateur) associé aux valeurs de la fonction objectif. Cependant, il est aisé de montrer que cette approche présente deux inconvénients majeurs : une obtention de partitions non naturelles et une complexité des calculs (en termes de temps et de mémoire) ;
- une généralisation des fonctions objectifs définies pour $K = 2$, afin de prendre en considération un nombre de groupes K plus élevé. Dans ce cas, il est nécessaire de connaître *a priori* le nombre de ces groupes.

Les groupes recherchés $C_1, \ldots, C_K$ sont définis de la même manière que pour le cas $K = 2$, à savoir :

1. $C_1 \cup \ldots \cup C_K = \mathcal{X}$,
2. $C_k \cap C_{k'} = \emptyset, \forall k \neq k'$.

Soit C_k le k^{eme} groupe de noeuds et $\overline{C_k}$ l'ensemble des noeuds autres que ceux de C_k. La généralisation des critères prédéfinis à $K = 2$ permet alors d'obtenir les fonctions objectifs suivantes :

– la coupe minimale multiple :

$$J_{MMinCut}(C_1, \ldots, C_K) = \sum_{k=1}^{K} Cut(C_k, \bar{C}_k). \tag{2.14}$$

– la coupe ratio multiple :

$$J_{MRatioCut}(C_1, \ldots, C_K) = \sum_{k=1}^{K} \frac{Cut(C_k, \bar{C}_k)}{|C_k|}. \tag{2.15}$$

– la coupe min-max multiple :

$$J_{MMinMaxCut}(C_1, \ldots, C_K) = \sum_{k=1}^{K} \frac{Cut(C_k, \bar{C}_k)}{Cut(C_k, C_k)}. \tag{2.16}$$

– la coupe normalisée multiple :

$$J_{MNCut}(C_1, \ldots, C_K) = \sum_{k=1}^{K} \frac{Cut(C_k, \bar{C}_k)}{vol(C_k)}. \tag{2.17}$$

De la même manière que pour le cas $K = 2$, il est possible de normaliser ces critères de coupe en divisant par $\frac{1}{K}$, afin d'obtenir des valeurs comprises dans l'intervalle $[0, 1]$.

2.3 Algorithmes de classification spectrale

Les algorithmes de classification spectrale minimisent le critère de coupe en résolvant un système de valeurs propres (ou un système de valeurs propres généralisées) grâce à l'extraction du spectre (l'ensemble des valeurs propres) de la matrice Laplacienne. Leurs processus de classification non-supervisée peuvent être résumés en trois étapes :

1. **Pré-traitement** :
   - Construction du graphe de données $G(V,E)$,
   - Construction de la matrice de similarités W associée au graphe $G(V,E)$,
2. **Représentation spectrale** :
   - Construction de la matrice Laplacienne associée au graphe $G(V,E,W)$,
   - Extraction des valeurs et vecteurs propres de la matrice Laplacienne,
   - Projection des objets dans l'espace spectral, basé sur le(s) vecteur(s) propre(s) retenu(s),
3. **Partitionnement** :
   - Recherche de groupes dans l'espace spectral.
   - Affectation des objets aux groupes.

Dans cette partie, quelques algorithmes de classification spectrale usuels sont décrits. La matrice de similarités utilisée dans toutes ces méthodes est la matrice construite à l'aide d'un noyau gaussien comme décrit dans le chapitre 1.

2.3.1 Algorithmes de bi-partition ($K = 2$)

Dans cette partie, nous décrivons deux algorithmes de classification spectrale permettant d'optimiser le critère de Coupe Normalisée du graphe de données. Ces méthodes utilisent un seul et unique vecteur propre afin de partitionner les objets en deux groupes disjoints.

Algorithme de Shi et Malik. Dans [Shi and Malik, 2000], les auteurs définissent le vecteur indicateur du groupe C_1 comme étant $f = (f_1, \ldots, f_N)$ avec $f_i = +1 \Leftrightarrow x_i \in C_1$ et $f_i = -1 \Leftrightarrow x_i \notin C_1$. Grâce à cette définition, mais également à celle du degré d_{ii} du noeud v_i, le critère J_{NCut} peut s'écrire (sachant que $f_i f_j = -1$) :

$$\begin{aligned} J_{NCut}(C_1,C_2) &= Cut(C_1,C_2) \times \left(\frac{1}{vol(C_1)} + \frac{1}{vol(C_2)}\right) \\ &= \frac{Cut(C_1,C_2)}{vol(C_1)} + \frac{Cut(C_1,C_2)}{vol(C_2)} \\ &= \frac{\sum_{i,f_i=+1}\sum_{j,f_j=-1} w_{ij}}{\sum_{i,f_i=+1} d_{ii}} + \frac{\sum_{i,f_i=-1}\sum_{j,f_j=+1} w_{ij}}{\sum_{i,f_i=-1} d_{ii}} \\ &= \frac{\sum_{i,f_i=+1}\sum_{j,f_j=-1} -f_i f_j w_{ij}}{\sum_{i,f_i=+1} d_{ii}} + \frac{\sum_{i,f_i=-1}\sum_{j,f_j=+1} -f_i f_j w_{ij}}{\sum_{i,f_i=-1} d_{ii}}. \end{aligned} \quad (2.18)$$

Grâce à un changement de variable $g = (\mathbf{1}+f) - b(\mathbf{1}-f)$ avec $b = \sum_{i,f_i=+1} d_{ii} / \sum_{i,f_i=-1} d_{ii}$, induisant les deux conditions $g_i \in \{1, -b\}$ et $g^T D\mathbf{1} = 0$, l'équation 2.18 peut être écrite comme un quotient de Rayleigh :

$$\min_g J_{NCut}(C_1, C_2) = \min_g \frac{g^T(D-W)g}{g^T D g}. \tag{2.19}$$

En relaxant les contraintes sur le vecteur indicateur f afin qu'il puisse prendre des valeurs réelles, la minimisation est obtenue en résolvant le système de valeurs propres généralisées :

$$(D-W)g = \lambda D g, \tag{2.20}$$

satisfaisant la contrainte $g^T D\mathbf{1} = 0$. En posant $z = D^{\frac{1}{2}} g$, un système propre standard, plus facile à résoudre, est alors dérivé :

$$\begin{aligned} D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}z &= \lambda z \\ (D^{-\frac{1}{2}}DD^{-\frac{1}{2}} - D^{-\frac{1}{2}}WD^{-\frac{1}{2}})z &= \lambda z \\ (I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}})z &= \lambda z \\ \overline{L}_2 z &= \lambda z. \end{aligned} \tag{2.21}$$

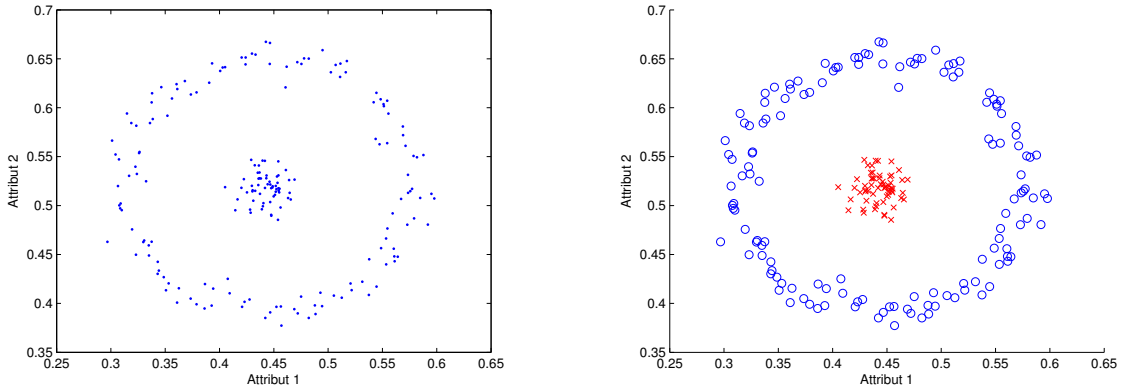
Ainsi, dans la seconde étape, Shi et Malik calculent la matrice Laplacienne normalisée symétrique $\overline{L}_2 = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$. Puis, ils extraient le second plus petit vecteur propre z_2 de $\overline{L}_2$, qui est alors transformé afin d'approximer le vecteur indicateur optimal recherché : $g = D^{-\frac{1}{2}} z_2$. Le premier vecteur propre z_1, colinéaire à $D^{\frac{1}{2}}\mathbf{1}$, est abandonné dans le but de respecter la condition $g^T D\mathbf{1} = 0$.

Pour l'étape finale, les noeuds, et par suite les objets, sont répartis en deux groupes selon le signe des valeurs des composantes du vecteur g (la valeur de la coupe optimale *NCut* est alors recherchée).

Afin d'illustrer les performances de l'algorithme de classification spectrale définie par Shi et Malik, nous choisissons de reprendre l'exemple présenté sur la figure 2.1, qui consiste en deux classes de points formant un rond (61 points) et un anneau (139 points). Le but est de trouver deux groupes de points. Pour cela, nous contruisons la matrice de similarités W en utilisant un noyau gaussien pour lequel le paramètre de dispersion σ est fixé à 10^{-3}.

La figure 2.4 montre les résultats obtenus par l'algorithme 1. Ce dernier est donc capable

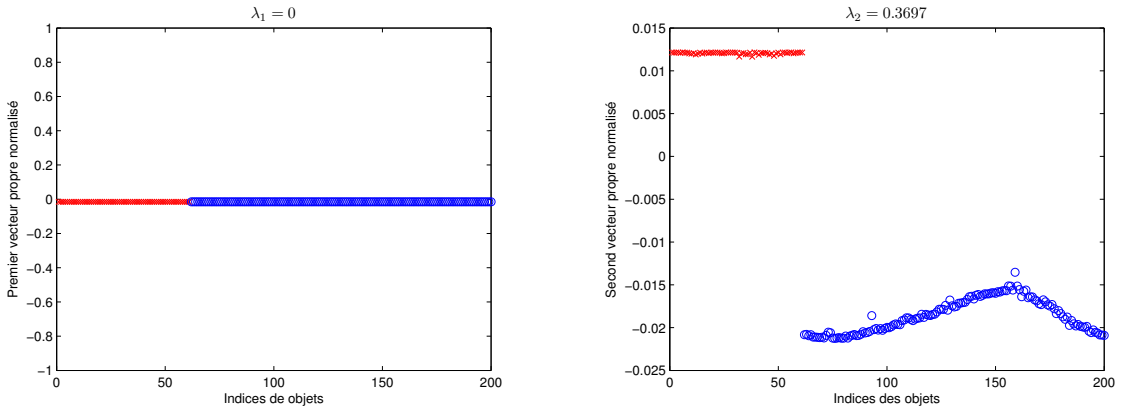
de regrouper les points par voisinage et de séparer les points à l'intérieur du rond des autres points. Cette méthode permet donc d'obtenir le résultat attendu, contrairement à l'algorithme traditionnel des K-moyennes (cf. figure 2.1).



(a) Données originales (b) Groupes obtenus

FIGURE 2.4 – Résultat obtenu en utilisant l'algorithme de Shi et Malik.

La figure 2.5 représente les deux premiers vecteurs propres de la matrice Laplacienne normalisée symétrique $\overline{L}_2$. Comme démontré précédemment, le premier vecteur propre z_1 est abandonné afin de satisfaire la condition $g^T D\mathbf{1} = 0$. La figure 2.5(a) montre, en effet, que ce vecteur, associé à la valeur propre nulle, est constant. Il n'est donc pas possible de partitionner les données selon ce vecteur. En revanche, le second vecteur propre (représenté sur la figure 2.5(b)) permet une meilleure discrimination des objets dans l'espace spectral.



(a) Premier vecteur propre (b) Second vecteur propre

FIGURE 2.5 – Représentation des deux premiers vecteurs propres de la matrice Laplacienne normalisée symétrique, en fonction des indices des objets.

Algorithme 1 Algorithme de Shi et Malik pour $K = 2$

Entrées : matrice de similarités W, nombre de groupes $K = 2$

Pré-traitement

1. Construire le graphe de données $G(V,E,W)$.
2. Construire la matrice de degrés $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$ et $d_{ij} = d_{ji} = 0$ si $i \neq j$.

Représentation spectrale

3. Calculer la matrice Laplacienne normalisée symétrique $\bar{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
4. Extraire le second plus petit vecteur propre z_2 de $\bar{L}_2$.

Partitionnement

5. Séparer les objets en deux groupes selon le signe des valeurs des éléments de $g = D^{-\frac{1}{2}} z_2$.

Sortie : Partition $C = \{C_1, C_2\}$

Algorithme de Von Luxburg. Dans son tutoriel [Luxburg, 2007], l'auteur définit le vecteur indicateur du groupe C_1 comme étant différent de celui de Shi et Malik. Celui-ci est égal à $f = (f_1, \ldots, f_N)$ avec $f_i = a \Leftrightarrow x_i \in C_1$ et $f_i = -a^{-1} \Leftrightarrow x_i \notin C_1$ (avec $a = \sqrt{\frac{Vol(C_2)}{Vol(C_1)}}$). En utilisant la propriété de la matrice Laplacienne normalisée, définie par l'équation 2.5, il est aisé d'écrire le critère J_{NCut} sous la forme d'une fonction quadratique de f :

$$J_{NCut}(C_1,C_2) = \frac{1}{2}\sum_{i,j}(f_i - f_j)^2 W_{ij} = f^T(D-W)f = f^T L f. \tag{2.22}$$

qui peut également être écrit de manière équivalente, sous la forme d'un quotient de Rayleigh (comme défini par Shi et Malik) :

$$\min_f J_{NCut}(C_1,C_2) = \min_f \frac{f^T(D-W)f}{f^T D f}. \tag{2.23}$$

Donc, de la même façon que Shi et Malik, la relaxation des contraintes sur le vecteur indicateur f afin qu'il puisse prendre des valeurs réelles, et le changement de variable $z = D^{\frac{1}{2}} f$, permettent d'obtenir un système propre standard :

$$D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}} z = \lambda z \Leftrightarrow \overline{L}_2 z = \lambda z. \tag{2.24}$$

Le problème à résoudre est alors identique à celui de Shi et Malik, à savoir la minimisation du critère de Coupe Normalisée, défini par :

$$\min_z J_{NCut}(C_1,C_2) = \min_z z^T \overline{L}_2 z, \text{ } \textbf{s.c.} \text{ } z^T z = 1, \tag{2.25}$$

avec exactement la même condition formelle $z^T D\mathbf{1} = 0$.

L'algorithme de Von Luxburg ainsi obtenu est identique à l'algorithme 1 pour le cas $K = 2$.

Le second plus petit vecteur propre z_2 de $\overline{L}_2$ est donc solution du problème de minimisation du critère de *NCut*. Il est possible d'obtenir une solution approchée du vecteur indicateur optimal f en post-multipliant le vecteur z_2 obtenu par $D^{-\frac{1}{2}}$. L'étape de partitionnement est ensuite identique à celle de l'algorithme de Shi et Malik. Les résultats obtenus sur l'exemple présenté précédemment sont donc similaires.

La présentation de ces deux méthodes montre que, bien que les vecteurs indicateurs de groupes soient définis différemment dans les deux algorithmes décrits pour $K = 2$, les solutions permettant d'optimiser le critère de Coupe Normalisée sont identiques dans les deux cas. Donc, malgré la différence des formalismes utilisés pour définir le vecteur optimal f, les auteurs [Shi and Malik, 2000] [Luxburg, 2007] résolvent finalement la même fonction objectif, basée sur la même matrice Laplacienne normalisée $\overline{L}_2$.

2.3.2 Algorithmes de K-partitions ($K > 2$)

Dans le cas où le nombre de groupes est supérieur à 2, il existe, dans la littérature, plusieurs méthodes de résolution du problème de coupe de graphe. Cependant, ces algorithmes peuvent être rangés dans deux catégories selon le nombre de vecteurs propres utilisés de la matrice Laplacienne :

- les méthodes récursives, utilisant récursivement un seul et unique vecteur propre de la matrice Laplacienne normalisée pour partitionner les données ;
- les méthodes directes, utilisant plusieurs vecteurs propres afin de calculer une partition directe des données.

Méthodes de bi-partition récursives

Algorithme de Shi et Malik, "Recursive two-way *NCut*". Shi et Malik [Shi and Malik, 2000] proposent une généralisation directe de leur algorithme dans le cas $K = 2$. En effet, la méthode proposée s'appuie sur une résolution récursive de plusieurs problèmes de bi-partitionnement.

Pour cela, ils extraient le second plus petit vecteur propre de $\overline{L}_2$ puis séparent les objets en deux sous-ensembles C_1 et C_2. A l'itération suivante, ils extraient le second vecteur propre de la matrice Laplacienne normalisée, construite grâce aux similarités calculées entre les objets d'un même groupe (C_1 ou C_2), puis partitionnent à nouveau les objets au sein d'un même groupe. L'algorithme, résumé ci-dessous (cf. algorithme 2), s'arrête alors quand le nombre de groupes souhaité est atteint ou quand la valeur de la coupe est supérieure à un seuil fixé par l'utilisateur.

Algorithme 2 Algorithme de bipartition récursif (Shi et Malik)

Entrées : matrice de similarités W, nombre de groupes $K > 2$

Pré-traitement

1. Construire le graphe de données $G(V,E,W)$.
2. Construire la matrice de degrés $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$ et $d_{ij} = d_{ji} = 0$ si $i \neq j$.

Représentation spectrale

3. Calculer la matrice Laplacienne normalisée symétrique $\bar{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
4. Extraire le second plus petit vecteur propre z_2 de $\bar{L}_2$.

Partitionnement

5. Ranger les élements de $g = D^{-\frac{1}{2}} z_2$ par ordre décroissant.
6. Partitionner le graphe $G(V,E,W)$ en deux sous-ensembles selon les valeurs des éléments de g.

Récursion

7. Répéter les étapes 1 à 7 sur chaque groupe d'objets jusqu'à obtenir K groupes.

Sortie : Partition $C = \{C_1, \ldots, C_K\}$

Algorithme de Perona et Freeman. L'algorithme de Shi et Malik nécessite la construction d'une matrice de similarités d'ordre $\frac{N\times(N-1)}{2}$ (donc de complexité $O(N^2)$). Il arrive fréquemment que le nombre d'objets est très important, rendant le calcul de la matrice de similarités délicat. Dans [Perona and Freeman, 1998], Perona et Freeman proposent un algorithme de complexité $O(N)$. Ce dernier s'appuie sur l'idée simple suivante : au lieu d'affecter un poids aux connexions entre paires de noeuds v_i et v_j (correspondants aux objets x_i et x_j), il s'agit d'affecter un poids p_i à chacun des noeuds v_i tel que $w_{ij} = p_i p_j$.

La détermination des poids p_i et p_j est obtenue à partir de la décomposition de la matrice de similarités. Soit $p = (p_1, \ldots, p_N)^T$ le vecteur poids. Ce vecteur minimise l'équation suivante :

$$\sum_{i=1}^{N} \sum_{j=1}^{N} (w_{ij} - p_i p_j)^2. \tag{2.26}$$

Il s'agit alors d'approximer la matrice de similarités W par une matrice de rang 1, définie par pp^T. La solution optimale de l'équation 2.26 est proportionnelle au premier vecteur propre de la matrice W (noté z_1). Le coefficient n'est autre que la racine carrée de la valeur propre maximale (notée λ_1) :

$$p = {\lambda_1}^{\frac{1}{2}} z_1. \tag{2.27}$$

p prend donc ses valeurs entre 0 et 1. La partition des données en deux sous-ensembles est donc obtenue en affectant les objets pour lesquels la valeur de p est nulle à un groupe C_1, et le reste des objets au groupe C_2. Le partitionnement final en K groupes est alors obtenu de la même manière que pour la méthode de Shi et Malik (bi-partitionnement récursif sur chacun des

groupes obtenus). Cet algorithme trouve ses applications essentiellement dans le domaine de la segmentation d'images.

Algorithme 3 Algorithme de bipartition récursif (Perona et Freeman)

Entrées : matrice de similarités W, nombre de groupes $K > 2$

Pré-traitement

1. Construire le graphe de données $G(V,E,W)$.

Représentation spectrale

2. Extraire le premier plus petit vecteur propre z_1 de W.
3. Calculer $p = \lambda_1^{\frac{1}{2}} z_1$.

Partitionnement

4. Partitionner le graphe $G(V,E,W)$ en deux sous-ensembles selon les valeurs des éléments de p.

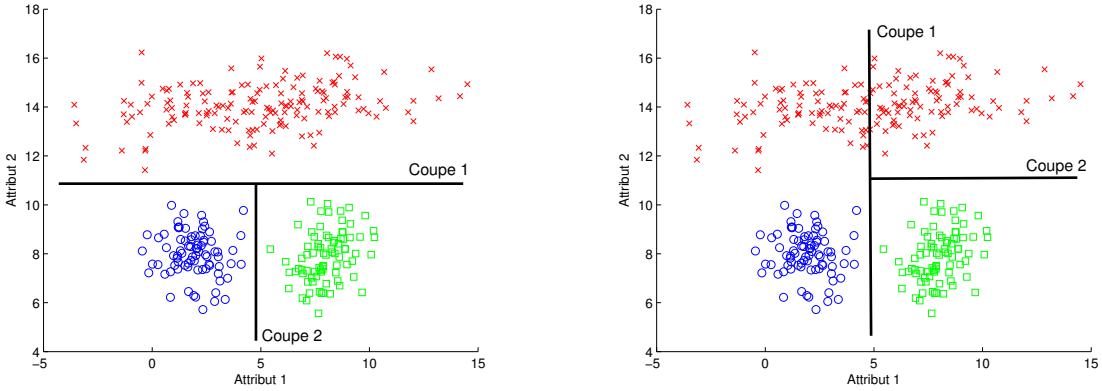
Récursion

5. Répéter les étapes 1 à 4 sur chaque groupe d'objets jusqu'à obtenir K groupes.

Sortie : Partition $C = \{C_1, \ldots, C_K\}$

Les deux méthodes proposées dans cette partie, ont donc recours au bi-partitionnement récursif afin d'obtenir une partition des données en K groupes. Cependant, nous pouvons noter que ce type d'algorithme nécessite un temps de calcul assez long et une allocation d'espace mémoire relativement importante.

La figure 2.6, met en évidence le principal inconvénient de ce type de méthodes, grâce à un exemple composé de 320 objets répartis en trois groupes distincts. La partition finale est très dépendante du premier découpage des données mais également de l'ordre de ces découpages. Le problème soulevé grâce à la figure 2.6(b) permet donc d'affirmer que les méthodes récursives de bipartitionnement peuvent amener à des partitions qui ne sont pas naturelles.



(a) Coupes souhaitées (b) Coupes obtenues

FIGURE 2.6 – Illustration de la limite des méthodes récursives de bipartitionnment.

C'est pourquoi, d'autres méthodes ont été élaborées afin de réduire le temps de calcul et de contrer cet inconvénient, par une résolution plus directe du problème de minimisation du critère de Coupe Multiple.

Méthodes directes de K-partitions

Algorithme de Von Luxburg. L'auteur généralise le critère *NCut* au critère de coupe multiple dit Multiple *NCut* (noté *MNCut*), en utilisant un critère moyen, comme montré dans l'équation 2.17. Elle propose ensuite de résoudre ce problème, en définissant K vecteurs f_k, désignant les vecteurs indicateurs de partitionnement de $\mathcal{X}$ en K groupes [Luxburg, 2007] : $f_k = (f_{1k}, \ldots, f_{Nk})$ avec $f_{ik} = \frac{1}{\sqrt{Vol(C_k)}} \Leftrightarrow x_i \in C_k$ et $f_{ik} = 0 \Leftrightarrow x_i \notin C_k$. Ces vecteurs indicateurs sont stockés en colonne dans la matrice F.

Elle exprime finalement son problème, de la même manière que pour le cas $K = 2$, grâce à la généralisation suivante :

$$\min_Z J_{MNCut}(C_1, \ldots, C_K) = \min_Z \sum_{k=1}^{K} {z_k}^T \overline{L}_2 z_k, \text{ } \textbf{s.c.} \text{ } {z_k}^T z_k = 1, \tag{2.28}$$

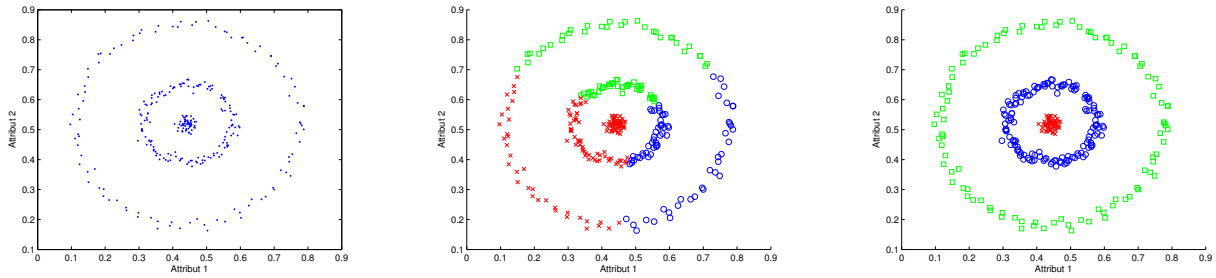
avec une condition formelle supplémentaire $F = D^{-\frac{1}{2}}Z : F^T DF = I$.

Par conséquent, les K premiers vecteurs propres de $\overline{L}_2$ (c'est-à-dire avec les K plus petites valeurs propres) minimisent le critère et permettent d'estimer les K vecteurs indicateurs de groupes. Dans le but de retrouver des valeurs discrètes pour ces vecteurs indicateurs, l'extraction des vecteurs propres est suivie par une étape d'application d'un algorithme de classification non supervisée sur les lignes de $F = D^{-\frac{1}{2}}Z$. En résumé, seules les étapes d'extraction des vecteurs propres et de partitionnement de l'algorithme 1 sont modifiées.

Afin d'illustrer les performances de l'algorithme de classification spectrale pour un nombre de groupes supérieur à 2, comme défini par Von Luxburg, nous choisissons de prendre un exemple comportant trois groupes de points (61 points pour le rond, 139 points pour le petit anneau et 99 points pour le grand anneau) dans un espace à deux dimensions (cf. figure 2.7). La matrice de similarités est alors construite grâce à l'utilisation d'un noyau gaussien, pour lequel σ est fixé à 5×10^{-4}.

La figure 2.7(c) montre les résultats obtenus par l'algorithme de Von Luxburg. Ce dernier est donc capable de regrouper les points les plus similaires entre eux mais également de séparer les

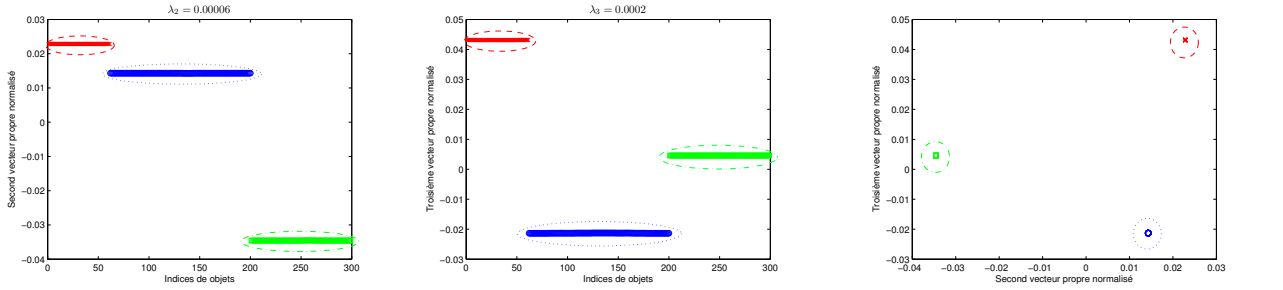
points les plus distants dans l'espace de représentation des données, contrairement à la méthode classique des K-moyennes (représentée sur la figure 2.7(b)).



(a) Données originales (b) Algorithme des K-moyennes (c) Algorithme de Von Luxburg

FIGURE 2.7 – Résultat obtenu en utilisant l'algorithme de Von Luxburg à $K = 3$.

La figure 2.8 représente la projection des points dans l'espace spectral, défini par les trois premiers vecteurs propres de la matrice Laplacienne normalisée symétrique $\overline{L}_2$. Les figures 2.8(a) et 2.8(b) montrent la configuration des différents points projetés sur le second et le troisième vecteur propre respectivement (le premier vecteur propre étant le vecteur constant, il n'apporte pas d'information permettant une discrimination des données). La figure 2.8(c) montre, quant à elle, les points projetés dans l'espace obtenu grâce à la combinaison du second et du troisième vecteurs propres. Sur cette dernière, nous pouvons voir que les points d'un même groupe ont des coordonnées égales dans cet espace de projection. Les méthodes de classification spectrale offrent donc des espaces de projection de faibles dimensions, permettant de discriminer correctement les données.



(a) Second vecteur propre (b) Troisième vecteur propre (c) Combinaison des deux vecteurs

FIGURE 2.8 – Projections spectrales des points sur les différents vecteurs propres de la matrice Laplacienne normalisée symétrique.

Algorithme de Ng et al. Les auteurs [Ng et al., 2002] proposent un algorithme basé sur celui de Weiss [Weiss, 1999], mais aussi sur Meilã et Shi [Meila and Shi, 2000], qui résout également le problème spectral (eq. 2.28), mais sans formuler le problème d'optimisation en termes

Algorithme 4 Algorithme de Von Luxburg pour $K > 2$

Entrées : matrice de similarités W, nombre de groupes $K > 2$

Pré-traitement

1. Construire le graphe de données $G(V,E,W)$.
2. Construire la matrice de degrés $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$ et $d_{ij} = d_{ji} = 0$ si $i \neq j$.

Représentation spectrale

3. Calculer la matrice Laplacienne normalisée $\overline{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
4. Extraire les K plus petits vecteurs propres $\{z_1,\ldots,z_K\}$ de $\overline{L}_2$.
5. Construire la matrice $Z = [z_1,\ldots,z_K] \in \Re^{N\times K}$.

Partitionnement

6. Appliquer un algorithme de partitionnement sur les lignes de la matrice $F = D^{-\frac{1}{2}} Z$.
7. Classer chaque objet de $\mathcal{X}$ en fonction du groupe auquel appartient l'image correspondante dans F.

Sortie : Partition $C = \{C_1,\ldots,C_K\}$

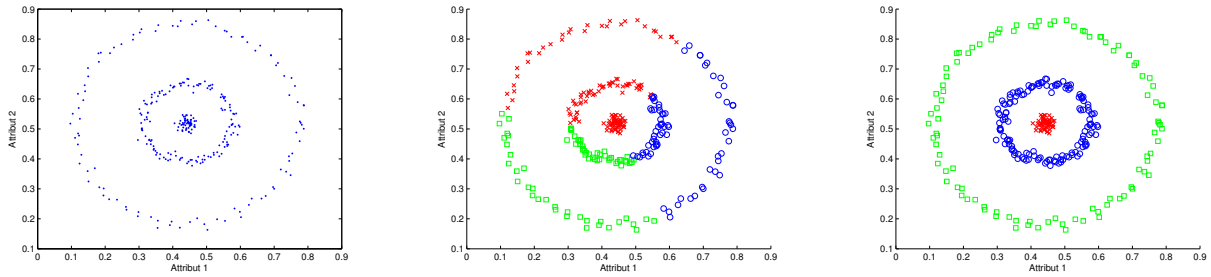
de vecteurs indicateurs.

Ils proposent de modifier la matrice de similarités initiale en imposant $w_{ii} = 0$, et d'utiliser les K plus grands vecteurs propres z_k de $L_{Ng} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, orthogonaux entre eux, afin de projeter les données. Remarquons que ces vecteurs propres sont les K plus petits vecteurs propres de $I - L_{Ng} = \overline{L}_2$ (avec I la matrice identité). Ensuite, à la place de calculer une matrice $F = D^{-\frac{1}{2}} Z$ à partir de la matrice Z stockant les vecteurs propres extraits, ils projettent les points dans l'espace spectral sur la sphère unité, en normalisant Z en F : $F_{ij} = Z_{ij} / \sqrt{\sum_j Z_{ij}^2}$. Enfin, pour partitionner les objets, les auteurs utilisent également l'algorithme des K-moyennes sur les lignes de F.

Nous illustrons maintenant, les performances de cet algorithme, en utilisant l'exemple présenté précédemment pour le cas $K = 3$. La figure 2.9 montre les résultats obtenus par l'algorithme 1. Ce dernier est donc capable de regrouper les points d'un même cercle et de séparer les points appartenant à des cercles différents.

La figure 2.9(c) montre les résultats obtenus par l'algorithme de Ng et al. De la même manière que l'algorithme de Von Luxburg, cette méthode est capable de regrouper les points les plus proches et de séparer les points les plus distants dans l'espace de représentation des données. Ceci montre que ces deux techniques permettent d'obtenir des résultats similaires en terme de partitionnement, mais également en terme de valeurs de coupe normalisée (la valeur de *MNCut* pour les deux méthodes, est égale à 1.05^{-3}).

Cependant, les étapes de normalisation des vecteurs propres étant différentes, nous propo-



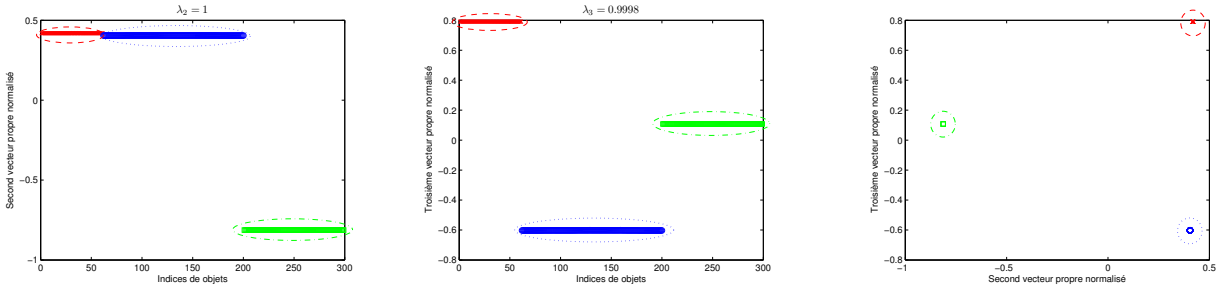
(a) Données originales (b) Algorithme des K-moyennes (c) Algorithme de Ng et al.

FIGURE 2.9 – Résultat obtenu en utilisant l'algorithme de Ng et al. à $K = 3$.

sons d'analyser l'espace de projection obtenu par Ng et al. La figure 2.10 représente la projection des points dans l'espace spectral. Les figures 2.10(a) et 2.10(b) montrent les projections des différents points sur le second puis sur le troisième vecteur propre, et la figure 2.10(c), la projection des points dans l'espace résultant de ces deux vecteurs propres. Cet espace diffère de celui de Von Luxburg par le fait que les points sont projetés sur une sphère de longueur unité. Cela implique que :

– des points appartenant au même groupe sont projetés dans une même direction,
– des points appartenant à des groupes différents sont projetés dans des directions différentes.

De la même façon que pour l'espace de projection obtenu par Von Luxburg, les points représentés par les croix rouges (et ceux représentés par les carrés verts et les cercles bleus) ont des coordonnées égales. C'est pourquoi, il n'est possible de distinguer qu'un seul et unique point pour tous ces objets, sur la figure 2.10(c).



(a) Second vecteur propre (b) Troisième vecteur propre (c) Combinaison des deux vecteurs

FIGURE 2.10 – Projections des points sur la sphère unité, grâce aux différents vecteurs propres de la matrice Laplacienne normalisée symétrique.

Algorithme 5 Algorithme de Ng et al. pour $K > 2$

Entrées : matrice de similarités W, nombre de groupes $K > 2$

Pré-traitement

1. Fixer $w_{ii} = 0$.
2. Construire le graphe de données $G(V,E,W)$.
3. Construire la matrice de degrés $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$ et $d_{ij} = d_{ji} = 0$ si $i \neq j$.

Représentation spectrale

4. Calculer la matrice Laplacienne normalisée $L_{Ng} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
5. Extraire les K plus grands vecteurs propres $\{z_1,\ldots,z_K\}$ de L_{Ng}.
6. Construire la matrice $Z = [z_1,\ldots,z_K] \in \Re^{N\times K}$.
7. Normaliser les lignes de Z afin qu'ils aient une longueur unité (projection sur la sphere unité) :

$$F_{ij} = \frac{Z_{ij}}{\sqrt{\sum_j Z_{ij}^2}}. \tag{2.29}$$

Partitionnement

8. Appliquer un algorithme de partitionnement sur les lignes de la matrice F.
9. Classer chaque objet de $\mathcal{X}$ en fonction du groupe auquel appartient l'image correspondante dans F.

Sortie : Partition $C = \{C_1,\ldots,C_K\}$

2.4 Ajustement des paramètres des algorithmes spectraux

Les algorithmes spectraux présentés dépendent en particulier, des paramètres de la fonction de similarité et du nombre de groupes recherchés. Ils doivent être ajustés à la structure des données.

2.4.1 Réglage du paramètre de dispersion σ de la matrice de similarités

La fonction gaussienne RBF est actuellement la fonction de similarités la plus utilisée dans le domaine de la classification. Le réglage de son paramètre de dispersion σ, permettant de prendre en compte la dispersion locale des données, est souvent négligé dans la description des algorithmes de classification spectrale. Cependant, des valeurs différentes de σ peuvent conduire à des résultats de partitionnement très variés.

Le réglage de ce paramètre σ est la plupart du temps, effectué manuellement. Pourtant, Ng et al. [Ng et al., 2002] suggèrent de sélectionner une valeur automatiquement, en exécutant leur algorithme de classification spectrale pour une gamme donnée de valeurs de σ. La valeur optimale est alors sélectionnée comme étant celle qui fournit des groupes ayant la plus faible distorsion (en terme de stabilité) dans l'espace spectral de représentation. Cependant, il est aisé de montrer que cette technique est très coûteuse en temps, et est fonction du nombre de valeurs

de σ à tester. De plus, la gamme de valeurs est construite manuellement et des données incluant des groupes ayant des dispersions locales différentes engendrent des valeurs de σ différentes.

Zelnik-Manor et Perona [Zelnik-Manor and Perona, 2004] proposent un processus de réglage de σ en calculant, non pas un seul et unique paramètre de dispersion pour l'ensemble des données, mais un paramètre σ_i pour chaque point x_i. La distance du point x_i au point x_j peut alors être vue comme étant égale à $d(i,j)/\sigma_i$. En ce sens, la distance du point x_j au point x_i s'écrit $d(j,i)/\sigma_j$. Cependant, les paramètres σ_i et σ_j sont par nature différents. Il est donc nécessaire de généraliser cette méthode en calculant un paramètre σ_{ij} fonction de σ_i et σ_j. Pour ce faire, Zelnik-Manor et Perona proposent de calculer la distance quadratique entre les deux points :

$$d^2(i,j) = \frac{d(i,j) \times d(j,i)}{\sigma_i \times \sigma_j}. \tag{2.30}$$

La valeur de la similarité entre ces deux points x_i et x_j, en utilisant le noyau gaussien, est définie comme suit :

$$w_{ij} = \exp \frac{-d^2(i,j)}{\sigma_i \times \sigma_j}. \tag{2.31}$$

Maintenant, il est nécessaire de sélectionner une valeur spécifique du paramètre de dispersion pour chaque point. Pour cela, les auteurs étudient les statistiques locales du voisinage du point x_i, et proposent la formule suivante :

$$\sigma_i = d(x_i, x_{\mathcal{K}}) = d(i, \mathcal{K}), \tag{2.32}$$

où $x_{\mathcal{K}}$ représente le $\mathcal{K}^{ieme}$ voisin du point x_i au sens de la distance euclidienne. L'avantage considérable de l'introduction du paramètre $\mathcal{K}$ réside dans le fait que son réglage reste complétement indépendant de la dispersion des données. Les auteurs proposent une valeur empirique pour $\mathcal{K}$ égale à 7.

2.4.2 Estimation du nombre de groupes recherchés

Les différents algorithmes de classification spectrale requièrent, en entrée, le nombre de groupes désirés. Une des plus importantes difficultés de ces méthodes est donc l'estimation de ce nombre. Généralement, ce paramètre est défini manuellement. En effet, à notre connaissance, peu de recherches ont été conduites sur une estimation automatique de ce paramètre.

Analyse des valeurs propres

L'approche la plus intuitive pour estimer le nombre de groupes souhaités est d'analyser les valeurs propres de la matrice Laplacienne. Ng et al. ([Ng et al., 2002], pages 3-4) ont montré que, dans le cas idéal où les objets appartenant à des groupes différents ($K = 3$) sont très distants, il est possible d'obtenir la matrice de similarités définie dans l'équation 2.33 avec $W^{(1)}$ étant la matrice de similarités entre les objets du groupe C_1, $W^{(2)}$ la matrice de similarités entre les objets du groupe C_2 et $W^{(3)}$ la matrice de similarités entre les objets du groupe C_3 [Ng et al., 2002]. La matrice Laplacienne, construite à partir de la division symétrique (par $D^{\frac{1}{2}}$), est alors définie par 2.34 (de même que pour W, $\bar{L}_2^{(1)}$ est la matrice Laplacienne pour les objets du groupe C_1, $\bar{L}_2^{(2)}$ la matrice Laplacienne pour les objets du groupe C_2, etc.).

$$W = \begin{pmatrix} W^{(1)} & 0 & 0 \\ 0 & W^{(2)} & 0 \\ 0 & 0 & W^{(3)} \end{pmatrix} \tag{2.33}$$

$$\bar{L}_2 = \begin{pmatrix} \bar{L}_2^{(1)} & 0 & 0 \\ 0 & \bar{L}_2^{(2)} & 0 \\ 0 & 0 & \bar{L}_2^{(3)} \end{pmatrix} \tag{2.34}$$

Les valeurs propres de $\bar{L}_2$ sont représentées par l'union des valeurs propres des sous-matrices $\bar{L}_2^{(1)}$, $\bar{L}_2^{(2)}$ et $\bar{L}_2^{(3)}$. La plus petite valeur propre de la matrice Laplacienne bloc-diagonale est une valeur propre nulle se répétant avec une multiplicité égale au nombre de groupes désirés. Cela implique qu'il est possible d'estimer facilement ce nombre K, en comptant le nombre de valeurs propres nulles.

Cependant, ce processus d'estimation n'est rendu possible que par le fait qu'il s'agisse d'un cas idéal où les groupes sont clairement et nettement séparés les uns des autres. En effet, si des groupes se chevauchent, les K premières valeurs propres ne sont plus forcément nulles. Le comptage devient donc difficile.

Une alternative est alors proposée par Shortreed et Meila [Shortreed and Meila, 2005] (reprise par Von Luxburg dans [Luxburg, 2007]). Elle consiste à rechercher un "saut" dans les magnitudes des valeurs propres ordonnées, appelé gap. Par conséquent, l'objectif est de choisir le nombre de groupes souhaités tel que les magnitudes des K premières valeurs propres $\lambda_1, \ldots, \lambda_K$ soient très faibles et que celle de λ_{K+1} soit élevée. Le calcul du gap est défini comme suit :

$$\text{gap(i)} = |\lambda_i - \lambda_{i+1}|. \tag{2.35}$$

Le nombre de groupes K est alors choisi en résolvant :

$$K = \arg \max_i \text{ gap(i)}. \tag{2.36}$$

L'inconvénient majeur de cette alternative réside dans le fait que la valeur maximale du gap n'est pas toujours significative et pertinente (à l'exception du cas idéal décrit précédemment). En effet, si la distribution des valeurs propres est quasi uniforme, le choix d'une valeur du nombre de groupes K s'avère alors très difficile. D'autres alternatives ont donc été proposées. Celles-ci consistent, principalement, à analyser les vecteurs propres de la matrice Laplacienne.

Analyse des vecteurs propres

Zelnik-Manor et Perona [Zelnik-Manor and Perona, 2004] proposent de trouver automatiquement le nombre de groupes K en minimisant le coût d'alignement des vecteurs propres selon un système de coordonnées canoniques. En effet, dans le cas idéal décrit précédemment, les K premiers vecteurs propres de la matrice Laplacienne $\bar{L}_2$ sont définis comme étant l'union des premiers vecteurs propres des sous matrices $\bar{L}_2^{(1)}$, $\bar{L}_2^{(2)}$ et $\bar{L}_2^{(3)}$, notés respectivement $z^{(1)}$, $z^{(2)}$ et $z^{(3)}$:

$$Z = \begin{pmatrix} z^{(1)} & 0 & 0 \\ 0 & z^{(2)} & 0 \\ 0 & 0 & z^{(3)} \end{pmatrix} \tag{2.37}$$

Les auteurs supposent alors qu'il existe un autre ensemble de vecteurs orthogonaux définis dans le même espace que les colonnes de Z. Il est donc possible de définir $Z^* = ZR$ pour toute matrice orthogonale $R \in \mathbb{R}^{K \times K}$. Il existe alors une rotation R^* telle que chaque ligne de la matrice Z^*R^* possède un unique élément non nul.

L'étape de sélection du nombre de groupes est donc la suivante : pour chaque valeur possible de K, l'objectif est de retrouver la rotation permettant d'aligner au mieux les colonnes de Z^* selon un système de coordonnées canoniques. Soit $H \in \mathbb{R}^{N \times K}$ la matrice obtenue après rotation de la matrice Z^*, c'est-à-dire $H = Z^*R$. Il est donc désormais possible de retrouver la rotation R pour laquelle, dans chaque ligne de H, il y a au plus un élément non nul. La fonction coût (optimisée grâce à la méthode de descente du gradient) est alors définie comme suit :

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{{H_{ij}}^2}{(\max_i H_{ij})^2}. \tag{2.38}$$

Algorithme 6 Algorithme de Zelnik-Manor et Perona

Entrées : ensemble de données $\mathcal{X}$, nombre maximal de groupes possible K

Pré-traitement

1. Calculer le paramètre local σ_i pour chaque objet $x_i \in \mathcal{X}$.
2. Calculer la matrice de similarités W, avec les éléments w_{ij} définis dans l'équation 2.31.
3. Fixer $w_{ii} = 0$.
4. Construire le graphe de données $G(V,E,W)$.
5. Construire la matrice de degrés $D \in \Re^{N \times N}$: $d_{ii} = \sum_j w_{ij}$ et $d_{ij} = d_{ji} = 0$ si $i \neq j$.

Représentation spectrale

6. Calculer la matrice Laplacienne normalisée $\bar{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
7. Extraire les K plus petits vecteurs propres $\{z_1, \ldots, z_K\}$ de $\bar{L}_2$.
8. Construire la matrice $Z = [z_1, \ldots, z_K] \in \Re^{N \times K}$.
9. Retrouver la rotation R qui permet d'aligner au mieux les colonnes de Z selon un système de coordonnées canoniques en utilisant la méthode de descente du gradient.
10. Calculer le coût d'alignement pour chaque nombre de groupes testé, selon l'équation 2.38
11. Sélectionner le nombre final de groupes K_{fin} comme étant la plus grande valeur de K avec un coût d'alignement minimal.

Partitionnement

12. Prendre le résultat d'alignement H des K_{fin} premiers vecteurs propres.
13. Affecter l'objet original x_i au groupe k si et seulement si $\max_j({Z_{ij}}^2) = {Z_{ik}}^2$.

Sortie : Partition $C = \{C_1, \ldots, C_K\}$

Calcul de la fonction de modularité Ψ

White et Smyth [White and Smyth, 2005] proposent un algorithme spectral permettant d'estimer automatiquement le nombre de groupes recherchés grâce à l'utilisation d'une fonction de modularité Ψ (proposée par Newman et Girvan [Newman and Girvan, 2004] et reprise par Fortunato [Fortunato, 2010]), définie comme suit :

$$\Psi(C^K) = \sum_{k=1}^{K} [\frac{A(V_k,V_k)}{A(V,V)} - (\frac{A(V_k,V)}{A(V,V)})^2], \tag{2.39}$$

avec C^K la partition en K groupes, $A(V_k,V_k) = vol(V) - Cut(V_k,V)$ et $A(V_k,V_{k'}) = \sum_{i \in V_k, j \in V_{k'}} w_{ij}$, donc $A(V_k,V_k)$ représente la somme des poids (sur les arcs de liaison) intra-groupes, $A(V_k,V)$ représente la somme des poids de tous les arcs de liaison attachés au groupe k et $A(V,V)$ représente la somme de tous les poids du graphe.

Grossièrement, la quantité Ψ mesure l'écart entre la probabilité des connexions du graphe dont les noeuds sont dans les mêmes classes (probabilité des connexions intra-classes) et la probabilité que les connexions soient aléatoires entre les noeuds du même graphe. En pratique, les valeurs obtenues sont généralement comprises entre 0.3 et 0.7.

Dans leur papier [Newman and Girvan, 2004], les auteurs montrent qu'une valeur importante de Ψ conduit généralement à des performances élevées de partitionnement. White et Smyth proposent alors de maximiser cette valeur en résolvant un système de valeurs et vecteurs propres. La méthode est résumée dans l'algorithme 7.

Algorithme 7 Algorithme de White et Smyth

Entrées : ensemble de données $\mathcal{X}$, nombre maximal de groupes possible K

Pré-traitement

1. Calculer la matrice de similarités W en utilisant un noyau gaussien.
3. Fixer $w_{ii} = 0$.
4. Construire le graphe de données $G(V,E,W)$.
5. Construire la matrice de degrés $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$ et $d_{ij} = d_{ji} = 0$ si $i \neq j$.

Représentation spectrale

6. Calculer la matrice Laplacienne asymétrique $\bar{L}_1 = D^{-1}W$.
7. Pour chaque valeur de k, $2 \leq k \leq K$:

– Extraire les k plus grands vecteurs propres $\{z_1,\ldots,z_k\}$ de $\bar{L}_1$
– Construire la matrice $Z = [z_1,\ldots,z_k] \in \Re^{N\times k}$.
– Normaliser les lignes de Z afin qu'ils aient une longueur unité (projection sur la sphere unité) :

$$F_{ij} = \frac{Z_{ij}}{\sqrt{\sum_j Z_{ij}^2}}. \tag{2.40}$$

– Appliquer un algorithme de partitionnement sur les lignes de la matrice Z.
– Classer chaque objet de $\mathcal{X}$ en fonction du groupe auquel appartient l'image correspondante dans Z.
– Calculer la valeur de la fonction de modularité Ψ.

8. Sélectionner la valeur finale K_{fin}, maximisant la valeur de Ψ.

Sortie : Partition $C^{K_{fin}} = \{C_1,\ldots,C_{K_{fin}}\}$, valeur de K_{fin}

Dans les algorithmes spectraux optimisant la coupe normalisée multiple *MNCut*, la matrice Laplacienne utilisée est celle normalisée symétrique. Cependant, White et Smyth montrent que la matrice Laplacienne normalisée asymétrique est celle permettant d'optimiser la fonction de modularité Ψ.

Autres techniques d'estimation automatique du nombre de groupes K

D'autres alternatives ont également été développées par :

- **Xiang et Gong** [Xiang and Gong, 2008] : l'hypothèse formulée est que les vecteurs propres de la matrice Laplacienne n'apportent pas tous de l'information nécessaire au partitionnement. Ils proposent alors une méthode afin de mesurer la pertinence d'un vecteur propre selon sa capacité à partitionner les données en groupes différents et disjoints. Pour cela, il est nécessaire de décrire la distribution de chacun des vecteurs en utilisant l'algorithme d'Espérance-Maximisation (noté EM) [Dempster et al., 1977]. Cependant, il est aisé de montrer que cette technique s'appuie sur une analyse non conjointe des vecteurs propres. Ceci amène donc à une sélection individuelle de chacun de ces vecteurs ce qui engendre une complexité en coût et en terme de temps.

- **Sanguinetti et al.** [Sanguinetti et al., 2005] : les auteurs proposent d'utiliser l'algorithme des K-moyennes modifié (”Elongated K-means”) qui diffère de l'algorithme traditionnel par sa façon de calculer les distances entre les objets et les centres. Il est donc basé sur des propriétés géométriques simples des vecteurs propres de la matrice Laplacienne et ne requiert pas la connaissance *a priori* du nombre de groupes K. Ce dernier est déterminé en appliquant cet algorithme sur les deux premiers vecteurs propres. Puis, il s'agit de répéter cette étape en ajoutant itérativement un vecteur propre, tant que des points sont situés autour de l'origine signifiant leur appartenance à un espace plus élevé. La méthode s'arrête alors lorsqu'il y a présence d'un groupe vide. Néanmoins, de la même manière que pour la technique proposée par Xiang et Gong, l'algorithme nécessite une complexité importante en terme de temps.

La technique de détermination automatique du nombre de groupes recherchés la plus adéquate pour un problème de partitionnement réel, semble donc être celle proposée par White et Smyth [White and Smyth, 2005] (c'est-à-dire basée sur le calcul de la fonction de modularité). Pour nos expérimentations, nous choisissons donc d'appliquer cette méthode, mais également celle basée sur le calcul du gap [Shortreed and Meila, 2005] afin de comparer les résultats obtenus.

2.4.3 Choix de l'algorithme pour l'étape de partitionnement

Les deux méthodes directes de classification spectrale présentées précédemment pour $K > 2$ (algorithmes 4 et 5), utilisent un algorithme de classification non supervisée sur la matrice composée des vecteurs propres, afin d'obtenir la partition finale des données. Généralement, cette étape consiste en l'application de la méthode des K-moyennes, qui est une méthode de parti-

tionnement linéaire. Ng et al. [Ng et al., 2002] initialisent les centres des groupes recherchés par les points les plus distants au sens du produit scalaire.

Cependant, d'autres alternatives existent dans la littérature. En effet, certains auteurs utilisent d'autres techniques afin de construire la solution finale à partir des valeurs réelles contenues dans les vecteurs propres :

- utilisation d'hyperplans pour partitionner les données [Lang, 2006],
- analyse du sous-espace constitué par les K premiers vecteurs propres et approximation de ce sous-espace en utilisant des vecteurs constants par morceaux. Cette méthode se ramène à la minimisation d'une distance euclidienne dans l'espace $\mathbb{R}^K$, qui peut être effectuée grâce à l'utilisation d'un algorithme des K-moyennes pondérées [Bach and Jordan, 2004].

2.4.4 Métriques d'évaluation du partitionnement

Dans cette section, nous présentons les deux métriques d'évaluation des partitions obtenues, utilisées dans ce mémoire : le score de F-mesure (grâce au calcul de la précision et du rappel) et l'indice de Rand, auxquels il faut associer la valeur du critère de coupe normalisée.

Matrice de confusion. Une matrice de confusion (appelée également tableau de contingence) permet d'évaluer la qualité d'une partition obtenue par comparaison avec celle réelle complète C_v (appelée "vérité terrain"). Cette matrice est construite en mettant sur les lignes, les classes réelles et sur les colonnes, les groupes obtenus par classification :

		Partition obtenue		
		Groupe 1	...	Groupe K
Partition "terrain"	Classe 1 (C_1)	${a_1}^1$	...	${a_1}^K$
	...	...	...	...
	Classe K (C_K)	${a_K}^1$	...	${a_K}^K$

TABLE 2.4 – Construction de la matrice de confusion.

Ainsi, en définissant la matrice de confusion multi-groupes 2.4, il est possible de calculer plusieurs critères d'évaluation comme le rappel et la précision.

F-mesure. Les mesures de rappel et de précision sont deux critères communément utilisés pour l'évaluation de l'extraction d'informations. Le rappel calcule le nombre d'objets correctement classés parmi tous ceux désirés. La précision, quant à elle, mesure le pourcentage d'objets correctement classés.

Ainsi, en reprenant la matrice de confusion multi-groupes 2.4, il est possible d'écrire les équations suivantes :

$$\text{Rappel}(C_i) = \frac{{a_i}^i}{\sum_{j=1}^{K} {a_j}^i}, \tag{2.41}$$

$$\text{Précision}(C_i) = \frac{{a_i}^i}{\sum_{j=1}^{K} {a_i}^j}, \tag{2.42}$$

$$\text{F-mesure}(C_i) = \frac{2(\text{Précision}(C_i) \times \text{Rappel}(C_i))}{(\text{Précision}(C_i) + \text{Rappel}(C_i))}. \tag{2.43}$$

La F-mesure peut alors être définie comme étant la pondération de la combinaison de la précision et du rappel. Elle est également nommée F-score. Pour un cas multi-groupes, le score final est obtenu en moyennant les F-mesures de chacun des groupes.

Indice de Rand. La qualité des partitions obtenues peut être mesurée grâce à l'indice de Rand. Cette mesure est fréquemment utilisée et reflète la similarité entre la partition "vérité-terrain" C_v et celle obtenue C [Rand, 2007] [Wagstaff, 2002]. Chaque partition peut être vue comme une collection de $N(N-1)/2$ paires d'objets x_i et x_j. Le principe de l'indice de Rand réside alors dans la comparaison des deux partitions par comptage de paires d'objets.

Pour chacune des paires d'objets, il est possible d'avoir les deux cas suivants : soit les partitions affectent les paires d'objets au même groupe, soit les deux objets sont affectés à des groupes différents. En posant :

- a le nombre de fois où les deux partitions (C_v et C) affectent les objets x_i et x_j au même groupe,
- b le nombre de fois où ces deux partitions affectent les objets x_i et x_j à des groupes différents

l'indice de Rand est donné par la formule suivante :

$$RI(C_v, C) = \frac{a+b}{N(N-1)/2}. \tag{2.44}$$

L'indice de Rand est donc fonction du nombre de paires d'objets classées de manière similaire dans les partitions C_v et C.

2.5 Conclusion

Nous avons présenté différents algorithmes de classification spectrale en distinguant deux cas : le cas où le nombre de groupes est égal à 2, et le cas où ce nombre est quelconque. Ces algorithmes s'appuient sur la notion de coupe de graphe. Dans une première partie, nous avons défini quelques notions et termes liés à la théorie spectrale des graphes. En effet, suivant l'hypothèse que tout graphe peut se mettre sous la forme d'une matrice, il est possible de définir une matrice Laplacienne combinant la matrice de similarités et la matrice de degrés des données. Plusieurs définitions ont été fournies selon le type de normalisation appliqué sur cette matrice. Cette dernière est désignée comme étant l'outil principal nécessaire aux algorithmes de classification spectrale récents.

Ensuite, dans une seconde partie, nous avons abordé la notion de coupe de graphe. Plusieurs critères de coupes ont été définis dans la littérature. Nous avons choisi d'en présenter quatre : la coupe minimale, la coupe ratio, la coupe min-max et la coupe normalisée. Nous avons décidé de nous intéresser principalement au critère de coupe normalisée, vu qu'il prend en considération les volumes des groupes, qui correspondent implicitement aux connexions intra-groupes.

Les méthodes de classification spectrale permettant d'optimiser le critère de coupe normalisée et donc de trouver la partition optimale des données, sont relativement nombreuses et variées, en particulier pour le cas multi-groupes. En effet, dans la littérature, la majeure partie des techniques se basent sur l'algorithme de Shi et Malik, développé en 2000, et l'utilise de manière récursive. Récemment, le problème de partitionnement à l'aide d'algorithmes spectraux, a été résolu plus directement, en extrayant non pas un, mais plusieurs vecteurs propres de la matrice Laplacienne.

Cependant, des inconvénients accompagnent les algorithmes de classification spectrale :

- l'utilisation du noyau gaussien pour la construction de la matrice de similarités, nécessite le réglage du paramètre de dispersion σ ;
- le nombre de groupes recherchés K doit être fixé *a priori*.

Pour pallier à ces problèmes, des techniques ont été présentées. La première consiste à calculer un paramètre de dispersion, non pas pour la totalité des données, mais pour un objet donné. Ce calcul, plus local, prend en considération le voisinage de l'objet. La seconde méthode permet d'estimer le nombre de groupes automatiquement grâce à l'analyse des valeurs propres ou au calcul de la fonction de modularité. Si la technique liée aux valeurs propres semble pertinente,

elle n'est pas toujours applicable aux données réelles, alors que la seconde apparaît plus réalisable.

La complexité de la structure des données nous a poussé à nous intéresser à l'intégration de connaissances *a priori* dans ces processus de classification spectrale. Ces connaissances sont des contraintes de comparaison et leur intégration dans les processus de classification, est décrite dans le chapitre suivant.

Chapitre 3

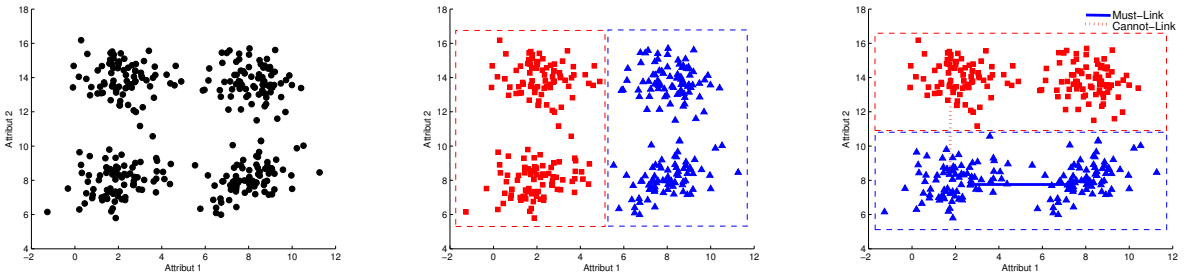
Approches spectrales semi supervisées contraintes

3.1 Introduction

Depuis quelques d'années, l'intérêt porté sur l'intégration de connaissances contextuelles dans la classification s'est accru. En effet, il a été démontré que les performances des algorithmes de classification guidés par l'apport et l'utilisation même d'une faible quantité d'informations, peuvent être améliorées de manière significative [Ge et al., 2007].

L'information *a priori* susceptible d'être intégrée dans les processus de classification, est généralement fournie sous plusieurs formes. Cependant, nous nous intéressons principalement aux contraintes de comparaison par paires, plus simples et faciles à collecter que les étiquettes de classes. Ces contraintes sont de type "Must-Link" si la paire de points doit être dans le même groupe ou "Cannot-Link" si la paire de points ne doit pas être dans le même groupe [Wagstaff, 2002].

La figure 3.1 illustre la conséquence de l'intégration de ce type d'informations sur un exemple composé de quatre sous-groupes d'objets. L'objectif est alors d'obtenir deux groupes distincts ($K = 2$), comme montré sur les figures 3.1(b) et 3.1(c). Les contraintes de comparaison par paires permettent donc de "guider" le partitionnement vers la solution désirée. Ce problème de présence d'une classe dans plusieurs sous-groupes peut avoir lieu dans des applications réelles, et en particulier dans le cas du partitionnement de cellules phytoplanctoniques, où une espèce peut appartenir à plusieurs groupes fonctionnels.



(a) Données originales (b) Algorithme original (c) Algorithme contraint

FIGURE 3.1 – Résultats obtenus en utilisant l'algorithme de classification spectrale original et sous contraintes ($K = 2$).

Les travaux actuels se focalisent sur l'adaptation de méthodes de classification existantes (comme l'algorithme des K-moyennes, les mélanges de densités, la classification hiérarchique) pour la prise en considération des contraintes. Or, leurs inconvénients résident dans le fait qu'elles possèdent les mêmes limites que les techniques dont elles s'inspirent.

Nous présentons plusieurs techniques spectrales incorporant des connaissances *a priori*, issues de la littérature. Ces algorithmes récents, permettant d'améliorer significativement les résultats obtenus, ont des objectifs différents :

- Recherche d'un espace de projection des données permettant de respecter les contraintes définies,
- Partitionnement des images des données dans l'espace de projection.

En effet, il est possible de distinguer différentes méthodes d'intégration et d'utilisation de l'information *a priori*. Dans la littérature, plusieurs auteurs s'attèlent au problème de la recherche d'un sous-espace de projection de plus faible dimension que l'espace original de représentation des données, respectant autant que possible les contraintes de comparaison par paires d'objets. En revanche, d'autres s'intéressent à l'introduction de ces dernières dans des algorithmes non supervisés (existants ou nouveaux), dans le but d'améliorer les résultats de la classification.

Globalement, ces méthodes d'intégration des contraintes peuvent être divisées en deux grandes catégories selon la façon dont elles utilisent la connaissance additionnelle. La première modifie directement les valeurs des similarités inter-objets et applique ensuite un algorithme de partitionnement non supervisé. La seconde, quant à elle, utilise l'information *a priori* afin de restreindre l'espace de solutions possibles grâce à une optimisation sous contraintes. Tous ces algorithmes prennent donc en considération, à la fois les ensembles de contraintes de compa-

raisons ”Must-Link” et ”Cannot-Link”, mais également l'ensemble des données pour lesquels aucune information additionnelle n'est disponible.

Ce chapitre présente un ensemble de méthodes spectrales sous contraintes, soit pour la réduction de la dimension (en particulier, pour la visualisation plane), soit pour la classification multidimensionnelle. Nous présentons ici un état de l'art des approches semi-supervisées avec intégration de contraintes.

3.2 Processus de génération des ensembles de contraintes

L'apport d'information *a priori* dans un algorithme de classification non supervisée permet d'affiner la partition obtenue. Cependant, pour un même algorithme et selon les contraintes choisies, la partition peut différer et ne pas être toujours optimale. Ceci peut s'expliquer notamment par les méthodes de génération des ensembles de contraintes $\mathcal{ML}$ et $\mathcal{CL}$ utilisées.

Nous présentons, dans cette section, un processus de génération aléatoire de paires d'objets sélectionnés comme contraintes. La caractérisation des contraintes s'appuie sur l'évaluation de l'information *a posteriori* apportée par l'algorithme de classification et sur la cohérence de celles-ci.

3.2.1 Génération aléatoire

Dans la littérature, les contraintes de comparaison entre paires d'objets sont très souvent générées à partir des étiquettes de classes connues et à disposition de l'utilisateur [Wang and Davidson, 2010]. Le principe est alors relativement simple : il suffit de sélectionner aléatoirement un certain nombre d'objets. Ensuite, pour une paire d'objets (x_i, x_j) résultant de la sélection :

- une contrainte de type ”Must-Link” est ajoutée à l'ensemble $\mathcal{ML}$ si et seulement si l'étiquette de classe de x_i est identique à celle de x_j,
- une contrainte de type ”Cannot-Link” est ajoutée à l'ensemble $\mathcal{CL}$ si et seulement si l'étiquette de classe de x_i est différente de celle de x_j.

Notons que pour ce type de génération, il existe $\frac{N(N-1)}{2}$ couples d'objets. Il est donc nécessaire de sélectionner un grand nombre de contraintes, afin que les contraintes de comparaison puissent constituer une information utile, et par suite être capables d'améliorer significativement les performances en partitionnement.

L'avantage de la génération de contraintes à partir des étiquettes de classes est qu'elles sont cohérentes. Par contre, toutes les contraintes de comparaison par paires n'apportent pas les mêmes performances en classification [Davidson et al., 2006].

C'est pourquoi, la plupart des auteurs travaillant sur l'intégration de ce type de contraintes dans les algorithmes non supervisés, moyennent leurs résultats de partitionnement sur un nombre important de répétitions de ce processus afin d'obtenir un résultat final qui minimise les erreurs liées à l'aspect aléatoire de la génération.

3.2.2 Caractérisation des contraintes

Dans leur papier [Davidson et al., 2006], Davidson et al. démontrent que la proposition consistant à affirmer que n'importe quel ensemble de contraintes est utile, est fausse. C'est pourquoi, ils introduisent deux mesures afin de quantifier des propriétés importantes des ensembles $\mathcal{ML}$ et $\mathcal{CL}$:

- **le caractère informatif** qui est calculé *a posteriori*. Il mesure la quantité d'information apportée par les ensembles de contraintes, mais que l'algorithme de classification ne peut pas déterminer par lui-même.
- **la cohérence** qui est calculée *a priori*. Elle mesure la quantité de conflits internes dans les ensembles de contraintes $\mathcal{ML}$ et $\mathcal{CL}$.

Le caractère informatif

Soit $\mathcal{E}$ un ensemble regroupant des contraintes "Must-Link" et "Cannot-Link", et $\mathcal{P}$ un algorithme de partitionnement des données. L'approximation de la mesure du caractère informatif des connaissances *a priori* sur les paires d'objets, est basée sur la mesure du nombre des contraintes que l'algorithme de partitionnement ne peut pas prédire lui-même.

Après avoir exécuté l'algorithme $\mathcal{P}$ sur l'ensemble des données sans prise en considération des contraintes, il est possible d'obtenir la partition $C^{(\mathcal{P})}$. Suite à cette partition, certaines contraintes sont satisfaites (ou respectées), d'autres non. L'information apportée par un ensemble de contraintes est évaluée par le nombre de contraintes non satisfaites par l'algorithme de partitionnement $\mathcal{P}$ sur le nombre totale de contraintes $|\mathcal{E}|$:

$$CINF^{(\mathcal{P})}(\mathcal{E}) = \frac{1}{|\mathcal{E}|}[\sum_{c\in\mathcal{E}} \text{unsat}(c, C^{(\mathcal{P})})] \tag{3.1}$$

avec c une contrainte appartenant à $\mathcal{E}$, et $\text{unsat}(c, C^{(\mathcal{P})})$ une fonction indicatrice telle que :

$$\text{unsat}(c, C^{(\mathcal{P})}) = \begin{cases} +1 & \text{si } C^{(\mathcal{P})} \text{ ne satisfait pas la contrainte } c, \\ 0 & \text{sinon.} \end{cases} \tag{3.2}$$

La cohérence

La mesure de cohérence des contraintes est indépendante de l'algorithme de partitionnement utilisé. Elle peut être interprétée géométriquement par l'étude des projections des vecteurs associés aux paires d'objets définissant les contraintes. Nous prenons le cas d'un couple d'objets $\{x_1, x_2\}$ devant appartenir au même groupe (donc appartenant à l'ensemble $\mathcal{ML}$), et d'un couple d'objets $\{x_3, x_4\}$ devant être dans des groupes différents (donc appartenant à l'ensemble $\mathcal{CL}$). Les objectifs sont alors de déterminer si ces contraintes sont cohérentes, mais également leur degré de cohérence.

Les notations utilisées dans cette section sont les suivantes :

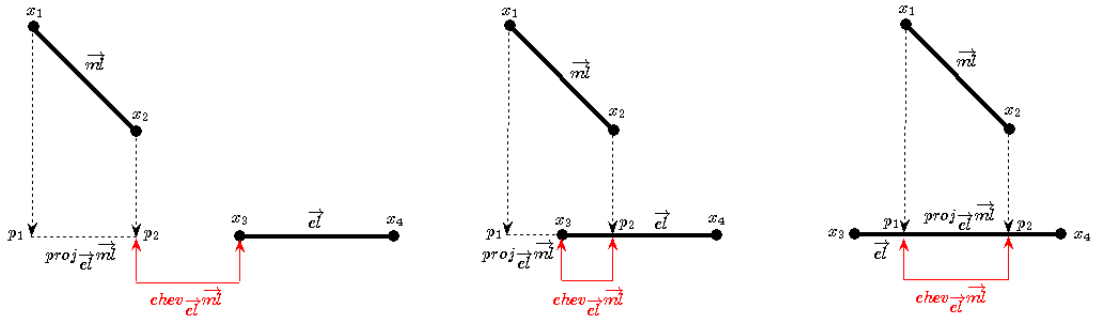
- ml la contrainte "Must-Link" associée au couple $\{x_1, x_2\}$;
- cl la contrainte "Cannot-Link" associée au couple $\{x_3, x_4\}$;
- $\overrightarrow{ml} = \overrightarrow{x_1 x_2}$ le vecteur associé au couple $\{x_1, x_2\}$ dans $\mathcal{ML}$;
- $\overrightarrow{cl} = \overrightarrow{x_3 x_4}$ le vecteur associé au couple $\{x_3, x_4\}$ dans $\mathcal{CL}$;
- p_1 la projection de x_1 sur l'axe $\overrightarrow{x_3 x_4}$;
- p_2 la projection de x_2 sur l'axe $\overrightarrow{x_3 x_4}$;
- $\overrightarrow{p} = \overrightarrow{p_1 p_2}$ le vecteur associé au couple $\{p_1, p_2\}$.

La cohérence de deux contraintes de comparaison peut alors être définie comme suit :

- **Définition 1** : une contrainte ml est dite cohérente avec une contrainte cl si la projection du vecteur $\overrightarrow{ml}$ sur $\overrightarrow{cl}$ possède un chevauchement nul (et inversement).
- **Définition 2** : une contrainte ml est dite partiellement cohérente avec une contrainte cl si la projection du vecteur $\overrightarrow{ml}$ sur $\overrightarrow{cl}$ possède un chevauchement non nul (et inversement).
- **Définition 3** : une contrainte ml est dite incohérente avec une contrainte cl si la projection du vecteur $\overrightarrow{ml}$ sur $\overrightarrow{cl}$ est inclue dans $\overrightarrow{cl}$ (et inversement).

Il est donc nécessaire de calculer le chevauchement résultant de la projection d'une contrainte sur l'autre. Davidson et al. présentent trois exemples illustrant cette méthode [Davidson et al., 2006], comme le montre la figure 3.2.

A partir de ces exemples, il est possible de projeter le vecteur $\overrightarrow{ml}$ sur l'axe défini par $\overrightarrow{cl}$:



(a) Chevauchement nul (b) Chevauchement partiel (c) Chevauchement total

FIGURE 3.2 – Exemples de chevauchement entre contraintes *ml* et *cl*.

$$\overrightarrow{p} = proj_{\overrightarrow{cl}}\overrightarrow{ml} = (||\overrightarrow{ml}||cos\theta)\frac{\overrightarrow{cl}}{||\overrightarrow{cl}||}, \tag{3.3}$$

avec θ représentant l'angle entre les deux vecteurs. Sachant que le vecteur $\overrightarrow{p}$, résultant de la projection de $\overrightarrow{ml}$ sur $\overrightarrow{cl}$, et le vecteur $\overrightarrow{cl}$ sont colinéaires (c'est-à-dire que l'angle est nul), il est possible de calculer la distance du point x_4 avec les points x_3, p_1 et p_2 (points reliés par le vecteur $\overrightarrow{p}$). En reprenant les exemples de la figure 3.2, la mesure de chevauchement est définie comme suit :

$$chev(cl,ml) = \begin{cases} 0 & \text{si } d(x_4,x_3) \leq d(x_4,p_2) \text{ et } d(x_4,x_3) \leq d(x_4,p_1), \\ d(x_3,p_2) & \text{si } d(x_4,p_2) < d(x_4,x_3) \text{ et } d(x_4,p_1) \geq d(x_4,x_3), \\ d(p_1,p_2) & \text{si } d(x_4,p_2) < d(x_4,x_3) \text{ et } d(x_4,p_1) < d(x_4,x_3). \end{cases} \tag{3.4}$$

La cohérence *COH* d'un ensemble $\mathcal{E}$ est alors définie comme le pourcentage de paires de contraintes ayant un chevauchement nul après projection de l'une sur l'axe défini par l'autre :

$$COH(\mathcal{E}) = \frac{\sum_{ml\in\mathcal{ML},cl\in\mathcal{CL}} \delta(\text{chev(cl,ml), chev(ml,cl)})}{|\mathcal{ML}||\mathcal{CL}|} \times 100. \tag{3.5}$$

avec δ une fonction indicatrice telle que :

$$\delta(\text{chev(cl,ml), chev(ml,cl)}) = \begin{cases} 1 & \text{si chev(cl,ml)=0 et chev(ml,cl)=0,} \\ 0 & \text{sinon.} \end{cases} \tag{3.6}$$

Un ensemble de contraintes $\mathcal{E}$ optimal possède donc les propriétés suivantes : un caractère informatif important et une grande cohérence entre les différentes contraintes définies.

Cependant, il est important de noter que la mesure de cohérence ne s'applique pas aux contraintes générées à partir des étiquettes de classes. En effet, ces dernières étant fournies par un expert, il est justifié de supposer que les erreurs d'étiquetage sont quasi-nulles.

Dans nos travaux, nous considérons uniquement les contraintes issues des étiquettes de classes. Ces dernières étant fournies par l'expert, nous supposons que les contraintes dérivées sont, par nature, cohérentes.

3.3 Recherche d'espace de projection sous contraintes

Dans la plupart des applications, il est fréquent de traiter un grand ensemble d'objets décrits par un nombre important d'attributs. Les données, de grande dimension, peuvent alors engendrer une augmentation de la complexité des algorithmes de classification, et même une baisse des performances. C'est pourquoi, il est nécessaire de se tourner vers des techniques de réduction de la dimension. Ces dernières consistent en la projection des objets originaux dans un sous-espace de plus faible dimension, permettant de préserver autant que possible la structure intrinsèque des données.

La matrice de données X, de dimensions $N \times M$, est notée par :

$$X = \begin{pmatrix} x_{11} & \dots & x_{1r} & \dots & x_{1M} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{ir} & \dots & x_{iM} \\ \dots & \dots & \dots & \dots & \dots \\ x_{N1} & \dots & x_{Nr} & \dots & x_{NM} \end{pmatrix} \tag{3.7}$$

La matrice de projection, de dimensions $N \times m$, est notée par :

$$X^* = \begin{pmatrix} x^*_{11} & \dots & x^*_{1m} \\ \dots & \dots & \dots \\ x^*_{i1} & \dots & x^*_{im} \\ \dots & \dots & \dots \\ x^*_{N1} & \dots & x^*_{Nm} \end{pmatrix} \tag{3.8}$$

Les connaissances contextuelles, notamment de contraintes de comparaison, sont utilisées dans le but de guider les méthodes de réduction de la dimension. Le sous-espace optimal est alors défini comme étant celui permettant de préserver la structure originale des données, mais également de faire respecter les contraintes de comparaison par paires, tel que :

– la distance entre les objets projetés et appariés en "Must-Link" est minimale,
– la distance entre les objets projetés et appariés en "Cannot-Link" est maximale.

Dans la suite, nous présentons la réduction de la dimension par l'approche d'analyse en composantes principales contrainte et par l'approche spectrale contrainte.

3.3.1 Analyse contrainte en composantes principales

Dans leur méthode de réduction de la dimension semi-supervisée (notée SSDR), Zhang et al. [Zhang et al., 2007] incorporent un critère prenant en compte les contraintes dans la méthode d'analyse en composantes principales (ACP). L'objectif est de chercher un vecteur de coefficients $z = (z_1, \ldots, z_M)^T$ (avec $x^* = z^T x$ et M le nombre d'attributs), tel que les représentations $x_i{}^*$ des objets x_i (tels que $x_i{}^* = z^T x_i$) dans le sous-espace de projection, puissent préserver la structure originale des données mais également les contraintes "Must-Link" et "Cannot-Link" définies.

Critère d'ACP standard. Le critère d'analyse en composantes principales non contraint est défini comme étant égal à la distance quadratique moyenne entre tous les objets dans l'espace de plus faible dimension obtenu. Il peut donc s'écrire :

$$J_{ACP} = \frac{1}{2}\sum_{i,j}(z^T x_i - z^T x_j)^2 \text{ avec } z^T z = 1. \tag{3.9}$$

Le but final de la méthode d'ACP est de rechercher le vecteur z telle que la dispersion globale des images des objets dans l'espace de projection soit maximale.

Critère d'ACP contraint. Dans [Zhang et al., 2007], les auteurs définissent, à partir du critère d'ACP standard, un critère d'ACP contraint qui tient compte des contraintes appartenant aux ensembles $\mathcal{ML}$ et $\mathcal{CL}$. En effet, l'idée principale de cette méthode est de chercher un vecteur de coefficients z, tel que la projection sur $x^* = z^T x$, des points x vérifie l'équation 3.9 et aussi les contraintes de comparaison.

Dans le cas à une dimension, où un axe principal est recherché (un vecteur z, tel que $||z||_2 = 1$), les auteurs résolvent donc :

$$J_{SSDR} = \underbrace{\frac{1}{2N^2}\sum_{i,j}(z^T x_i - z^T x_j)^2}_{\text{ACP non contraint}} + \underbrace{\frac{\alpha}{|\mathcal{CL}|}\sum_{\{x_i,x_j\}\in\mathcal{CL}}(z^T x_i - z^T x_j)^2}_{\text{Contraintes "Cannot-Link"}} - \underbrace{\frac{\beta}{|\mathcal{ML}|}\sum_{\{x_i,x_j\}\in\mathcal{ML}}(z^T x_i - z^T x_j)^2}_{\text{Contraintes "Must-Link"}}. \tag{3.10}$$

Les termes α et β correspondent aux poids associés au respect des contraintes "Cannot-Link" et "Must-Link" respectivement. Ces paramètres permettent de fixer un degré d'importance des contraintes mais également de contre-balancer les différents termes. Il est donc aisé de remarquer que des valeurs nulles des poids α et β conduisent à la résolution du critère d'ACP standard. Cependant, Zhang et al. proposent des valeurs de manière empirique : $\alpha = 1$ et $\beta \geq 1$. De cette façon, ils privilégient les contraintes de type "Must-Link". Ils montrent, ensuite, que l'équation 3.10 peut être écrite en utilisant une matrice de pondération $Q \in \mathbb{R}^{N\times N}$:

$$\begin{aligned} J_{SSDR} &= \frac{1}{2N^2}\sum_{i,j}(z^T x_i - z^T x_j)^2 + \frac{\alpha}{|\mathcal{CL}|}\sum_{\{x_i,x_j\}\in\mathcal{CL}}(z^T x_i - z^T x_j)^2 - \frac{\beta}{|\mathcal{ML}|}\sum_{\{x_i,x_j\}\in\mathcal{ML}}(z^T x_i - z^T x_j)^2 \\ &= \frac{1}{2}\sum_{i,j}(z^T x_i - z^T x_j)^2 Q_{ij}, \end{aligned} \tag{3.11}$$

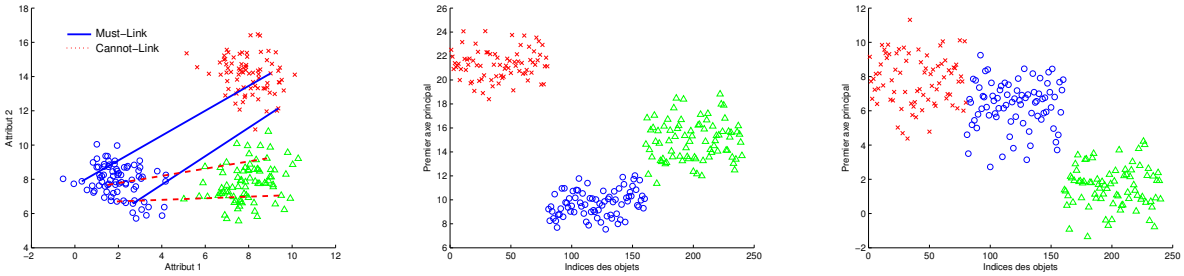
et en utilisant les poids suivants :

$$Q_{ij} = Q_{ji} = \begin{cases} \frac{1}{N^2} + \frac{\alpha}{|\mathcal{CL}|} & \text{si } \{x_i,x_j\} \in \mathcal{CL}, \\ \frac{1}{N^2} - \frac{\beta}{|\mathcal{ML}|} & \text{si } \{x_i,x_j\} \in \mathcal{ML}, \\ \frac{1}{N^2} & \text{sinon.} \end{cases} \tag{3.12}$$

Il est alors possible d'écrire le critère en utilisant des produits matriciels :

$$\begin{aligned} J_{SSDR} &= \frac{1}{2}\sum_{i,j}(z^T x_i - z^T x_j)^2 Q_{ij} = \frac{1}{2}\sum_{i,j}\left(z^T x_i x_i^T z + z^T x_j x_j^T z - 2z^T x_i x_i^T z\right) Q_{ij} \\ &= \frac{1}{2}\sum_{i,j}\left(z^T x_i Q_{ij} x_i^T z + z^T x_j Q_{ij} x_j^T z - 2z^T x_i Q_{ij} x_j^T z\right) \\ &= \frac{1}{2}\sum_{i,j}\left(2z^T x_i Q_{ij} x_i^T z - 2z^T x_i Q_{ij} x_j^T z\right) \\ &= \sum_{i,j}\left(z^T x_i Q_{ij} x_i^T z - z^T x_i Q_{ij} x_j^T z\right) \\ J_{SSDR} &= z^T X R X^T z - z^T X Q X^T z = z^T X (R-Q) X^T z, \end{aligned} \tag{3.13}$$

avec $R \in \mathbb{R}^{N \times N}$ la matrice diagonale telle que $R_{ii} = \sum_j Q_{ij}$. L'équation 3.11 peut alors être résolue en maximisant l'équation 3.13 sous la condition $z^T z = 1$, grâce à l'extraction des vecteurs propres de $X(R-Q)X^T$, correspondants aux plus grandes valeurs propres. La matrice de projection Z est alors composée de ces vecteurs. La méthode d'analyse contrainte en composantes principales est donc linéaire et préserve la structure globale des données tout en respectant les contraintes.



(a) Données originales (2D) (b) ACP classique (1D) (c) ACP contraint (1D)

FIGURE 3.3 – Visualisation du premier axe principal de l'analyse en composantes principales non contrainte et contrainte.

L'exemple sur la figure 3.3(a) montre trois classes et sert à mettre en évidence l'intérêt de la méthode d'analyse contrainte en composantes principales. La figure 3.3(b) présente le résultat obtenu par l'ACP traditionnelle. L'axe des abscisses est l'indice des objets et l'axe des ordonnées est le premier axe principal. La visualisation de ce dernier montre que le groupe d'objets bleus représentés par des cercles, est le plus éloigné de celui composé d'objets rouges (représentés par des croix). Nous proposons de regrouper les deux classes les plus lointaines par ajout de contraintes de comparaison.

L'intégration de connaissances sous la forme de deux contraintes "Must-Link" (entre deux objets du groupe bleu et deux du groupe rouge) et deux contraintes "Cannot-Link" entre deux objets du groupe bleu et deux du groupe vert), a permis d'obtenir un nouvel espace de projection (avec $\alpha = \beta = 1$). La figure 3.3(c) représente les objets projetés sur le premier axe principal, et montre que celui-ci permet de préserver la structure originale des objets tout en respectant les contraintes. En effet, les contraintes tendent à rapprocher les groupes rouge et bleu.

Cependant, dans un problème d'application réelle, il est fréquent que les contraintes de type "Cannot-Link" soit prépondérantes. Zhang et al. proposent de privilégier les contraintes de type "Must-Link" sur celles "Cannot-Link" dans leur méthode d'ACP contrainte. Tang et Zhong [Tang and Zhong, 2007] démontrent, grâce à des résultats obtenus sur plusieurs bases

de données issues de la collection "20-NewsGroups" [1], que l'utilisation seule de contraintes "Must-Link" peut engendrer une diminution des performances de partitionnement dans l'espace de projection obtenu (contrairement aux contraintes "Cannot-Link" seules). Ces deux visions opposées tendent à montrer que le réglage des poids α et β peut s'avérer délicat.

3.3.2 Projection contrainte préservant la structure locale

La projection préservant la structure locale des données (en anglais : Locality Preserving Projection, notée LPP) est une méthode de réduction de la dimension récemment utilisée dans la littérature. Celle-ci cherche la "meilleure" projection des données grâce à la résolution d'un problème préservant de manière optimale, la structure de voisinage des objets [He and Niyogi, 2002]. Elle est donc considérée comme étant une technique alternative à l'analyse en composantes principales. L'objectif est alors de trouver une matrice de projection Z, telle qu'à chaque objet x_i défini dans $\mathbb{R}^M$, correspond un nouvel objet $x_i{}^*$ dans $\mathbb{R}^m$ (avec $m << M$). Il est donc possible d'écrire : $x_i{}^* = Z^T x_i$.

Critère de LPP standard. Le critère lié à la méthode de projection préservant la structure locale des données diffère de celui de l'analyse en composantes principales par la pondération des distances quadratiques par les valeurs de similarités des objets correspondants [He and Niyogi, 2002]. Il s'écrit donc :

$$J_{LPP} = \frac{1}{2}\sum_{i,j}(z^T x_i - z^T x_j)^2 w_{ij}. \tag{3.14}$$

L'objectif de cette méthode est donc d'obtenir un espace de projection de plus faible dimension, et permettant de préserver à la fois, la structure originale des données, mais également la dispersion locale des objets grâce aux valeurs de similarités.

Critère de LPP contraint. La procédure utilisée par la méthode de projection contrainte préservant la dispersion locale (notée CLPP), peut être décrite en quatre étapes [Cevikalp and Verbeek, 2008] [Yu et al., 2010] :

- **Étape 1** : construction du graphe des $\mathcal{K}$ plus proches voisins pondéré $G(V,E,W)$. Le noyau gaussien est utilisé afin de calculer les similarités inter-objets ;
- **Étape 2** : intégration de l'information contenue dans l'ensemble $\mathcal{ML}$. Si deux noeuds v_i et v_j sont appariés en "Must-Link", le poids associé à l'arc de liaison entre v_i et v_j est

1. http ://people.csail.mit.edu/jrennie/20Newsgroups/

fixé à 1. Il est ensuite possible de propager cette information en réglant les valeurs des similarités entre ces noeuds et leurs voisins communs à 1 ;

– **Étape 3** : intégration de l'information contenue dans l'ensemble $\mathcal{CL}$. Si deux noeuds v_i et v_j sont appariés en "Cannot-Link", le poids associé à l'arc de liaison entre v_i et v_j est fixé à -1. La propagation de cette information est réalisée selon la règle de l'héritage (cf. chapitre 1, section 1.3.3) ;

– **Étape 4** : pour la construction de la matrice de projection, le critère d'optimisation est le suivant :

$$\begin{aligned} J_{CLPP} &= \frac{1}{2}(\underbrace{\sum_{i,j}(z^T x_i - z^T x_j)^2 w_{ij}}_{\text{Structure originale}} + \underbrace{\sum_{\{x_i,x_j\}\in\mathcal{ML}}(z^T x_i - z^T x_j)^2}_{\text{Contraintes "Must-Link"}} - \underbrace{\sum_{\{x_i,x_j\}\in\mathcal{CL}}(z^T x_i - z^T x_j)^2}_{\text{Contraintes "Cannot-Link"}}) \\ &= z^T X(D^{(U)} - W^{(U)})X^T z + z^T X(D^{(M)} - W^{(M)})X^T z + z^T X(D^{(C)} - W^{(C)})X^T z \\ &= z^T X(L^{(U)} + L^{(M)} + L^{(C)})X^T z \end{aligned} \tag{3.15}$$

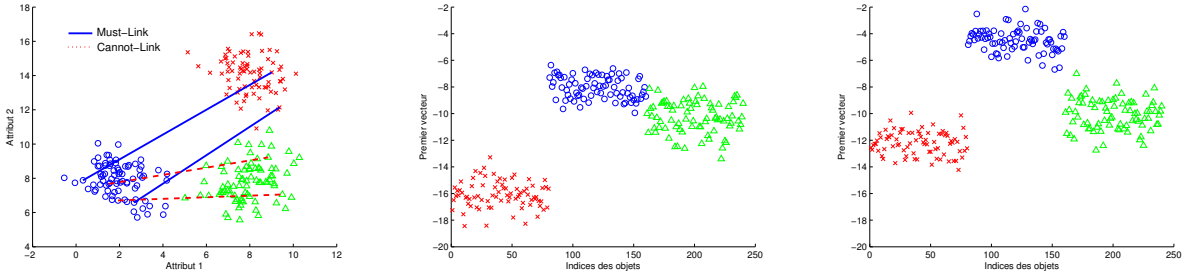
avec ${w_{ij}}^{(U)} = w_{ij}$, ${w_{ij}}^{(M)} = 1$ si $\{x_i,x_j\} \in \mathcal{ML}$, et ${w_{ij}}^{(C)} = -1$ si $\{x_i,x_j\} \in \mathcal{CL}$. $D^{(U)}$, $D^{(M)}$ et $D^{(C)}$ sont les matrices diagonales des degrés correspondantes, définies telles que : ${d_{ii}}^{(U)} = \sum_j {w_{ij}}^{(U)}$, ${d_{ii}}^{(M)} = \sum_j {w_{ij}}^{(M)}$ et ${d_{ii}}^{(C)} = \sum_j {w_{ij}}^{(C)}$. $L^{(U)}$, $L^{(M)}$ et $L^{(C)}$ sont les matrices Laplaciennes non normalisées, définies telles que : $L^{(U)} = D^{(U)} - W^{(U)}$, $L^{(M)} = D^{(M)} - W^{(M)}$ et $L^{(C)} = D^{(C)} - W^{(C)}$.

Ainsi, la matrice Laplacienne est décomposée en somme de trois matrices Laplaciennes qui sont alors désignées par $L^{(U)}$, $L^{(M)}$ et $L^{(C)}$. La matrice de projection finale Z est construite en extrayant les vecteurs propres du système de valeurs propres généralisées suivant :

$$X(L^{(UN)} + L^{(ML)} + L^{(CL)})X^T z = \lambda X(D^{(UN)} + D^{(ML)} + D^{(CL)})X^T z \tag{3.16}$$

sous la condition $z^T X(D^{(UN)} + D^{(ML)} + D^{(CL)})X^T z = 1$, nécessaire afin de normaliser les vecteurs de projection des données.

Cet algorithme de réduction de la dimension sous contraintes possède donc la propriété de préservation de la dispersion locale des données, grâce à l'introduction des poids du graphe. L'espace de projection ainsi obtenu permet de respecter les contraintes définies mais tient également compte de la structure originale des objets (premier terme de J_{LPP}).



(a) Données originales (2D) (b) LPP classique (1D) (c) LPP contraint (1D)

FIGURE 3.4 – Visualisation du premier vecteur propre du système 3.16 pour le LPP traditionnel et le LPP contraint.

L'exemple sur la figure 3.4(a) montre l'intérêt de la méthode de projection contrainte préservant la structure locale. Le résultat présenté sur la figure 3.4(b) est celui obtenu par le LPP traditionnel. La visualisation du premier vecteur propre du système 3.16 montre que le groupe d'objets en bleu (représentés par des cercles) est le plus éloigné du groupe rouge (représenté par des croix). Après intégration de connaissances *a priori* (de la même façon que pour l'ACP contrainte), il est possible d'obtenir un nouvel espace de projection. La figure 3.4(c) représente les objets projetés sur le premier vecteur propre, et montre que celui-ci permet de préserver la structure locale des objets tout en respectant les contraintes. Le groupe vert (représenté par des triangles) et bleu (représenté par des cercles) sont alors plus distants que pour la méthode LPP traditionnelle.

Les deux méthodes de recherche d'espace de projection sous contraintes présentées (ACP contraint et LPP contraint) diffèrent donc principalement par la manière de pondérer les distances quadratiques entre objets :

- pour l'ACP contraint : la pondération s'effectue en construisant une matrice Q pouvant prendre les valeurs $\frac{1}{N^2}$ (pour les objets non contraints), $\frac{\alpha}{|\mathcal{CL}|}$ (pour les objets appariés en "Cannot-Link") et $-\frac{\beta}{|\mathcal{ML}|}$ (pour les objets appariés en "Must-Link") ;
- pour le LPP contraint : la pondération s'effectue à l'aide d'une matrice de similarités W modifiée selon le type de contraintes considérées.

3.4 Classification spectrale contrainte par modification de la matrice de similarités

La classification spectrale contrainte exploite des algorithmes de partitionnement utilisant des mesures de similarités. La matrice W est alors modifiée dans le but de renforcer l'idée de

rapprochement des objets appariés en "Must-Link" et de séparer ceux appariés en "Cannot-Link", selon le schéma fonctionnel présenté sur la figure 3.5 [Li et al., 2009].

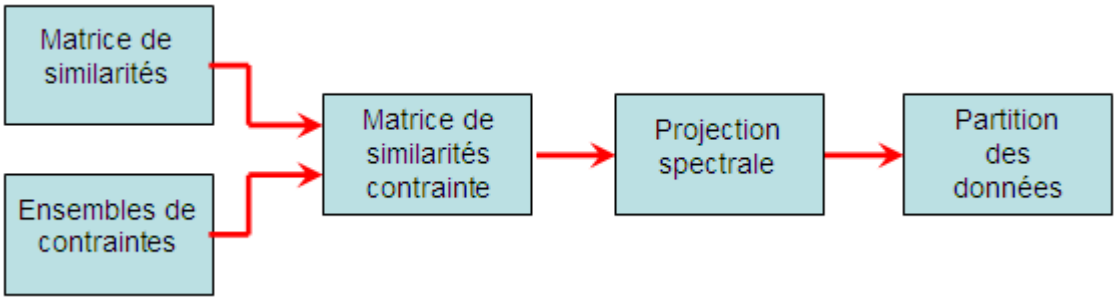


FIGURE 3.5 – Schéma fonctionnel de la classification spectrale contrainte par modification de la matrice de similarités.

Il existe, dans la littérature, différentes façons d'intégrer ces connaissances *a priori* dans la matrice de similarités W. Nous présentons, dans cette section, deux types d'approches :

- une augmentation des dimensions de la matrice de similarités par ajout de noeuds dans le graphe pondéré,
- une modification directe des valeurs de similarités entre les objets liés par une contrainte.

3.4.1 Intégration des contraintes par ajout de noeuds

Soit X_l un ensemble d'objets étiquetés et X_u un ensemble d'objets non étiquetés. Nous supposons que pour chaque groupe k, un objet étiqueté existe (ce qui permet de supposer que le nombre de groupes est fixé et connu). Dans [Shortreed, 2006], Shortreed définit un critère de coupe sous contraintes (notée J_{CCut}), présenté comme une extension de la coupe simple dans un contexte semi-supervisé. Le principe est alors de pénaliser la valeur de cette coupe par le nombre d'objets mal classés.

$$\begin{aligned} J_{CCut}(C_1,\ldots,C_K) &= J_{Cut}(C_1,\ldots,C_K) + \rho \sum_{l=1}^{N_l} I[\text{l mal classés}] \\ &= \sum_{k=1}^{K} Cut(C_k,\bar{C}_k) + \rho \sum_{l=1}^{N_l} I[\text{l mal classés}], \end{aligned} \tag{3.17}$$

avec ρ étant un poids positif associé au terme de pénalisation. Il est alors aisé de montrer que :

- si $\rho = 0$, alors la solution du problème d'optimisation de J_{CCut} est similaire à celle de $JCut$,

– si $\rho \to \infty$ alors seules les partitions offrant des groupes pour lesquels les objets ayant une étiquette de classe identique sont regroupés et ceux ayant des étiquettes de classes différentes sont séparés, sont prises en considération.

Afin de simplifier le calcul du critère J_{CCut}, Shortreed propose de ramener le problème de minimisation de la coupe sous contraintes à un problème de minimisation de la coupe simple. Afin d'atteindre cet objectif, il est nécessaire de définir une matrice de similarités "augmentée" correspondant à un graphe "augmenté", et défini comme suit :

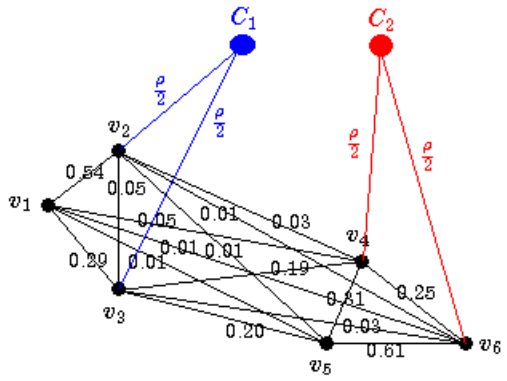


FIGURE 3.6 – Exemple de graphe augmenté $\tilde{G}(V,E,W)$ (pondéré et totalement connecté).

– $\tilde{G}(V,E,W)$ est le graphe $G(V,E,W)$ augmenté : ajout de K noeuds "fictifs" représentant chacun un groupe. Ces noeuds sont connectés uniquement avec les objets ayant une étiquette de classe connue et chaque liaison est pondérée par une valeur égale à $\frac{\rho}{2}$, comme montré sur l'exemple de la figure 3.6 ;
– $\tilde{W}$ est la matrice de similarités augmentée, correspondante au graphe augmenté $\tilde{G}$, et définie comme suit :

$$\tilde{w}_{ij} = \tilde{w}_{ji} = \begin{cases} w_{ij} = w_{ji} & \text{si } i,j \leq N, \\ \frac{\rho}{2} & \text{si } i > N, j \leq N \text{ et } x_j \text{ doit être dans le groupe } C_i, \\ \frac{\rho}{2} & \text{si } j > N, i \leq N \text{ et } x_i \text{ doit être dans le groupe } C_j, \\ 1 & \text{si } i = j > N, \\ 0 & \text{sinon.} \end{cases} \tag{3.18}$$

A la vue de cette matrice de similarités augmentée, il est possible d'affirmer que cette méthode permet de traiter uniquement les contraintes de type "Must-Link". En effet, seuls les similarités des objets devant appartenir à un même groupe prennent des valeurs particulières

dans cette matrice $\tilde{W}$. Les objets non étiquetés et ceux liés par une contrainte "Cannot-Link" n'interviennent pas (leur valeur est fixée à 0).

Grâce à cette matrice de similarités "augmentée", il est possible d'écrire le critère de coupe normalisée sous contraintes de la manière suivante :

$$\begin{aligned} J_{CMNCut}(C_1,\ldots,C_K) &= J_{\tilde{MNCut}}(C_1,\ldots,C_K) \\ &= \sum_{k=1}^{K} \frac{J_{\tilde{Cut}}(C_1,\ldots,C_K)}{\tilde{vol}(C_k)} \\ &= \sum_{k=1}^{K}\sum_{k\neq k'} \frac{\sum_{x_i\in C_k, x_j\in C_{k'}} w_{ij} + \frac{\rho}{2}m_{C_{k'},C_k} + \frac{\rho}{2}m_{C_k,C_{k'}}}{vol(C_k) + \frac{\rho}{2}n_{C_k} + \frac{\rho}{2}(n_{C_k} - m_{C_k,+}) + \frac{\rho}{2}m_{+,C_k}}, \end{aligned} \tag{3.19}$$

avec :

- $J_{\tilde{Cut}}(C_1,\ldots,C_K)$ le critère de coupe minimale du graphe augmenté $\tilde{G}$,
- $\tilde{vol}(C_k)$ le volume du graphe augmenté $\tilde{G}$,
- n_{C_k} le nombre de points ayant une étiquette connue k dans C_k,
- $m_{C_k,C_{k'}}$ le nombre de points devant être dans C_k mais affectés à $C_{k'}$,
- $m_{C_{k'},C_k}$ le nombre de points devant être dans $C_{k'}$ mais affectés à C_k,
- $m_{C_k,+}$ le nombre de points devant être dans C_k mais affectés à un autre groupe :

$$m_{C_k,+} = \sum_{k=1}^{K} m_{C_k,C_{k'}}, \tag{3.20}$$

- m_{+,C_k} le nombre de points devant être dans un autre groupe mais affectés à C_k :

$$m_{+,C_k} = \sum_{k=1}^{K} m_{C_{k'},C_k}. \tag{3.21}$$

Ce critère de coupe normalisée sous contraintes s'appuie implicitement sur deux hypothèses :

- **Hypothèse 1** : les K noeuds supplémentaires sont affectés dans des groupes différents. En effet, la minimisation de la coupe normalisée J_{MNCut} du graphe augmenté $\tilde{G}$ est équivalente à la minimisation du critère de coupe normalisée sous contraintes J_{CMNCut} si, et seulement si, l'on suppose que les noeuds supplémentaires se retrouvent chacun dans un groupe propre (donc qu'ils se retrouvent dans K groupes distincts).

– **Hypothèse 2** : l'application de l'algorithme des K-moyennes sur les N objets est équivalent à l'application de ce même algorithme sur les $N+K$ objets (pour les N premiers objets). Dans un même espace de projection (c'est-à-dire de mêmes dimensions), les partitions obtenues doivent être identiques pour les N premiers objets. En effet, Shortreed pose son problème de minimisation du J_{CMNCut} comme un problème de minimisation de J_{MNCut}. Or, afin d'optimiser ce critère, il est nécessaire d'appliquer l'algorithme de classification spectrale sur la totalité des objets (donc avec les noeuds supplémentaires).

Le problème de J_{MNCut} est résolu grâce à l'algorithme de classification spectrale. Donc, si l'hypothèse 1 est vérifiée (les noeuds supplémentaires sont répartis dans différents groupes), l'application de la méthode de classification spectrale sur la matrice de similarités augmentée permet de se ramener à un problème de J_{CMNCut} et minimise alors le critère présenté dans l'équation 3.19. Néanmoins, si ce n'est pas le cas, l'algorithme minimise la formule générale du critère J_{CMNCut} dans lequel les termes de pénalités ne tendent pas nécessairement à faire respecter les contraintes.

L'alternative proposée par Shortreed pour pallier ce défaut est alors de tenir compte des noeuds supplémentaires dans la définition des projections, mais de les ignorer dans l'application de l'algorithme des K-moyennes (seuls les N objets correspondant aux objets originaux, sont injectés dans cet algorithme).

Sous l'hypothèse 1, et à condition que les noeuds supplémentaires n'influencent pas le partitionnement des autres objets (hypothèse 2), cet algorithme peut alors s'apparenter à celui de classification spectrale. Le critère minimisé est donc bien celui que l'auteur définit mais cela reste un cas particulier. Il est donc théoriquement préférable que l'hypothèse 2 soit vérifiée afin de savoir si le critère J_{CMNCut} est optimisé de manière approximative ou non et donc s'il est possible d'obtenir réellement la partition minimisant ce critère. L'algorithme 8 résume l'approche proposée par Shortreed.

L'inconvénient de cette méthode d'ajout de noeuds "fictifs", réside principalement dans sa complexité et dans le fait qu'il est possible de "casser" la structure originale du graphe (selon le nombre de noeuds ajoutés et le poids liant ces derniers aux objets). En effet, dans un problème où le nombre de classes est important, les dimensions de la matrice de similarités sont fortement augmentées. De plus, cet algorithme ne permet pas d'intégrer des contraintes de comparaison de type "Cannot-Link". Une méthode alternative a été proposée par Kamvar et al. [Kamvar et al., 2003] pour laquelle les valeurs de similarités entre objets appariés par une

Algorithme 8 Algorithme d'ajout de noeuds (Shortreed)

Entrées : matrice de similarités augmentée $\tilde{W}$, nombre de groupes K

Pré-traitement

1. Construire le graphe de données augmenté $\tilde{G}$.
2. Construire la matrice de degrés augmentée $\tilde{D} \in \Re^{(N+K)\times(N+K)}$: $\tilde{d_{ii}} = \sum_j \tilde{w_{ij}}$.

Représentation spectrale

3. Calculer la matrice Laplacienne normalisée asymétrique $\bar{L}_1 = I - \tilde{D^{-1}}\tilde{W}$.
4. Extraire les K plus petits vecteurs propres $\{z_1, \ldots, z_K\}$ de $\bar{L}_1$.
5. Construire la matrice $Z = [z_1, \ldots, z_K] \in \Re^{(N+K)\times K}$.

Partitionnement

6. Appliquer l'algorithme des K-moyennes sur les lignes de la matrice Z en prenant en considération uniquement les N premiers objets.
7. Classer chaque objet de $\mathcal{X}$ en fonction du groupe auquel appartient l'image correspondante dans Z.

Sortie : Partition $C = \{C_1, \ldots, C_K\}$

contrainte, sont directement modifiées.

3.4.2 Modification directe des valeurs de similarités

Dans la méthode proposée par Kamvar et al. [Kamvar et al., 2003] (en anglais : Spectral Learning, notée **SL**), l'algorithme de classification spectrale contraint décrit est construit à partir de la méthode basique de classification spectrale, pour laquelle deux étapes sont modifiées :

– la matrice de similarités W, construite grâce à un noyau gaussien sur un ensemble $\mathcal{X}$ de N objets, est modifiée afin que : pour chaque paire d'objets $\{x_i, x_j\} \in \mathcal{ML}$, les éléments $w_{ij} = w_{ji}$ sont fixés à 1, et pour chaque paire d'objets $\{x_i, x_j\} \in \mathcal{CL}$, les éléments $w_{ij} = w_{ji}$ sont nuls. Cette matrice est alors définie comme suit :

$$w_{ij} = w_{ji} = \begin{cases} 0 & \text{si } \{x_i, x_j\} \in \mathcal{CL}, \\ +1 & \text{si } \{x_i, x_j\} \in \mathcal{ML}, \\ w_{ij} & \text{sinon.} \end{cases} \tag{3.22}$$

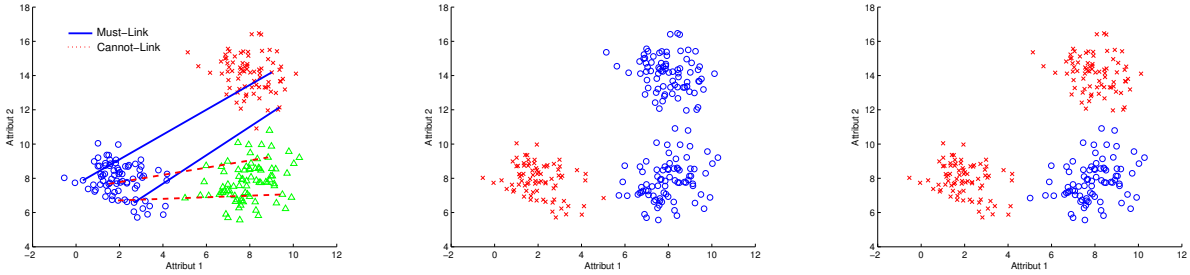
– la normalisation utilisée pour la construction de la matrice Laplacienne est la normalisation additive :

$$\overline{L}_3 = \frac{W + d_{max} I - D}{d_{max}}, \tag{3.23}$$

avec d_{max} la valeur maximale de la diagonale de D ; la matrice obtenue est une matrice de probabilités de transition symétrique ; les auteurs soulignent le fait que les paires d'objets en "Must-Link" ont une valeur de transition mutuelle plus importante que pour les autres paires d'objets ; les vecteurs propres sont alors extraient de cette matrice et les lignes sont

normalisées afin qu'elles aient une longueur unité.

Le partitionnement des données est effectué grâce à l'application de l'algorithme non supervisé des K-moyennes. En reprenant l'exemple de la section 3.3.1 où le nombre de groupes recherchés est égal à 2, il est possible de montrer que l'algorithme présenté est capable de guider le processus de partitionnement grâce à l'introduction de contraintes de comparaison par paires d'objets. En effet, les résultats obtenus par classification spectrale et par la méthode de Kamvar et al. (présentés respectivement sur les figures 3.7(b) et 3.7(c)) montrent des partitions différentes des données. La coupe naturelle, obtenue par la méthode de classification spectrale traditionnelle, permet de regrouper les objets rouges (représentés par des croix) et les objets verts (représentés par des triangles) de la figure 3.7(a). En revanche, l'algorithme proposé par Kamvar et al., scinde les données en regroupant les objets bleus (représentés par des cercles) et les objets rouges (représentés par des croix), comme attendu par les contraintes imposées.



(a) Données originales (2D) (b) Classification spectrale (c) Algorithme de Kamvar et al.

FIGURE 3.7 – Résultats obtenus par classification spectrale et par l'algorithme de Kamvar et al.

La méthode de Kamvar et al. est résumée dans l'algorithme 9.

Méthode alternative dérivée. Comme présenté dans le chapitre 2, Xu et al. montrent l'inconvénient de la normalisation additive [Xu et al., 2005] : les éléments non nuls de la diagonale de cette matrice tendent à générer des groupes ne contenant qu'un seul et unique objet lorsque des outliers sont présents dans les données. C'est pourquoi, nous choisissons de remplacer la matrice Laplacienne normalisée par d_{max}, par la matrice Laplacienne symétrique $\overline{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. Nous appelons cette nouvelle méthode : **SL-**$\overline{L}_2$.

La figure 3.8 présente les résultats obtenus sur deux bases de données extraites des archives UCI : "Ionosphere" (cf. figure 3.8(a)) et "Dermatology" (cf. figure 3.8(b)), en terme d'indice de Rand et en fonction des pourcentages d'étiquettes supposées connues. En effet, pour chaque exemple, quelques pourcentages d'objets sont sélectionnés aléatoirement, de façon à construire

Algorithme 9 Algorithme de modification des similarités (Kamvar et al.)

Entrées : matrice de similarités W, nombre de groupes K

Pré-traitement

1. Pour chaque paire d'objets (x_i, x_j) liée par une contrainte "Must-Link", fixer $w_{ij} = w_{ji} = 1$.
2. Pour chaque paire d'objets (x_i, x_j) liée par une contrainte "Cannot-Link", fixer $w_{ij} = w_{ji} = 0$.
3. Construire le graphe de données $G(V, E, W)$.
4. Construire la matrice de degrés $D \in \Re^{N \times N}$: $d_{ii} = \sum_j w_{ij}$.

Représentation spectrale

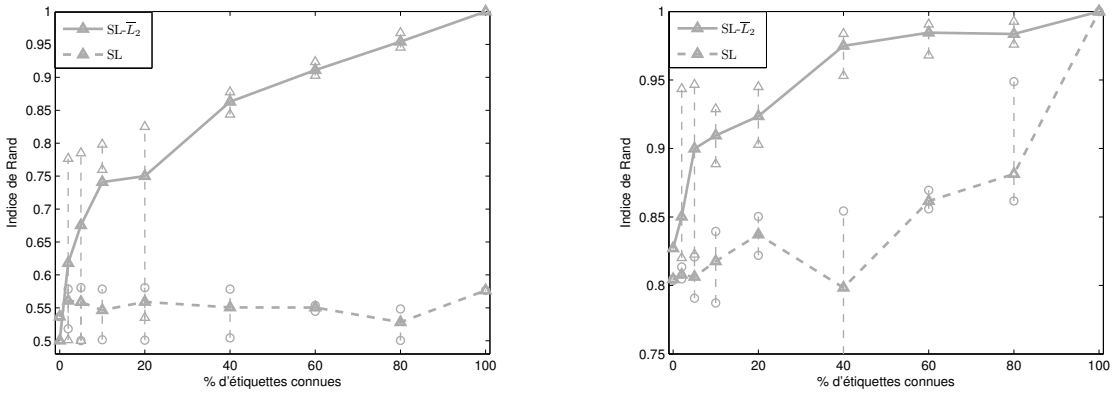
5. Calculer la matrice Laplacienne normalisée $\bar{L}_3 = \frac{(W + d_{max} I - D)}{d_{max}}$.
6. Extraire les K plus grands vecteurs propres $\{z_1, \ldots, z_K\}$ de $\bar{L}_3$.
7. Construire la matrice $Z = [z_1, \ldots, z_K] \in \Re^{N \times K}$.
8. Normaliser les lignes de Z afin qu'ils aient une longueur unité (projection sur la sphere unité) :

$$F_{ij} = \frac{Z_{ij}}{\sqrt{\sum_j Z_{ij}^2}}. \tag{3.24}$$

Partitionnement

9. Appliquer l'algorithme des K-moyennes sur les lignes de la matrice F.
10. Classer chaque objet de $\mathcal{X}$ en fonction du groupe auquel appartient l'image correspondante dans F.

Sortie : Partition $C = \{C_1, \ldots, C_K\}$



(a) Ionosphere ($K = 2$) (b) Dermatology ($K = 6$)

FIGURE 3.8 – Indice de Rand (moyenne, maximum et minimum), en fonction du pourcentage d'étiquettes connues, sur deux bases de données UCI, pour **SL** et **SL-**$\overline{L}_2$.

des ensembles d'étiquettes d'objets. Ensuite, ces derniers sont utilisés afin de déduire les ensembles de contraintes $\mathcal{CL}$ et $\mathcal{ML}$. A chaque pourcentage testé, l'ensemble des objets sélectionnés au test précédent est conservé. Il suffit alors d'ajouter un nouvel ensemble d'objets afin d'atteindre le pourcentage désiré d'étiquettes supposées connues. Les résultats obtenus sont ensuite moyennés sur dix répétitions de ce processus de génération des contraintes.

Pour les deux bases de données considérées, il est aisé de montrer que l'alternative proposée obtient de meilleurs résultats de partitionnement que la méthode originale proposée par Kamvar et al. (en terme d'indice de Rand). En effet, pour n'importe quel pourcentage d'étiquettes connues, les scores de performance sont nettement meilleurs pour la méthode **SL-**$\overline{L}_2$ (par exemple, pour la base "Dermatology" et pour 5% d'étiquettes connues, **SL-**$\overline{L}_2$ obtient un indice de Rand égal à 0.9 contre 0.8 pour **SL**, ce qui représente 19.9% d'objets mieux classés).

3.5 Classification spectrale contrainte par optimisation sous conditions

Les techniques de classification spectrale par optimisation sous contraintes modifient directement les processus de partitionnement des algorithmes non-supervisés. En effet, l'information *a priori* est utilisée afin de guider la recherche de la partition d'origine. Ceci peut être possible en modifiant la fonction objectif (cf. figure 3.9 [Li et al., 2009]) dans le but d'évaluer les groupes obtenus en fonction des contraintes [Demiriz et al., 1999], ou de renforcer l'impact de ces dernières lors du processus de partitionnement [Wagstaff et al., 2001].

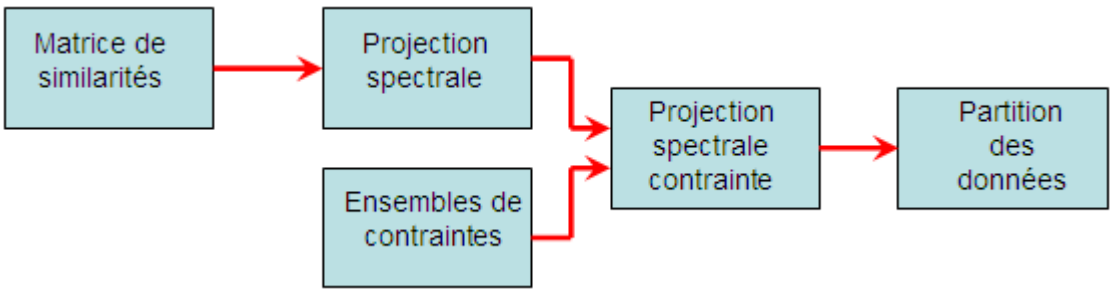


FIGURE 3.9 – Schéma fonctionnel de la classification spectrale contrainte par optimisation sous conditions.

3.5.1 Classification par projection spectrale sous contraintes

L'objectif de l'algorithme original de classification par projection spectrale (en anglais : Laplacian Eigenmap, notée LE) proposé par Belkin et Niyogi [Belkin and Niyogi, 2003], est de réduire la dimension des données afin de pouvoir représenter les objets dans un espace adéquat. Contrairement aux méthodes de projection linéaire (telle que l'ACP), l'algorithme LE offre l'avantage de pouvoir traiter des ensembles de données de structure non linéaire.

Le principe de l'algorithme LE est basé sur la construction d'un graphe des $\mathcal{K}$ plus proches voisins afin de préserver la structure locale des objets. Etant donnée une matrice de similarités W, et par suite une matrice Laplacienne L, il est possible d'extraire les valeurs propres et vecteurs propres du système généralisé suivant :

$$Lf = \lambda Df \tag{3.25}$$

avec D représentant la matrice diagonale des degrés ($d_{ii} = \sum_{j=1}^{N} w_{ij}$) et L étant la matrice Laplacienne non normalisée.

En notant $f_1, \ldots, f_K$ les K plus petits vecteurs propres de l'équation 3.25, les nouvelles coordonnées de x_i sont données par la i^{eme} ligne de $F = [f_1, \ldots, f_K]$. La fonction objectif peut donc s'écrire :

$$J_{LE} = arg\min_{f} \sum_{i,j} ||f_i - f_j||^2 w_{ij} = arg\min_{f} f^T Lf, \text{ } \textbf{s.c.} \; f^T Df = 1, \; f^T D\mathbf{1} = 0. \tag{3.26}$$

A partir de cette équation, Chen et al. [Chen et al., 2009] proposent de dériver une méthode de projection spectrale sous contraintes (notée CLE) en contraignant l'espace de solutions de LE. Pour cela, les auteurs proposent de transformer leur problème multi-groupes en un ensemble de problème à deux groupes (c'est-à-dire avec deux groupes : C_k et $\bar{C}_k$). Puis, ils utilisent deux étapes : une modification de la matrice de similarités puis la recherche d'un sous-espace de projection.

En posant e_i l'ensemble des objets étiquetés appartenant au groupe C_i, il est possible de modifier la matrice W de la manière suivante :

- $w_{ij} = 1$ si $x_i, x_j \in e_k$,
- $w_{ij} = 1$ si $x_i \in e_a$ et $x_j \in e_b$, $a \neq k$ et $b \neq k$,
- $w_{ij} = w_{ji} = 0$ si $x_i \in e_k$ et $x_j \in e_a$, et $a \neq k$.

Après modification, il est aisé de montrer que les liaisons intra-groupes (pour C_k ou $\bar{C}_k$) deviennent plus importantes, et que les liaisons inter-groupes (entre C_k et $\bar{C}_k$) deviennent plus faibles. Il est alors possible de recalculer la matrice diagonale D et la matrice Laplacienne non normalisée L à partir de cette nouvelle matrice de similarités W.

Dans le cas idéal, l'objectif est de projeter les objets appartenant au même groupe au même endroit (autrement dit, en un seul point). Cependant, étant donné que seuls quelques objets

sont étiquetés, les auteurs souhaitent représenter les points appartenant à C_k par un vecteur à une dimension, représenter les autres points étiquetés par un autre vecteur et projeter les points similaires aussi proches que possible les uns des autres. En posant $y = [y_1, \ldots, y_N]^T$ le résultat de la k^{ieme} projection, il est possible d'écrire l'équation suivante :

$$y_{opt} = arg\min_{y} \sum_{i,j} (y_i - y_j)^2 w_{ij} = arg\min_{y} y^T L y, \tag{3.27}$$

avec les contraintes suivantes :
- $y_i = y_j$ si $x_i, x_j \in e_k$,
- $y_i = y_j$ si $x_i \in e_a$ et $x_j \in e_b$, $a \neq k$ et $b \neq k$,
- $y_i \neq y_j$ si $x_i \in e_k$ et $x_j \in e_a$, et $a \neq k$,
- $y^T D y = 1$.

Chen et al. supposent alors que les n_1 premiers objets sont les objets étiquetés appartenant au groupe C_1, les n_2 objets suivants sont ceux étiquetés et appartenant au groupe C_2, et ainsi de suite. Les objets restants ne sont pas étiquetés. Puis, dans le but de reprendre les contraintes définies ci-dessus, les auteurs définissent une matrice de contraintes $Q_k \in \{0, \pm 1\}^{m \times (2+m-p)}$, où $p = n_1 + \ldots + n_K$. Chaque ligne i de Q_k correspond à un objet x_i. La matrice est construite comme suit :

$$Q_k = \begin{pmatrix} \mathbf{1}_{n_1} & -\mathbf{1}_{n_1} & 0_{n_1 \times (N-p)} \\ \cdots & \cdots & \cdots \\ \mathbf{1}_{n_{k-1}} & -\mathbf{1}_{n_{k-1}} & 0_{n_{k-1} \times (N-p)} \\ \mathbf{1}_{n_k} & \mathbf{1}_{n_k} & 0_{n_k \times (N-p)} \\ \mathbf{1}_{n_{k+1}} & -\mathbf{1}_{n_{k+1}} & 0_{n_{k+1} \times (N-p)} \\ \cdots & \cdots & \cdots \\ \mathbf{1}_{n_K} & -\mathbf{1}_{n_K} & 0_{n_K \times (N-p)} \\ \mathbf{1}_{N-p} & 0_{N-p} & I_{N-p \times (N-p)} \end{pmatrix} \tag{3.28}$$

avec $\mathbf{1}_{n_i}$ étant un vecteur unité composé de n_i éléments à 1. En utilisant cette matrice de contraintes Q_k, il est possible de projeter les objets étiquetés en deux points différents, grâce à l'introduction d'un vecteur z tel que $y = Q_k z$. Il est donc possible de réécrire l'équation 3.27 sous la forme :

$$z_{opt} = arg\min_{z} z^T Q_k^T L Q_k z, \ \textbf{s.c.} \ z^T Q_k^T D Q_k z = 1. \tag{3.29}$$

En utilisant le théorème de Rayleigh, la solution du problème est obtenue en extrayant le

deuxième plus petit vecteur propre z_2 du système de valeurs propres généralisées suivant :

$$Q_k^T L Q_k z = \lambda Q_k^T D Q_k z. \tag{3.30}$$

La méthode proposée par Chen et al. est résumée dans l'algorithme 10.

Algorithme 10 Algorithme de projection spectrale sous contraintes (Chen et al.)

Entrées : matrice de similarités W, nombre de groupes K

Pré-traitement

1. Construire le graphe des k plus proches voisins $G(V,E,W)$.
2. Pour deux objets x_i et x_j étiquetés :
   - si x_i et x_j appartiennent à C_k, alors $w_{ij} = 1$,
   - si x_i et x_j n'appartiennent pas à C_k, alors $w_{ij} = 1$,
   - si x_i appartient à C_k et x_j n'appartient pas à C_k, alors $w_{ij} = 0$,
   - si x_i n'appartient pas à C_k et x_j appartient à C_k, alors $w_{ij} = 0$.
3. Construire la matrice de degrés $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$.

Représentation spectrale

4. Calculer la matrice Laplacienne non normalisée $L = D - W$.
5. Construire la matrice de contraintes Q_k pour la k^{ieme} projection obtenue.
6. Extraire le second plus petit vecteur propre z_2 de $Q_k^T L Q_k z = \lambda Q_k^T D Q_k z$.
7. Déduire la solution de la k^{ieme} projection en calculant $y = Q_k z_2$.
8. Répéter les étapes 2 à 7 afin d'obtenir K projections.
9. Combiner les résultats de chaque projection en construisant $Y = [y_1, \ldots, y_K]$.

Partitionnement

10. Appliquer l'algorithme des K-moyennes sur les lignes de la matrice Y.
11. Classer chaque objet de $\mathcal{X}$ en fonction du groupe auquel appartient l'image correspondante dans Y.

Sortie : Partition $C = \{C_1, \ldots, C_K\}$

Cependant, cette technique possède également des limites. En effet, son objectif est de faire respecter le plus de contraintes de comparaison par paires d'objets possibles, alors que ceci n'est pas toujours nécessaire et utile. De plus, dans les applications pratiques, ceci peut s'avérer difficile. C'est pourquoi, Wang et Davidson [Wang and Davidson, 2010] proposent un algorithme de classification spectrale flexible permettant de tolérer un pourcentage de contraintes violées. Ceci est rendu possible en donnant plus d'importance à la partie du critère correspondant à la structure originale des données.

3.5.2 Classification spectrale contrainte et flexible

Dans [Wang and Davidson, 2010], Wang et Davidson expriment leur problème de classification spectrale contrainte, comme un problème d'optimisation sous contraintes qui est résolu

par une étape d'extraction de vecteurs propres. Leur approche est, par conséquent, moins empirique que les méthodes présentées précédemment dans le cadre de la modification des valeurs de similarités. Le terme "flexible" provient du fait que l'algorithme proposé permet de donner une réponse au problème de réglage du poids et de l'impact des contraintes sur la structure originale des données. Ceci est alors réalisé à l'aide d'un seuil fixé sur une mesure reflétant la quantité minimum de contraintes que les auteurs souhaitent respecter. Cette approche cherche donc à rendre l'algorithme flexible quant à la violation de certaines contraintes.

Le problème de classification spectrale semi-supervisée flexible (notée **FCSC**) est décrit dans le cas où $K = 2$. Le vecteur indicateur recherché est alors noté $f \in \{-1,+1\}^N$ (avec N étant le nombre total d'objets), et le respect des contraintes données est mesuré grâce à la matrice Q, définie telle que :

$$Q_{ij} = Q_{ji} = \begin{cases} -1 & \text{si } \{x_i, x_j\} \in \mathcal{CL}, \\ +1 & \text{si } \{x_i, x_j\} \in \mathcal{ML}, \\ 0 & \text{sinon.} \end{cases} \tag{3.31}$$

Grâce à cette matrice Q, il est possible de définir $f^T Qf$ telle que :

$$f^T Qf = \begin{cases} +1 & \text{si } q_{ij} = 1 \text{ et } x_i \text{ et } x_j \text{ ont le même signe dans } f, \\ +1 & \text{si } q_{ij} = -1 \text{ et } x_i \text{ et } x_j \text{ ont des signes différents dans } f, \\ -1 & \text{si } q_{ij} = 1 \text{ et } x_i \text{ et } x_j \text{ ont des signes différents dans } f, \\ -1 & \text{si } q_{ij} = -1 \text{ et } x_i \text{ et } x_j \text{ ont le même signe dans } f. \end{cases} \tag{3.32}$$

Le problème est alors formulé comme un problème d'optimisation sous contraintes, en posant :

$$z = D^{\frac{1}{2}} f \text{ et } \overline{Q} = D^{-\frac{1}{2}} Q D^{-\frac{1}{2}}, \tag{3.33}$$

$$\min_z z^T \overline{L}_2 z \textbf{, s.c. } z^T \overline{Q} z \geq \alpha, z^T z = Vol(G), z \neq D^{\frac{1}{2}} \mathbb{1}. \tag{3.34}$$

La première contrainte permet de définir la borne inférieure du respect des contraintes de comparaison, la seconde permet de normaliser le vecteur indicateur de groupes, et la dernière est utilisée afin d'éviter la solution triviale liée aux algorithmes de classification spectrale (c'est-à-dire, le vecteur constant).

Le problème est finalement résolu en employant le théorème proposé par Kuhn et Tucker [Kuhn and Tucker, 1982] (initialement introduit par Karush [Karush, 1939]) qui utilise des multiplicateurs de Lagrange, mais l'ensemble infini de solutions doit être réduit en contraignant ces multiplicateurs.

Un ensemble possible de vecteurs propres z, est alors solution du problème de valeurs propres généralisées suivant, pour lequel les valeurs propres λ sont strictement positives (condition nécessaire pour le respect des contraintes) :

$$\overline{L}_2 z = \lambda(\overline{Q} - \frac{\theta}{Vol(G)} I)z. \tag{3.35}$$

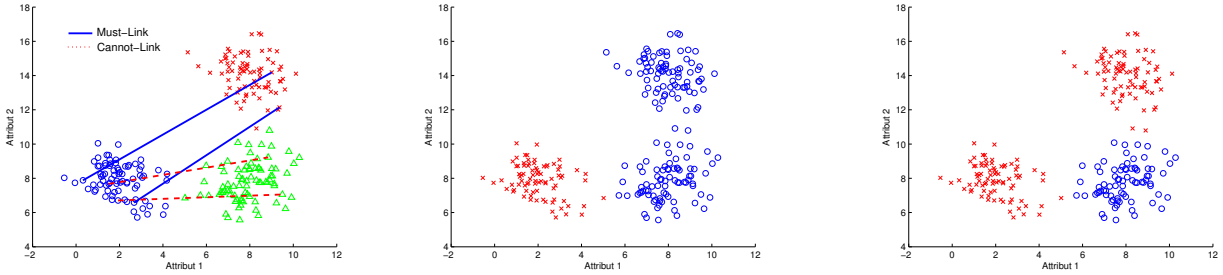
Le vecteur z optimal est alors sélectionné comme étant celui qui minimise la mesure de coupe définie telle que $z^T \overline{L}_2 z$, et étant différent de la solution triviale $D^{\frac{1}{2}}\mathbb{1}$. Le vecteur indicateur solution finale f est enfin obtenu à partir du post-traitement usuel : $f = D^{-\frac{1}{2}} z$, défini dans le chapitre 2.

Le paramètre θ est utilisé afin de pondérer l'impact des contraintes : $\theta < \lambda_{\max} vol(G)$, avec $\lambda_{\max}$ la valeur propre la plus élevée de $\overline{Q}$. Les auteurs proposent la valeur empirique suivante :

$$\theta = \lambda_{\max} \times vol(G) \times \left(0.5 + 0.4 \times \frac{\#\text{ Contraintes}}{N^2}\right).$$

Comme démontré dans leur papier [Wang and Davidson, 2010] dans le cas $K = 2$, cet algorithme obtient de meilleurs résultats de partitionnement que certaines méthodes modifiant directement les valeurs de la matrice de similarités (notamment, l'algorithme de Kamvar et al. [Kamvar et al., 2003]). Cette approche apparaît donc pertinente en terme de performances. Les figures 3.10(b) et 3.10(c) montrent les résultats obtenus sur l'exemple défini dans la section 3.3.1. L'algorithme proposé par Wang et Davidson permet donc de prendre correctement en considération les contraintes "Must-Link" et "Cannot-link" afin d'obtenir le résultat souhaité. En effet, la partition obtenue par cette méthode (présenté sur la figure 3.10(c)) montre que les groupes composés des objets rouges (représentés par des croix) et bleus (représentés par des cercles) sont regroupés. La coupe naturelle, obtenue par l'algorithme de classification spectrale traditionnelle, n'est donc pas respectée.

Dans le cas où $K > 2$, les auteurs généralisent la méthode en sélectionnant non pas le premier, mais les K premiers vecteurs propres généralisés. Cependant, il est aisé de montrer que, contrairement à l'approche de Von Luxburg (2.3.2), les conditions vérifiées par les vec-



(a) Données originales (2D) (b) Classification spectrale (c) Algorithme de Wang et Davidson

FIGURE 3.10 – Résultats obtenus par classification spectrale et par l'algorithme de Kamvar et al.

teurs propres ne sont pas justifiées par le formalisme utilisé ($f_k \in \{-1,1\}^N$) : ni les équations $z_k^T L z_{k'} = 0 \Leftrightarrow k \neq k'$ et $z_k^T(\overline{Q} - \frac{\theta}{vol(G)})z_k \Leftrightarrow k \neq k'$, ni l'équation $f_k^T D\mathbb{1} = 0$.

Algorithme 11 Algorithme de classification spectrale flexible (Wang et Davidson)

Entrées : matrice de similarités W, matrice de contraintes Q, nombre de groupes $K = 2$, paramètre θ

Pré-traitement

1. Construire le graphe de données $G(V,E,W)$.
2. Construire la matrice de degrés $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$.
3. Calculer le volume de $G(V,E,W)$: $vol(G) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}$.

Représentation spectrale

4. Calculer la matrice Laplacienne normalisée symétrique $\bar{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
5. Calculer la matrice de contraintes normalisée $\bar{Q} = D^{-\frac{1}{2}} Q D^{-\frac{1}{2}}$.
6. Extraire la plus grande valeur propre de $\bar{Q}$: λ_{max}.
7. Si $\theta \leq \lambda_{max}$:

– Extraire les vecteurs propres généralisés du système défini dans l'équation 3.35.
– Supprimer les vecteurs propres associés à des valeurs propres non positives.
– Normaliser les vecteurs propres restants par : $z = \frac{z}{||z||} vol(G)^{\frac{1}{2}}$.
– Extraire le vecteur propre solution optimal en résolvant : $\arg\min_z z^T \overline{L}_2 z$.

Partitionnement

8. Séparer les objets en deux groupes selon le signe des valeurs des éléments de z.

Sortie : Partition $C = \{C_1, C_2\}$

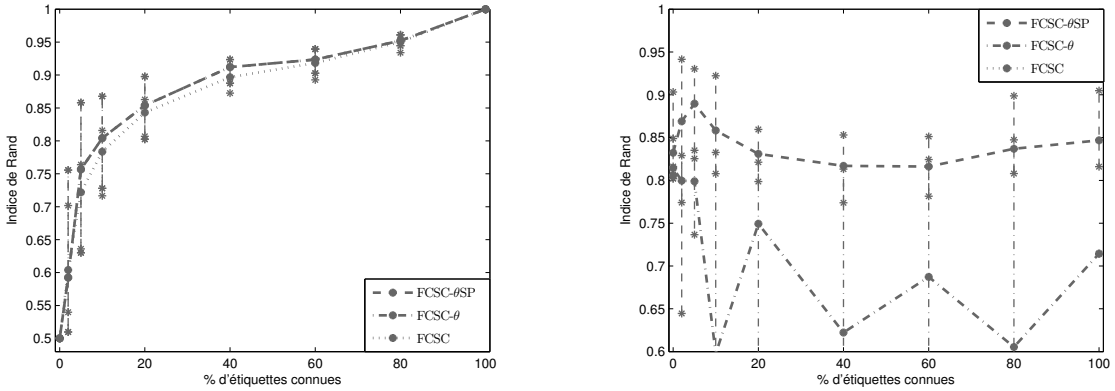
Méthodes alternatives dérivées. Comme démontré dans [Wang and Davidson, 2010], Wang et Davidson proposent un intervalle de valeurs pour le choix du paramètre θ, étant défini comme suit : $[\lambda_{\min} vol(G), \lambda_{\max} vol(G)]$. En effet, les auteurs montrent que cet intervalle est suffisant pour assurer l'existence de K vecteurs propres généralisés respectant les contraintes du problème d'optimisation. En ce sens, nous dérivons une méthode alternative pour laquelle le poids θ est choisi *a posteriori* dans cet intervalle de valeurs, en utilisant une recherche exhaustive. Nous appelons cet algorithme : **FCSC-**θ.

Dans un second temps, nous intégrons l'étape finale de projection sur la sphère unité, comme introduit par Ng et al. [Ng et al., 2002] (nous donnons le nom de **FCSC-θSP** à cette alternative). En effet, comme montré sur la figure 3.11, les scores de performance de partitionnement semblent sensibles à cette étape.

Pour les variantes définies précédemment (**FCSC-θ** et **FCSC-θSP**), le poids du terme de pénalité θ est optimisé *a posteriori*, en discrétisant leur intervalle de définition en 100 valeurs équidistantes, et en choisissant celle qui maximise le critère suivant :

$$E = (1 - J_{MNCut}) + ML_{resp} + CL_{resp}, \tag{3.36}$$

où ML_{resp} et CL_{resp} représentent les taux de contraintes "Must-Link" et "Cannot-Link" respectées. En ce sens, la valeur séélectionnée permet d'obtenir une partition respectant un maximum de contraintes tout en minimisant la valeur de coupe normalisée $MNCut$.



(a) Ionosphere ($K = 2$) (b) Dermatology ($K = 6$)

FIGURE 3.11 – Indice de Rand (moyenne, maximum et minimum), en fonction du pourcentage d'étiquettes connues, sur deux bases de données UCI, pour **FCSC**, **FCSC-θ** et **FCSC-θSP**.

De la même manière que pour l'algorithme **SL** (section 3.4.2), et en suivant le même protocole expérimental, la figure 3.11 présente les résultats obtenus pour les variantes de la méthode **FCSC**, sur les deux mêmes bases de données.

Pour $K = 2$ ("Ionosphere"), l'algorithme **FCSC** obtient de moins bons résultats que les autres variantes. La supériorité de ces dernières peut être expliquée par le fait qu'elles cherchent la valeur optimale de θ (au sens du critère 3.36) dans l'intervalle $[\lambda_{\min} vol(G), \lambda_{\max} vol(G)]$. Nous pouvons également remarquer que les scores de performances ainsi obtenus pour les variantes **FCSC-θ** et **FCSC-θSP** sont identiques.

Pour $K > 2$ (”Dermatology”), il semble important de noter que les résultats obtenus pour l'algorithme original **FCSC** sur la base ”Dermatology” ne sont pas représentés sur la figure 3.11(b). En effet, le problème d'optimisation ne peut pas être résolu pour une valeur de θ donnée ; la règle proposée par Wang et Davidson apparaît comme incompatible pour le cas $K > 2$. De plus, les scores élevés de performances obtenus par la méthode **FCSC-θSP** permettent d'affirmer que l'étape finale de projection sur la sphère unité est primordiale pour le cas multi-groupes. Comme montré sur la figure 3.8(b), cette méthode semble également plus stable que la variante **FCSC-θ**.

3.6 Conclusion

Nous avons présenté un état de l'art des algorithmes utilisant des connaissances contextuelles sous la forme de contraintes de comparaison par paires d'objets. Ces algorithmes peuvent être rangés dans différentes catégories (ou sous-catégories) selon leur objectif :

- la recherche d'espace de projection préservant la structure originale des données et respectant des contraintes de comparaison [Zhang et al., 2007] [Cevikalp and Verbeek, 2008] ;
- la classification des objets par intégration des contraintes dans les algorithmes, selon deux façons :
  - par ajout de noeuds ”fictifs” dans un graphe pondéré [Shortreed, 2006],
  - par modification directe des valeurs de similarités entre les objets liés par des contraintes [Kamvar et al., 2003].

Les ensembles de contraintes $\mathcal{ML}$ et $\mathcal{CL}$ peuvent, quant à eux, être générés de deux manières différentes : soit par un processus totalement aléatoire, soit manuellement de manière intéractive. Il est à noter que les contraintes possédant un caractère informatif élevé, couplé à une cohérence importante, doivent améliorer les performances des classifieurs semi-supervisés.

Nous avons donc distingué les algorithmes de réduction de la dimension sous contraintes et les algorithmes de classification contrainte. La méthode la plus connue en projection linéaire pour la réduction de la dimension est l'analyse en composantes principales. Nous avons donc présenté une version d'ACP contrainte utilisant des poids α et β afin de pondérer l'impact des contraintes ”Cannot-Link” et ”Must-Link” respectivement, sur la structure originale des données. Puis, dans un second temps, nous avons montré l'intérêt et la pertinence des méthodes de projection sous contraintes permettant de préserver la structure locale des données en s'appuyant sur le voisinage de chacun des objets. L'espace de projection obtenu par ces deux mé-

thodes, permet alors de respecter les contraintes définies préalablement, mais également de tenir compte de la structure originale des objets.

Nous avons abordé le problème de classification sous contraintes. Si certains algorithmes introduisent la connaissance contextuelle en modifiant directement les valeurs de similarités entre objets liés par des contraintes, d'autres préférent s'appuyer sur la modification du processus de partitionnement en prenant en considération les contraintes définies.

Dans le cadre de l'application consistant à identifier les cellules phytoplanctoniques présentes dans un échantillon d'eau, ces problèmes peuvent engendrer quelques difficultés : d'une part, il n'est pas obligatoire de faire respecter la totalité des contraintes définies. D'autre part, une même espèce de cellules peut être représentée par plusieurs groupes distincts dans l'espace des attributs : il est donc nécessaire d'avoir recours à des méthodes appliquables au cas multi-groupes. C'est pourquoi, nous proposons un nouvel algorithme de classification semi-supervisée. Cet algorithme, ainsi qu'une analyse comparative avec d'autres algorithmes utilisant les contraintes de comparaison par paires d'objets, sont décrits dans le chapitre suivant.

Chapitre 4

Classification spectrale multi-groupes semi-supervisée

4.1 Introduction

Les algorithmes récents de la littérature, utilisant des connaissances contextuelles, possèdent certaines limites, notamment dans la manière d'intégrer les contraintes de comparaison dans un processus de partitionnement multi-groupes [Wang and Davidson, 2010], mais également dans le choix de la pondération de l'impact de ces contraintes sur la structure des données projetées [Kamvar et al., 2003]. C'est pourquoi, nous proposons un nouvel algorithme capable d'intégrer des contraintes de comparaison dans une classification spectrale pour des problèmes multi-groupes, grâce à l'utilisation d'un terme de pénalité bâti de la même manière que pour l'analyse en composantes principales sous contraintes [Zhang et al., 2007] utilisée pour la réduction de la dimension. La technique proposée permet également de traiter à la fois les contraintes de type "Must-Link" et celles de type "Cannot-Link".

La méthode proposée a pour objectif de minimiser le critère de multi-coupes normalisé (cf. chapitre 2, équation 2.17) tout en tenant compte des contraintes de comparaison par paires [Wacquet et al., 2011c]. Nous accordons la possibilité de donner une valeur plus ou moins importante aux différents types de contraintes selon le degré de confiance accordé (à la manière de Wang et Davidson [Wang and Davidson, 2010]), et montrons l'avantage de notre algorithme par rapport aux algorithmes de la littérature.

Dans un premier temps, notre algorithme est comparé avec différents algorithmes de la littérature sur des exemples illustratifs. Dans un second temps, une analyse comparative est fournie

sur des bases de données provenant des archives UCI [2]. Les résultats obtenus sont finalement présentés en fonction de différents pourcentages d'étiquettes connues, ainsi que quelques indicateurs de performance pour les algorithmes testés.

4.2 Algorithme de classification spectrale semi-supervisée proposé

Dans cette section, nous proposons un algorithme de classification spectrale semi-supervisée (noté **SSSC**). La fonction objectif combine deux critères : celui de classification spectrale classique (*MNCut*) et un critère tenant compte des contraintes.

4.2.1 Pondération de la contribution des contraintes

Dans la littérature, la coupe normalisée peut être exprimée :
- à partir d'un formalisme des vecteurs indicateurs de groupes f_k tels que $f_k \in \{a,b\}^N$ (où $\{a,b\}$ peut prendre les valeurs $\{0,1\}$ ou $\{-1,+1\}$) ;
- à partir des vecteurs propres z de la matrice Laplacienne normalisée $\overline{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.

Notre travail s'appuie sur la seconde alternative. De plus, la plupart des méthodes de la littérature opèrent un post-traitement sur les vecteurs propres, soit par une multiplication par $D^{-\frac{1}{2}}$, soit par une étape de projection sur la sphère unité (celle choisie pour nos travaux).

La projection finale sur la sphère unité rend le critère de contribution des contraintes dépendant des angles entre les projections spectrales, définies par les K premiers vecteurs propres de la matrice Laplacienne $\overline{L}_2$ (K étant le nombre de groupes désirés). Le critère de contribution des contraintes J_{PC} est alors défini en utilisant des produits scalaires entre objets contraints. En effet, nous considérons que ce produit traduit correctement l'altération des angles :

$$\begin{aligned} J_{PC} &= -\frac{1}{|\mathcal{CL}|} \sum_{\{x_i,x_j\}\in\mathcal{CL}} \sum_{k=1}^{K} z_{ik}.z_{jk} + \frac{1}{|\mathcal{ML}|} \sum_{\{x_i,x_j\}\in\mathcal{ML}} \sum_{k=1}^{K} z_{ik}.z_{jk} \\ &= \sum_{k=1}^{K} \left[-\frac{1}{|\mathcal{CL}|} \sum_{\{x_i,x_j\}\in\mathcal{CL}} z_{ik}.z_{jk} + \frac{1}{|\mathcal{ML}|} \sum_{\{x_i,x_j\}\in\mathcal{ML}} z_{ik}.z_{jk} \right]. \end{aligned} \tag{4.1}$$

2. http ://archive.ics.uci.edu/ml/

Ce critère est donc fonction des ensembles de contraintes $\mathcal{ML}$ et $\mathcal{CL}$. Nous exprimons ensuite ce critère de contribution des contraintes J_{PC} comme un produit matriciel, en utilisant une matrice de pondération Q similaire à celle de Wang et Davidson [Wang and Davidson, 2010]. Cette dernière est définie comme suit :

$$Q_{ij} = Q_{ji} = \begin{cases} -\frac{1}{|\mathcal{CL}|} & \text{si } \{x_i, x_j\} \in \mathcal{CL}, \\ +\frac{1}{|\mathcal{ML}|} & \text{si } \{x_i, x_j\} \in \mathcal{ML}, \\ 0 & \text{sinon.} \end{cases} \tag{4.2}$$

Le critère d'optimisation des contributions des contraintes J_{PC} peut alors être écrit de la manière suivante :

$$J_{PC} = \frac{1}{2} \sum_{i,j} \sum_{k=1}^{K} z_{ik} z_{jk} Q_{ij} = \sum_{k=1}^{K} z_k^T Q z_k. \tag{4.3}$$

4.2.2 Critère de coupes multiples normalisé contraint

Le critère de contribution des contraintes J_{PC}, est combiné au critère de coupes multiples normalisé J_{MNCut} (cf. chapitre 2, équation 2.17) afin de définir un problème d'optimisation spectrale utilisant des contraintes de comparaison par paires. La fonction objectif globale est alors définie comme étant égale à :

$$J = J_{MNCut} - J_{PC}. \tag{4.4}$$

La minimisation de cette fonction objectif permet de caractériser une projection spectrale qui reflète la structure originale des données et les contraintes proposées. Nous pouvons maintenant écrire aisément les critères J_{MNCut} et J_{PC} comme des quotients de Rayleigh, dans le but de poser notre problème comme un problème d'extraction de valeurs propres et de vecteurs propres :

$$J_{MNCut} = \sum_{k=1}^{K} \frac{{z_k}^T \overline{L}_2 z_k}{{z_k}^T z_k} \tag{4.5}$$

$$J_{PC} = \sum_{k=1}^{K} \frac{z_k^T Q z_k}{{z_k}^T z_k}, \tag{4.6}$$

avec $\overline{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ la matrice Laplacienne normalisée symétrique. Les termes J_{MNCut} et J_{PC} sont alors introduits dans l'équation 4.4 :

$$\begin{aligned} J &= \sum_{k=1}^{K} \frac{{z_k}^T \overline{L}_2 z_k - z_k^T Q z_k}{{z_k}^T z_k} \\ &= \sum_{k=1}^{K} \frac{z_k^T (\overline{L}_2 - Q) z_k}{{z_k}^T z_k}. \end{aligned} \tag{4.7}$$

Le problème d'optimisation sous contraintes peut alors être écrit comme suit :

$$\min_{Z} \sum_{k=1}^{K} z_k^T (\overline{L}_2 - Q) z_k, \text{ } \textbf{s.c.} \ {z_k}^T z_k = 1. \tag{4.8}$$

L'optimisation de la fonction objectif globale J se résume donc à l'extraction des vecteurs propres de la matrice $(\overline{L}_2 - Q)$. Ce problème est clairement similaire à celui de la classification spectrale traditionnelle défini par l'équation 2.28, à l'exception de la matrice Laplacienne normalisée $\overline{L}_2$ qui est pénalisée par la matrice Q bâtie à partir des ensembles de contraintes $\mathcal{ML}$ et $\mathcal{CL}$.

4.2.3 Réglage des contributions de la coupe normalisée et des contraintes

Nous proposons d'intégrer un coefficient γ afin de pondérer l'impact des contraintes sur la structure originale des données. Cependant, il est tout d'abord nécessaire d'effectuer une normalisation permettant de rendre J plus facilement interprétable. En effet, l'expression de J_{MNCut} : $z_k^T \overline{L}_2 z_k$ appartenant à $[0,1]$ et celle du critère de contraintes J_{PC} : $z_k^T Q z_k$ appartenant à $[\lambda_{Qmin}, \lambda_{Qmax}]$, nous proposons de normaliser la matrice Q en utilisant la différence entre sa valeur propre maximale λ_{Qmax} et celle minimale λ_{Qmin} :

$$\overline{Q} = \frac{Q - \lambda_{Qmin}}{\lambda_{Qmax} - \lambda_{Qmin}}. \tag{4.9}$$

Grâce au terme $\gamma \in [0,1]$, jouant le rôle de balance, les valeurs du critère J appartiennent désormais à $[0,1]$, et le problème final peut alors s'écrire :

$$\min_{Z} \sum_{k=1}^{K} ((1-\gamma).{z_k}^T \overline{L}_2 z_k - \gamma.{z_k}^T \overline{Q} z_k), \textbf{ s.c.} \ {z_k}^T z_k = 1. \tag{4.10}$$

L'optimisation de la fonction objectif globale J peut donc se résumer à la résolution du système propre standard suivant :

$$((1-\gamma).\overline{L}_2 - \gamma.\overline{Q})z = \lambda z, \tag{4.11}$$

c'est-à-dire, l'extraction des vecteurs propres de la matrice $(1-\gamma).\overline{L}_2 - \gamma.\overline{Q}$. Contrairement à la méthode **FCSC** proposée dans [Wang and Davidson, 2010] par Wang et Davidson, l'espace spectral de projection est obtenu à partir d'un algorithme de projection spectrale traditionnelle. Ceci permet donc de toujours obtenir une solution possible au problème d'optimisation sous contraintes, même pour un problème multi-groupes [Wacquet et al., 2011c].

4.2.4 Solution retenue

Les vecteurs z obtenus à partir de la résolution du système propre standard 4.11 sont projetés sur la sphère unité ($F_{ij} = \frac{z_{ij}}{\sqrt{\sum_j z_{ij}^2}}$). La solution retenue est alors le second plus petit vecteur propre f_2 dans le cas $K = 2$. En effet, le premier vecteur (f_1) est constant et représente donc une solution triviale. La partition finale est alors obtenue en partitionnant les données selon le signe des valeurs de f_2.

Dans le cas $K > 2$, nous maintenons l'usage des K premiers vecteurs propres, en considérant que le vecteur constant f_1 n'influence pas la construction de l'espace spectral. Ces K premiers vecteurs propres sont ensuite utilisés pour partitionner les données grâce à l'algorithme des K-moyennes.

L'algorithme proposé dans sa version multi-groupes est résumé dans l'algorithme 12.

4.2.5 Pondération des contributions "Must-Link" et "Cannot-Link"

L'algorithme **SSSC** proposé est capable d'intégrer des pondérations sur les ensembles de contraintes de comparaison. Ces pondérations permettent d'affiner le poids des contraintes "Cannot-Link" vis-à-vis des contraintes de type "Must-Link", et inversement. Le critère tenant compte des contraintes de comparaison peut alors s'écrire comme suit :

$$\begin{aligned} J_{PC} &= -\frac{\alpha}{|\mathcal{CL}|} \sum_{\{x_i,x_j\}\in\mathcal{CL}} \sum_{k=1}^{K} z_{ik}.z_{jk} + \frac{\beta}{|\mathcal{ML}|} \sum_{\{x_i,x_j\}\in\mathcal{ML}} \sum_{k=1}^{K} z_{ik}.z_{jk} \\ &= \sum_{k=1}^{K} \left[-\frac{\alpha}{|\mathcal{CL}|} \sum_{\{x_i,x_j\}\in\mathcal{CL}} z_{ik}.z_{jk} + \frac{\beta}{|\mathcal{ML}|} \sum_{\{x_i,x_j\}\in\mathcal{ML}} z_{ik}.z_{jk} \right]. \end{aligned} \tag{4.13}$$

Algorithme 12 Algorithme **SSSC** proposé

Entrées : matrice de similarités W, matrice de contraintes Q, nombre de groupes $K > 2$

Pré-traitement

1. Fixer $w_{ii} = 0$.
2. Fixer la valeur du coefficient γ.
3. Construire le graphe de données $G(V,E,W)$.
4. Construire la matrice de degrés $D \in \Re^{N\times N}$: $d_{ii} = \sum_j w_{ij}$.
5. Construire la matrice de pondération des contraintes Q :

$$Q_{ij} = \begin{cases} -\frac{\alpha}{|\mathcal{CL}|} & \text{si } \{x_i,x_j\} \in \mathcal{CL}, \\ +\frac{\beta}{|\mathcal{ML}|} & \text{si } \{x_i,x_j\} \in \mathcal{ML}, \\ 0 & \text{sinon.} \end{cases}$$

6. Calculer les valeurs propres minimales et maximales (notées λ_{Qmin} et λ_{Qmax}) de Q.
7. Calculer la matrice de pondération des contraintes $\overline{Q}$: $\overline{Q} = \frac{Q-\lambda_{Qmin}}{\lambda_{Qmax}-\lambda_{Qmin}}$

Représentation spectrale

8. Calculer la matrice Laplacienne normalisée symétrique $\overline{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$.
9. Extraire les K plus petits vecteurs propres $\{z_1,\ldots,z_K\}$ de la matrice :

$$(1-\gamma)\overline{L}_2 - \gamma\overline{Q},$$

10. Construire la matrice $Z = [z_1,\ldots,z_K] \in \Re^{N\times K}$.
11. Normaliser les lignes de Z afin qu'ils aient une longueur unité (projection sur la sphere unité) :

$$F_{ij} = \frac{Z_{ij}}{\sqrt{\sum_j Z_{ij}^2}}. \tag{4.12}$$

Partitionnement

12. Appliquer un algorithme de partitionnement sur les lignes de la matrice F.
13. Classer chaque objet de $\mathcal{X}$ en fonction du groupe auquel appartient l'image correspondante dans F.

Sortie : Partition $C = \{C_1,\ldots,C_K\}$

Les poids α et β sont utilisés afin de contre-balancer les contributions des contraintes "Must-Link" et "Cannot-Link". Dans [Zhang et al., 2007], Zhang et al. intégrent des coefficients de pondération des contraintes très similaires dans leur méthode d'analyse contrainte en composantes principales. L'expression du critère de contribution des contraintes J_{PC} sous la forme d'un produit matriciel, est alors réalisée en définissant une matrice de pondération Q comme suit :

$$Q_{ij} = Q_{ji} = \begin{cases} -\frac{\alpha}{|\mathcal{CL}|} & \text{si } \{x_i, x_j\} \in \mathcal{CL}, \\ +\frac{\beta}{|\mathcal{ML}|} & \text{si } \{x_i, x_j\} \in \mathcal{ML}. \\ 0 & \text{sinon.} \end{cases} \tag{4.14}$$

L'optimisation de la fonction objectif globale J est ensuite identique au cas où $\alpha = \beta = 1$, c'est-à-dire qu'il s'agit d'extraire les K premiers vecteurs propres de la matrice $(1-\gamma).\overline{L}_2 - \gamma.\overline{Q}$.

4.2.6 Discussions vis-à-vis des méthodes de la littérature

Dans cette section, nous confrontons l'algorithme semi-supervisé proposé avec ceux de la littérature, afin de mettre en évidence les apports de notre méthode. Pour cela, nous choisissons de comparer quelques algorithmes utilisant les contraintes de comparaison par paires d'objets, présentés dans le chapitre 3.

Contrairement à la méthode d'analyse contrainte en composantes principales [Zhang et al., 2007], l'algorithme **SSSC** proposé permet d'obtenir une représentation non linéaire des données. Cette dernière offre également l'avantage d'intégrer des pondérations pour chacun des objets (grâce à la matrice de similarités). En effet, la méthode d'ACP contrainte fixe les poids des données non contraintes à $\frac{1}{N^2}$ (avec N étant égal au nombre total d'objets).

Bien que l'algorithme de projection contrainte préservant la structure locale (noté CLPP [Cevikalp and Verbeek, 2008]) soit une méthode non linéaire (au même titre que la méthode **SSSC**), il est important de noter qu'il est principalement utilisé comme outil de visualisation des données. De plus, comme montré dans le chapitre 3, cette méthode ne permet pas de pondérer différemment l'apport des objets non contraints, ceux contraints en "Must-Link" et ceux contraints en "Cannot-Link". Cet algorithme propage alors ces contraintes de comparaison grâce aux règles de transitivité ("Must-Link") et d'héritage ("Cannot-Link"). Pour notre algorithme, nous choisissons volontairement de ne pas propager ces contraintes afin d'éviter la propagation d'informations aberrantes.

Par comparaison avec la méthode proposée dans [Kamvar et al., 2003], il est possible de montrer que notre algorithme semi-supervisé offre l'avantage de pondérer l'impact des contraintes de comparaison. En effet, Kamvar et al. fixe les valeurs de similarités à 1 pour les objets contraints en "Must-Link" et à 0 pour les objets contraints en "Cannot-Link". De plus, cette approche ne permet pas de tenir compte de la structure locale originale des données. La matrice Laplacienne obtenue est donc, par nature, différente de celle obtenue par notre algorithme.

La dernière comparaison concerne la méthode **SSSC** proposée et l'algorithme de classification spectrale contrainte et flexible de Wang et Davidson [Wang and Davidson, 2010]. Bien que ces deux méthodes intégrent un paramètre de pondération des contraintes, Wang et Davidson utilisent une formalisation Lagrangienne afin de résoudre leur problème d'optimisation sous contraintes. Ceci peut alors conduire à une absence de solutions possibles pour certaines valeurs du paramètre de pondération, dans le cas multi-groupes (il n'est pas toujours possible d'obtenir K valeurs propres généralisées positives associées à K vecteurs propres généralisés). La pertinence de notre méthode réside donc dans le fait de toujours obtenir une solution quelque soit la valeur du paramètre de pondération.

Ces différentes comparaisons permettent donc de montrer les apports de la méthode semi-supervisée proposée, ainsi que la pertinence de la technique d'optimisation employée.

4.3 Résultats expérimentaux

Dans cette section, notre méthode de classification spectrale semi-supervisée (notée **SSSC**) est appliquée, dans un premier temps, sur quelques exemples synthétiques illustratifs, puis sur des bases de données publiques appartenant aux archives UCI. Pour chaque base de données, des contraintes de comparaison sont générées à partir des étiquettes de classe connues, et les résultats sont analysés en utilisant le critère objectif J_{MNCut}, ou encore les taux de contraintes respectées. Ces résultats sont ensuite comparés avec ceux obtenus par d'autres méthodes semi-supervisées.

4.3.1 Algorithmes proposés pour la comparaison

Pour toutes les expérimentations, l'algorithme proposé est comparé avec une méthode non supervisée et deux autres méthodes de classification semi-supervisée (présentées dans le chapitre 3) :

- algorithme de classification non supervisée : **SC** (Spectral Clustering). L'algorithme de classification spectrale traditionnelle (algorithme de Ng et al. [Ng et al., 2002]) est utilisé comme référence dans le but d'évaluer l'impact de l'ajout des contraintes de comparaison sur la classification et donc sur la partition finale ;

- algorithmes de classification semi-supervisée :
  - une méthode modifiant directement les valeurs de la matrice de similarités : **SL-$\overline{L}_2$**. Cette méthode est une version dérivée de l'algorithme **SL** [Kamvar et al., 2003], pour

laquelle la matrice Laplacienne est remplacée par celle utilisée dans notre méthode (c'est-à-dire $\overline{L}_2 = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$) ;

– une méthode utilisant une optimisation sous contraintes : **FCSC-θSP**. Cette méthode est une version dérivée de l'algorithme **FCSC** [Wang and Davidson, 2010]), où le paramètre θ est recherché dans l'intervalle proposé par les auteurs.

Nous avons retenu ces méthodes puisque, comme démontré dans le chapitre 3, ce sont celles obtenant les meilleurs scores de performance sur les bases UCI. Ainsi, dans le but de faciliter la comparaison des méthodes, quelques homogénéisations sont effectuées. La première d'entre elles est l'intégration de l'étape finale de projection sur la sphère unité.

En considérant qu'une information de type "Must-Link" possède la même importance qu'une information de type "Cannot-Link", mais également que les valeurs nécessaires pour les faire respecter doivent être égales, nous fixons le poids $\alpha = \beta = 1$. Les poids de chaque type de contraintes sont donc similaires et dépendent du nombre de contraintes définies (normalisation par le cardinal de $\mathcal{ML}$ et $\mathcal{CL}$).

Dans la variante proposée pour la méthode **FCSC**, la matrice Q de poids des contraintes utilisée pour les expérimentations est donc celle définie dans l'algorithme 12. De plus, l'intervalle de valeurs utilisé pour θ est défini comme étant : $[\lambda_{min} vol(G), \lambda_{max} vol(G)]$. Il est montré que cet intervalle est suffisant pour assurer l'existence de K vecteurs propres généralisés respectant les contraintes du problème d'optimisation [Wang and Davidson, 2010].

Pour la méthode proposée **SSSC**, ainsi que pour l'algorithme **FCSC-θSP**, le poids des termes de pénalité θ et γ est optimisé *a posteriori*, en discrétisant leur intervalle de définition en 100 valeurs équidistantes, et en choisissant celle qui maximise le critère suivant :

$$E = (1 - J_{MNCut}) + ML_{resp} + CL_{resp}, \tag{4.15}$$

où ML_{resp} et CL_{resp} représentent les taux respectifs de contraintes "Must-Link" et "Cannot-Link" respectées. En ce sens, nous cherchons à optimiser à la fois la valeur de la coupe (c'est-à-dire, à préserver la structure originale des données), mais également à faire respecter les contraintes définies.

4.3.2 Exemple illustratif

Afin d'étudier l'effet des contraintes sur le partitionnement des données, nous proposons d'utiliser les contraintes de comparaison par paires d'objets dans un problème multi-groupes.

Description du problème étudié

La base de données synthétiques est composée de 400 points construits à partir d'un mélange de cinq distributions gaussiennes, comme montré sur la figure 4.2(a). Chaque distribution est représentée par une couleur et un symbole différents. La proportion de chacune d'entre elles est de $\frac{1}{5}$. Pour cet exemple, le nombre de groupes K désirés est fixé à 4.

Comme illustré sur la figure 4.1, deux types de contraintes sont alors considérés : deux contraintes de type "Must-Link" (lignes pleines) entre des objets appartenant à des groupes différents, et une contrainte de type "Cannot-Link" (ligne pointillée) entre deux objets appartenant à un même groupe gaussien. Ces contraintes de comparaison par paires sont choisies délibérément de façon à obtenir une partition différente de celle obtenue par minimisation de la coupe naturelle, qui est la solution de l'algorithme de classification spectrale traditionnel (**SC**). Nous souhaitons donc fusionner les classes qui comportent des points contraints en "Must-Link" et scinder la classe comportant un couple de points appariés en "Cannot-Link".

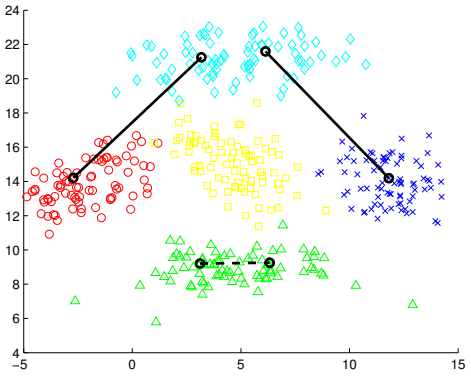


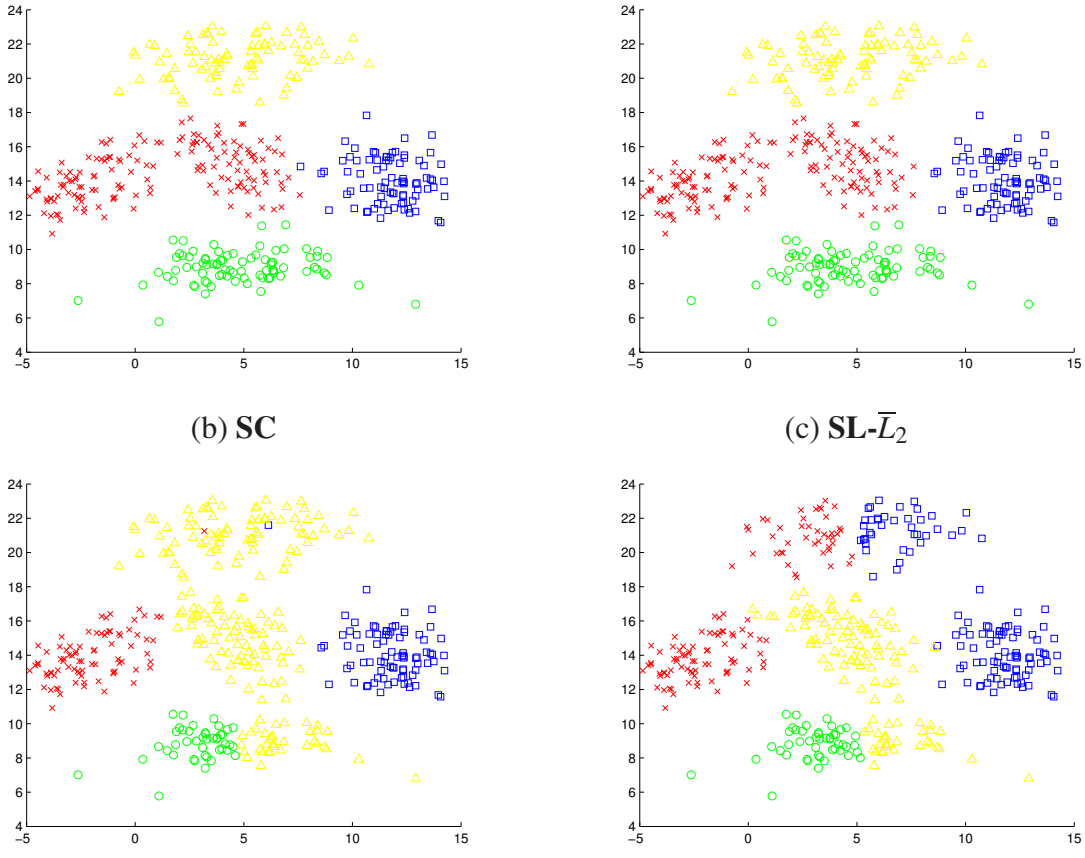
FIGURE 4.1 – Données originales (cinq groupes gaussiens), avec deux contraintes *ML* (lignes pleines) et une contrainte *CL* (ligne pointillée).

Pour cet exemple, la matrice de similarités est construite à partir d'un noyau gaussien, prenant comme arguments un paramètre de dispersion locale σ fixé à 1, et une distance d définie

comme étant la distance euclidienne.

Résultats obtenus et discussions

La figure 4.2 montre les partitions obtenues pour les quatre méthodes testées. Alors que la méthode **SL-$\overline{L}_2$** ne parvient pas à ”casser” le groupement naturel obtenu par la méthode de classification spectrale originale, les algorithmes **SSSC** et **FCSC-θSP** réussissent à faire respecter les trois contraintes définies, comme montré sur les figures 4.2(d) et 4.2(e). Cependant, la méthode proposée **SSSC** se démarque de par son pouvoir de généralisation aux objets voisins. La combinaison de ces trois contraintes permet donc de diriger le processus de partitionnement, même dans le cas d'une contrainte de type ”Cannot-Link” non naturelle (son objectif est de scinder le groupe représenté par des cercles verts en deux sous-groupes).



(b) **SC** (c) **SL-$\overline{L}_2$**

(d) **FCSC-θSP** (e) **SSSC**

FIGURE 4.2 – Résultats de partitionnement sur cinq groupes gaussiens ($K = 4$), avec deux contraintes *ML* (lignes pleines) et une contrainte *CL* (ligne pointillée).

Dans le but de compléter l'analyse de ces résultats de partitionnement, quelques indicateurs de performance tels que les valeurs de coupe et les pourcentages totaux de contraintes respectées ($ML_{resp} + CL_{resp}$), sont présentés dans le tableau 4.1.

Méthodes	J_{MNCut}	$\%total(ML_{resp} + CL_{resp})$
SC	**0.004**	0.0
SL-$\overline{L}_2$	0.013	0.0
FCSC-θSP	0.048	**100.0**
SSSC	0.031	**100.0**

TABLE 4.1 – Valeur de coupe et pourcentage total de contraintes respectées, pour les différentes méthodes, avec deux contraintes *ML* et une contrainte *CL*.

Les valeurs de coupe les plus élevées sont donc obtenues pour les méthodes **SSSC** et **FCSC-θSP** (0.031 et 0.048 respectivement). Ceci peut s'expliquer par le fait que ces algorithmes sont les seuls à faire respecter la totalité des contraintes de comparaison par paires d'objets. Cela entraîne donc l'apparition d'une nouvelle partition, différente de celle naturelle. La structure originale des données n'est alors plus totalement préservée, ce qui implique une augmentation de la valeur de coupe. Cependant, pour **SSSC**, cette dernière est plus faible que celle obtenue pour **FCSC-θSP**, comme montré dans le tableau 4.1.

Cette expérience montre que l'introduction de connaissances contextuelles est correctement gérée par la méthode **SSSC** proposée. La comparaison avec l'algorithme de classification spectrale traditionnel montre que l'apport d'information *a priori*, sous la forme de contraintes de comparaison par paires, permet d'approcher de manière plus précise la partition optimale. De plus, dans cet exemple, la méthode **SSSC** proposée parvient à faire respecter conjointement les contraintes (contrairement à **SL-$\overline{L}_2$**) et un score de coupe minimal (contrairement à **FCSC-θSP**).

D'autres expérimentations ont été menées à $K = 2$, montrant la capacité de la méthode proposée par Kamvar et al. à faire respecter quelques contraintes mais avec aucun pouvoir de généralisation (cf. Annexe A).

4.3.3 Application aux bases de données UCI

Dans cette section, notre méthode de classification spectrale semi-supervisée multi-groupes (**SSSC**) est appliquée sur quelques bases de données bien connues dans le domaine de la classification (bases de données UCI). De la même façon que pour les expérimentations réalisées dans le chapitre 3, pour chacun des exemples, quelques pourcentages d'objets sont sélection-

nés de manière aléatoire, dans le but de construire des ensembles d'étiquettes d'objets. Ensuite, après génération de toutes les paires possibles d'objets, il est aisé de déduire les ensembles de contraintes $\mathcal{CL}$ et $\mathcal{ML}$. A chaque pourcentage testé, l'ensemble des objets sélectionnés au test précédent est conservé. Il suffit alors d'ajouter un nouvel ensemble d'objets afin d'atteindre le pourcentage désiré d'étiquettes supposées connues. Pour les expérimentations, la totalité des contraintes générées est utilisée.

Les métriques d'évaluation des partitions, utilisées dans cette section, sont les suivantes : l'indice de Rand et la valeur de F-mesure (présentés dans le chapitre 2). Pour ces deux scores de performance, les résultats obtenus sont moyennés sur 10 répétitions du processus de génération aléatoire des contraintes.

Présentation des bases de données UCI

Le tableau 4.2 présente les six bases de données utilisées. Dans cette section, nous décrivons briévement chaque base. Pour chaque exemple, la matrice de similarités est construite à partir d'un noyau gaussien pour lequel le paramètre de dispersion locale des données σ est égal à la moyenne des variances des attributs.

	Nb. Objets	Nb. Attributs	Nb. Classes
Glass1	214	9	2
Hepatitis	80	19	2
Ionosphere	351	34	2
Wine	178	13	3
Dermatology	366	34	6
Glass2	214	9	6

TABLE 4.2 – Bases de données UCI.

Les bases "Glass1" et "Glass2" ($\sigma = 0.7$). L'étude de la classification des différents types de verres a été motivée par l'avancée des enquêtes criminologiques. En effet, sur la scène d'un crime, le verre présent peut être utilisé comme une preuve. C'est pourquoi les mesures de quantité des élements constituants (Magnésium, Aluminum, Silicon, etc.) du verre, mais également les propriétés physiques ou optiques de celui-ci (comme l'indice de réfraction) sont importants. La base de données à disposition est donc composée de 214 observations décrites par 9 attributs ($N = 214$, $M = 9$), et réparties :

– soit en deux classes ($K = 2$) : "Verre de fenêtre" (163 objets) et "Autre type de verre" (51 objets). Cette base est nommée "Glass1" ;

– soit en six classes ($K = 6$) : "Verre 1" (70 objets), "Verre 2" (17 objets), "Verre 3" (76 objets), "Verre 4" (13 objets), "Verre 5" (9 objets) et "Verre 6" (29 objets). Cette base est nommée "Glass2".

La base "Hepatitis" ($\sigma = 457.2$). Cette base de données est constituée des observations réalisées sur 155 patients. Chaque observation est composée de 19 attributs (booléens, entiers ou réels) qui peuvent être soit d'ordre personnel (âge, sexe, etc.), soit d'ordre symptomatique (fatigue, malaise, etc.), soit d'ordre chimique (présence de bilirubine, d'albumine, etc.). Cependant, certains de ces attributs étant manquants, nous choisissons de supprimer les observations correspondantes. Nous obtenons alors 80 observations réparties en deux classes ($K = 2$) : "Vivant" (67 objets) ou "Mort" (13 objets).

La base "Ionosphere" ($\sigma = 0.3$). Des données radar ont été recueillies par un système composé d'un réseau de 16 antennes hautes-fréquences. Les cibles sont les électrons libres dans l'ionosphère. Les radars qualifiés de "corrects" sont ceux pour lesquels il existe un retour du signal permettant de mettre en évidence une structure de l'ionosphère. Ceux qualifiés de "incorrects" sont ceux dont les signaux émis passent à travers l'ionosphère. Les signaux reçus sont traités en utilisant une fonction d'autocorrélation dont les arguments sont le temps et le nombre des impulsions. Pour le système utilisé, un signal est décrit par 17 impulsions. Ces derniers sont décrits par deux attributs chacun, correspondants aux valeurs complexes renvoyées par la fonction résultant du signal électromagnétique complexe. La base de données à disposition est composée de 351 observations décrites par 34 attributs ($N = 351$, $M = 34$), et réparties en deux classes ($K = 2$) : "Correct" (225 objets) ou "Incorrect" (126 objets).

La base "Wine" ($\sigma = 7645.5$). La base de donnée "Wine" contient les résultats d'une analyse chimique des vins produits dans une même région en Italie, mais provenant de trois cultivateurs différents. L'analyse détermine les quantités de 13 constituants majeurs de chacun des trois types de vin. La base de données est donc constituée de 178 observations caractérisées par 13 attributs ($N = 178$, $M = 13$). Ces données sont divisées en trois classes ($K = 3$) : "Vin 1" (59 objets), "Vin 2" (71 objets) et "Vin 3" (48 objets).

La base "Dermatology" ($\sigma = 7.6$). Cette base de données est constituée des observations réalisées sur 366 patients. Ces observations sont, tout d'abord, composées de 12 attributs d'ordre cliniques (âge, démengeaisons, atteinte de la muqueuse buccale, etc.), mais également de 22 attributs histo-pathologiques (exocytose, spongiose, etc.) recueillis sur des échantillons de peau. Les valeurs des attributs histo-pathologiques sont déterminées par une analyse microscopique

des échantillons. La base de données contient donc 366 observations décrites par 34 attributs ($N = 366$, $M = 34$), et réparties en six classes ($K = 6$) : "Psoriasis" (112 objets), "Dermatite séborrhéique" (61 objets), "Lichen plan" (72 objets), "Pityriasis rosé" (49 objets), "Dermatite Chronique" (52 objets) et "Pilaris de rubra de Pityriasis" (20 objets).

Résultats obtenus et discussions

La figure 4.3 montre les mesures de performance de chacune des méthodes appliquées sur les bases de données UCI, en terme d'indice de Rand et en fonction du pourcentage d'étiquettes connues. Comme nous pouvons l'observer :

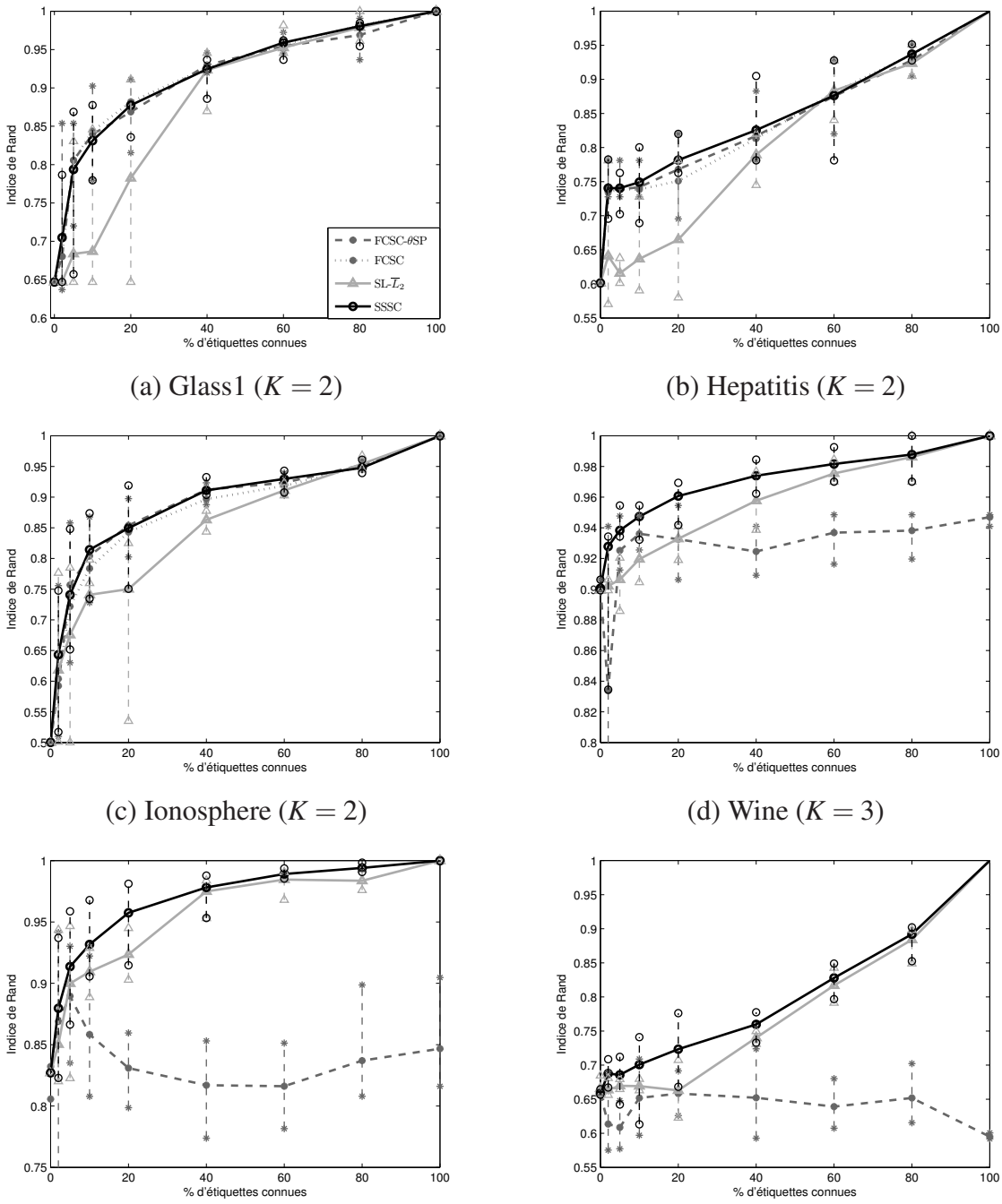
- globalement, la totalité des méthodes parviennent à améliorer de manière significative le résultat obtenu par la classification spectrale traditionnelle (correspondant à l'abscisse égal à 0). Dans certains cas ("Glass1", "Ionosphere" et "Dermatology"), augmenter le nombre de contraintes améliore globalement les performances, et cette augmentation est très rapide entre les abscisses 0% et 5%. Ceci signifie que les méthodes sont capables d'améliorer le partitionnement grâce à l'apport d'une faible quantité de contraintes.

- pour $K = 2$, les meilleurs résultats sont obtenus pour les méthodes **SSSC**, **FCSC** et **FCSC-θSP**. Leurs indices de Rand sont les plus élevés et les plus stables : ils ne diminuent pas avec le nombre de contraintes ajoutées. La méthode **SL-**$\overline{L}_2$, quant à elle, montre des performances inférieures, notamment pour de faibles pourcentages d'étiquettes connues. Cet algorithme devient intéressant, uniquement lorsque le nombre de contraintes considérées est élevé : les poids 0 et 1 semblent donc trop faibles (en valeur absolue) pour avoir un impact sur la partition finale.

- pour $K > 2$, la méthode **SSSC** obtient de meilleures performances que tous les autres algorithmes. Pour **FCSC**, le problème d'optimisation ne pouvant être résolu pour une valeur de θ donnée, les résultats obtenus pour cette méthode ne sont pas représentées sur les figures. De plus, les algorithmes **FCSC-θSP** et **SL-**$\overline{L}_2$ obtenant des scores de performance plus faibles, la méthode **SSSC** proposée, apparaît donc intéressante pour les problèmes multi-groupes.

Détails des résultats obtenus sur la base "Dermatology"

La figure 4.4 montre les résultats de performance obtenus sur la base de données "Dermatology" en termes d'indices de Rand, de valeurs de coupe *MNCut* et de pourcentages de contraintes de type "Must-Link" et "Cannot-Link" respectées pour les quatre méthodes testées :



(a) Glass1 ($K = 2$)

(b) Hepatitis ($K = 2$)

(c) Ionosphere ($K = 2$)

(d) Wine ($K = 3$)

(e) Dermatology ($K = 6$)

(f) Glass2 ($K = 6$)

FIGURE 4.3 – Indice de Rand (moyen, maximum et minimum), en fonction du pourcentage d'étiquettes connues, sur les bases de données UCI.

– pour n'importe quel pourcentage d'étiquettes connues, les méthodes **SSSC** et **SL-$\overline{L}_2$** permettent d'obtenir une partition plus proche de celle optimale que l'algorithme **FCSC-θSP**.

– jusqu'à 40% d'étiquettes connues, nous remarquons que le taux de respect des contraintes de type "Cannot-Link" pour **SL-$\overline{L}_2$** est plus faible que pour **SSSC**. Ceci peut s'expliquer par le fait que cette méthode **SL-$\overline{L}_2$** ne fixe pas de poids négatifs sur ses contraintes "Cannot-Link" (0 pour cette méthode et -1 pour **SSSC**). La pondération pour les contraintes "Cannot-Link" est donc, par nature, différente de celle utilisée dans **SSSC**. L'algorithme proposé **SSSC** permet d'obtenir des taux élevés de respect des contraintes des deux types. De plus, nous remarquons que ses valeurs de coupe *MNCut* restent très satisfaisantes (contrairement à **SL-$\overline{L}_2$**).

Quel que soit l'indice de performance considéré, notre méthode apparaît ici pertinente vis-à-vis de la base de données visée et face à ses concurrentes. Nous pouvons noter cependant que, même si les valeurs de *MNCut* sont plus faibles pour **FCSC-θ_2SP** que pour **SSSC** à partir de 60% d'étiquettes connues, les pourcentages de respect des contraintes des deux types pour la méthode proposée sont nettement plus élevés. Par ailleurs, sur les 10 répétitions du processus de génération de contraintes, les résultats obtenus par l'algorithme **SSSC** sont stables : les écarts entre les valeurs minimales et maximales sont de faibles amplitudes.

Les tableaux 4.3, 4.4 et 4.5 permettent de confronter les résultats obtenus par les différents algorithmes. Nous comparons donc, dans ces tableaux et pour la base "Dermatology", la méthode proposée avec les variantes **SL-$\overline{L}_2$** et **FCSC-θSP**.

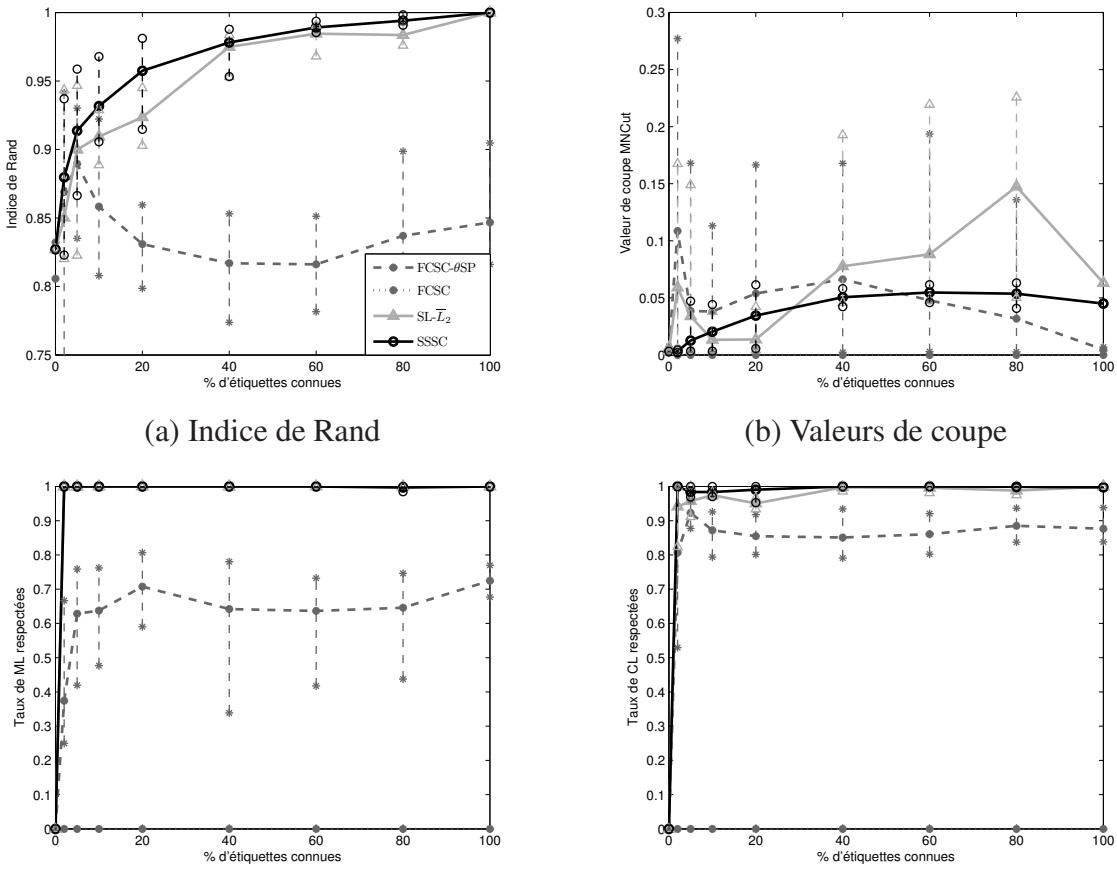
2%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	45.9%	37.7%	42.4%	41.2%
	Non Reconnus	7.9%	8.5%	5.9%	10.5%

TABLE 4.3 – Comparaison de l'algorithme **SSSC** avec **SL-$\overline{L}_2$** et **FCSC-θSP**, pour la base "Dermatology" (2% d'étiquettes connues).

5%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	72.1%	18.6%	56.3%	34.4%
	Non Reconnus	8.7%	0.6%	8.7%	0.6%

TABLE 4.4 – Comparaison de l'algorithme **SSSC** avec **SL-$\overline{L}_2$** et **FCSC-θSP**, pour la base "Dermatology" (5% d'étiquettes connues).

Lorsque le pourcentage d'étiquettes connues est de 100%, les méthodes **SSSC** et **SL-$\overline{L}_2$** semblent équivalentes puisque le pourcentage d'objets reconnus correctement est de 100.0%



(a) Indice de Rand

(b) Valeurs de coupe

(c) Taux de *ML* respectées

(d) Taux de *CL* respectées

FIGURE 4.4 – Indices de Rand, valeurs de coupe et taux de contraintes "Must-Link" et "Cannot-Link" respectées, en fonction du pourcentage d'étiquettes connues, sur la base "Dermatology".

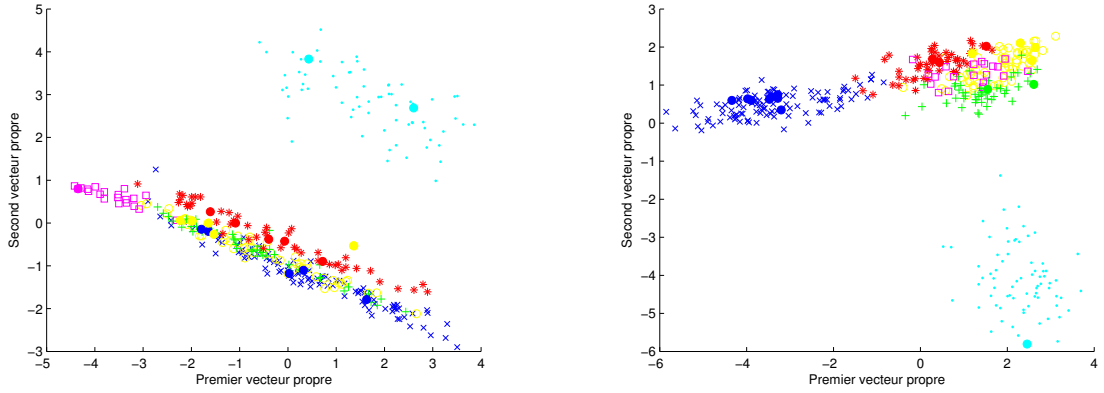
100%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	100.0%	0.0%	48.3%	51.7%
	Non Reconnus	0.0%	0.0%	0.0%	0.0%

TABLE 4.5 – Comparaison de l'algorithme **SSSC** avec **SL-$\overline{L}_2$** et **FCSC-θSP**, pour la base "Dermatology" (100% d'étiquettes connues).

pour les deux cas. En revanche, il n'en est pas de même pour l'algorithme **FCSC-θSP** qui ne reconnaît pas 51.7% d'objets.

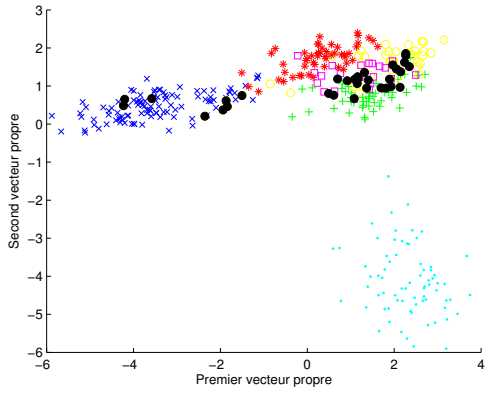
Pour un faible pourcentage d'étiquettes connues (5%), il est possible de montrer que la méthode **SSSC** proposée permet de reconnaître 18.6% d'objets non reconnus par l'algorithme **SL-$\overline{L}_2$** alors que l'inverse n'est que de 8.7%. De même, **SSSC** offre la possibilité de reconnaître 34.4% d'objets non reconnus par **FCSC-θSP** (8.7% pour l'inverse). Les résultats de partition-

nement pour 5% d'étiquettes connues, sont représentés dans l'espace obtenu par l'algorithme de projection sous contraintes préservant la structure locale des données (cf. figure 4.5).

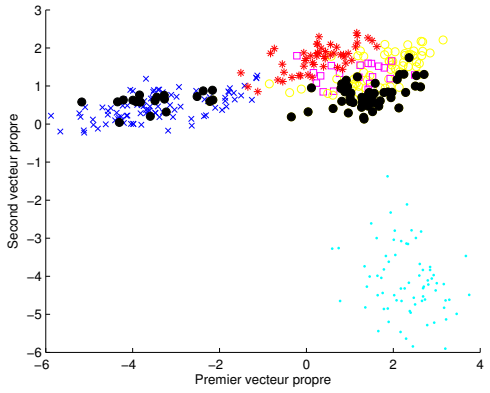


(a) LPP classique

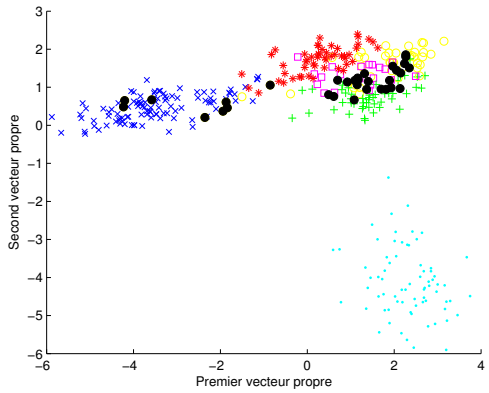
(b) LPP contraint (5%)



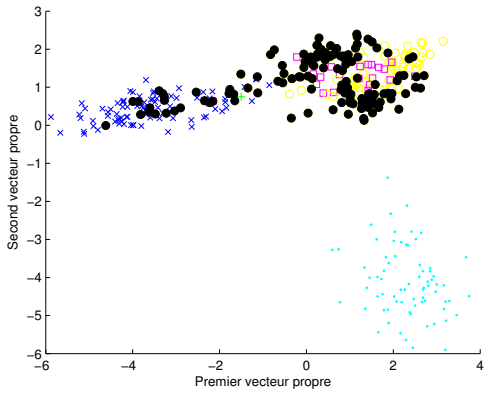
(c) **SSSC**



(d) **SL-$\overline{L}_2$**



(e) **SSSC**



(f) **FCSC-θSP**

FIGURE 4.5 – Visualisations planes des partitions obtenues sur la base "Dermatology".

La partition représentée sur les figures 4.5(a) et (b) est la "vérité terrain". Pour la figure (b), les objets contraints sont représentés par des ronds pleins (colorés selon le groupe d'appartenance). En ce qui concerne la figure (c), les ronds noirs pleins sont les objets non reconnus par **SSSC** et reconnus par **SL-$\overline{L}_2$** (inversement pour la figure (d)). De même, les ronds noirs pleins sur la figure (e), sont les objets non reconnus par **SSSC** et reconnus par **FCSC-θSP** (inversement pour la figure (f)). Comme montré sur les figures 4.5(c) et (e), la majorité des objets non reconnus par l'algorithme proposé mais correctement reconnus par les méthodes concurrentes se situent au niveau des groupes présentant un chevauchement important (étoiles rouges, cercles jaunes et '+' verts). Cependant, il est aisé de montrer que notre algorithme parvient à discriminer ces trois groupes, contrairement à **SL-$\overline{L}_2$** (les groupes composés des objets '+' verts et des cercles jaunes sont fusionnés) et à **FCSC-θSP** (les trois groupes sont fusionnés). Les résultats de comparaison entre algorithmes, mais sur les autres bases de données UCI, sont présentés dans l'annexe B.

Le tableau 4.6 montre quelques indicateurs de performance des différentes méthodes appliquées sur un exemple spécifique, *Dermatology*, pour lequel le nombre de groupes K est fixé à 6. Pour chaque pourcentage d'étiquettes connues, le meilleur résultat est représenté en caractères gras. La méthode proposée apparaît donc très compétitive par rapport aux autres méthodes testées. En effet, pour ces bases de données, la méthode **SSSC** atteint fréquemment les taux les plus élevés de contraintes respectées (supérieur à 99% pour chaque cas), tout en conservant une valeur de coupe satisfaisante pour chaque pourcentage d'étiquettes connues (presque toujours plus faibles que pour les autres méthodes).

Par exemple, pour un faible pourcentage d'étiquettes connues (2%), la proportion totale de contraintes respectées ("Must-Link" et "Cannot-Link") pour la méthode proposée **SSSC** est meilleure que pour les autres méthodes (99.9%) et la valeur de coupe est faible (0.013). De plus, cette valeur reste cohérente avec celle obtenue par la classification spectrale traditionnelle (correspondant à 0% d'étiquettes connues et égale à 0.013) et est plus faible que pour **SL-$\overline{L}_2$** (0.059) et **FCSC-θSP** (0.109). Le meilleur indice de Rand est également obtenu (0.880), tout comme la valeur de F-mesure (0.720) : le résultat final pour **SSSC** est donc plus proche de la partition optimale que pour les autres méthodes. Pour un autre pourcentage d'étiquettes connues (5%), la méthode **SSSC** ne fait pas respecter exactement toutes les contraintes (99.2%), mais ses scores de performance ainsi que sa valeur de coupe normalisée, restent les meilleurs.

% d'étiquettes connues	Méthodes	% *ML*	% *CL*	% Total	J_{MNCut}	Indice de Rand	F-mesure
0	SL-$\overline{L}_2$	/	/	/	**0.013**	**0.827**	**0.591**
	FCSC-θSP	/	/	/	**0.013**	**0.827**	**0.591**
	SSSC	/	/	/	**0.013**	**0.827**	**0.591**
2	SL-$\overline{L}_2$	**100.0**	94.1	97.1	0.059	0.850	0.683
	FCSC-θSP	37.4	80.7	59.1	0.109	0.869	0.698
	SSSC	**100.0**	**99.7**	**99.9**	**0.013**	**0.880**	**0.720**
5	SL-$\overline{L}_2$	**100.0**	95.7	97.9	0.038	0.900	0.786
	FCSC-θSP	62.8	92.3	77.6	0.039	0.890	0.769
	SSSC	**100.0**	**98.4**	**99.2**	**0.018**	**0.914**	**0.790**
100	SL-$\overline{L}_2$	**100.0**	**100.0**	**100.0**	0.063	**1.000**	**1.000**
	FCSC-θ_2SP	72.5	87.7	80.1	**0.005**	0.847	0.670
	SSSC	**100.0**	**100.0**	**100.0**	0.045	**1.000**	**1.000**

TABLE 4.6 – Mesures d'évaluation sur la base de données "Dermatology" ($K = 6$) avec différents pourcentages d'étiquettes connues.

4.4 Conclusion

Nous avons proposé un nouvel algorithme de classification spectrale semi-supervisée multi-groupes utilisant les contraintes de comparaison de types "Must-Link" et "Cannot-Link" comme des connaissances contextuelles. De la même manière que pour l'algorithme de classification spectrale traditionnel, le problème de classification se traduit par un problème d'optimisation, consistant en la minimisation d'une fonction objectif prenant en considération la coupe normalisée multiple. Ce critère est ensuite complété par l'intégration des contraintes définies.

L'algorithme proposé génère un sous-espace spectral de la même manière que l'algorithme de projection sous contraintes préservant la structure locale des données [Cevikalp and Verbeek, 2008] [Yu et al., 2010]. Le critère d'optimisation introduit des pondérations entre la contribution des données non supervisées et les taux de respect des contraintes de comparaison.

Les différents algorithmes ont été testés sur différentes bases de données dans les mêmes conditions expérimentales. Les résultats obtenus ont montré l'avantage de l'algorithme **SSSC** proposé, qui reste performant même avec une faible quantité de contraintes apportée par un expert (2 à 5% d'étiquettes connues).

Chapitre 5

Caractérisation des cellules phytoplanctoniques d'un échantillon de culture

5.1 Introduction

Aujourd'hui encore, la technique classique pour déterminer la composition phytoplanctonique d'un échantillon d'eau provenant du milieu naturel est le comptage en microscopie optique à inversion après sédimentation [Lund et al., 1958]. Cependant, cette méthode est coûteuse en temps de traitement et d'analyse (les cellules doivent être fixées) et demande du personnel spécialisé en taxonomie phytoplanctonique. De plus, elle ne permet pas de détecter les changements de l'écosystème marin à micro-échelle spatio-temporelle, combinant la haute fréquence et la qualité des informations obtenues, sans qu'il n'y ait une participation humaine compétente considérable pour prélever et classifier des échantillons.

Désormais, il existe des méthodes alternatives à la microscopie permettant de réduire les temps d'analyse, d'augmenter la précision des comptages, mais également d'effectuer un grand nombre de mesures offrant ainsi la possibilité pour les microbiologistes d'avoir de nouveaux aperçus dans le fonctionnement des communautés phytoplanctoniques. Parmi ces techniques alternatives, la cytométrie en flux offre la possibilité d'étudier les petites cellules (de l'ordre du nano et du pico-mètre), mais depuis peu permet également l'étude des cellules de grande taille (c'est-à-dire, supérieures à 20 micro-mètres). C'est une technique intéressante par sa rapidité, sa précision, sa sensibilité d'analyse, sa simplicité d'utilisation et sa stabilité.

Des méthodes, permettant une classification automatique à partir des données cytométriques, ont été developpées. Par exemple, dans [Lepage, 2004], les auteurs utilisent un perceptron multi-couches afin d'identifier et de suivre l'évolution des cellules phytoplanctoniques. Cependant, cette méthode est limitée à l'identification d'une seule et unique espèce. Plus récemment, des méthodes de reconnaissance semi-automatisée couplant la cytométrie en flux et l'analyse d'image (grâce au dispositif "FlowCam" [Buskey and Hyatta, 2006]) dans le cadre de l'étude du microplancton, ont fait leur apparition (uniquement pour les cellules de diamètre 20 à 200 micromètres). Dans [Sieracki, 2005], des expérimentations ont été menées sur 13 groupes de cellules (avec un total de 982 images). Les taux de bonne reconnaissance obtenus, grâce à l'utilisation d'une machine à vecteurs supports, sont de l'ordre de 70%. D'autres auteurs, et en particulier Malkassian et al. [Malkassian et al., 2011], utilisent, en plus des attributs fournis par le cytomètre, la décomposition en transformée de Fourier et considèrent les coefficients ainsi obtenus comme des attributs caractéristiques permettant de discriminer les cellules.

Afin de mieux cerner les données utilisées pour l'application visée, nous présentons, dans un premier temps, le principe de la cytométrie en flux et les signaux correspondants aux profils des cellules. Ensuite, nous présentons les résultats obtenus pour une base de données contenant des cellules issues d'un échantillon de culture [Guiselin, 2010] [Wacquet et al., 2011b] [Caillault et al., 2009]. En effet, une base d'espèces de diatomées (cultivées par les biologistes du laboratoire LOG) a été constituée à partir du milieu naturel. Dans un objectif pédagogique, nous travaillons ici sur une base réduite composée de sept espèces étiquetées manuellement. L'objectif est alors d'évaluer les performances des algorithmes de classification à discriminer ces espèces en se basant sur une base de données "attributs" et sur des comparaisons de profils (signaux) des cellules. Pour cela, nous nous plaçons dans un contexte supervisé en employant des classifieurs issus de différents domaines : un perceptron multi-couches, l'algorithme du plus proche voisin et une machine à vecteurs supports, mais également dans un contexte semi-supervisé, solution plus souple et plus réaliste.

5.2 La cytométrie et les signaux disponibles

Le phytoplancton est le plancton végétal microscopique qui erre à la surface des eaux et au gré des courants. La taille des différentes cellules phytoplanctoniques peut varier entre un micromètre et plusieurs millimètres (pour les colonies). Aujourd'hui, environ 7000 espèces différentes ont été recensées au niveau mondial dont 70 seraient potentiellement toxiques ou nuisibles (en particulier, près des zones côtières).

L'apparition de ces événements nuisibles a contribué au regain d'intérêt vis-à-vis des études concernant les écosystèmes marins, notamment sur le plan écologique, climatique et économique [Smayda, 1990]. Le phytoplancton, constituant un des compartiments biologiques les plus dynamiques de par sa capacité à répondre rapidement aux forçages environnementaux [Cloern, 1996], a été retenu dans le cadre de la mise en oeuvre de la Directive Européenne Cadre sur l'Eau [Dir00, 2000], comme paramètre biologique pour la classification de l'état écologique des masses d'eau.

5.2.1 La cytométrie en flux

La cytométrie en flux (notée CMF) est définie comme l'étude précise de particules isolées (cellules, bactéries, etc.) entraînées par un flux liquide, afin de les caractériser individuellement, quantitativement et qualitativement. Les premiers cytomètres en flux ont été inventés dans les années 1950. Le principe général de fonctionnement réside dans le défilement des particules à grande vitesse dans le faisceau lumineux d'un laser. La lumière alors réémise (par diffusion ou fluorescence) permet de caractériser les cellules par leurs profils.

FIGURE 5.1 – Dispositif CytoSense©.

Le cytomètre en flux sur lequel les travaux sont basés est le "CytoSense©" (cf. figure 5.1). La figure 5.2 présente le schéma fonctionnel de la cytométrie en flux. Le spécialiste recueille un échantillon d'eau de mer qu'il introduit dans le cytomètre. Une gaine liquide va alors entraîner cette solution cellulaire dans un flux afin de faire défiler chaque particule en file indienne. Ces dernières vont alors passer devant le faisceau d'un laser (un laser émettant dans le bleu spectral

est utilisé). La lumière ainsi diffusée renseigne sur la morphologie et la structure de la particule. Ici, la diffusion de la lumière étant mesurée dans l'axe du rayon incident, l'intensité du signal peut donc être corrélée avec la taille de la cellule (signal ForWard Scatter, noté FWS). En revanche, sous un angle de 90°, la mesure correspond à la structure intracellulaire de la cellule (granularité, rapport nucléo-cytoplasmique, organelles, etc.) (signal SideWard Scatter, noté SWS).

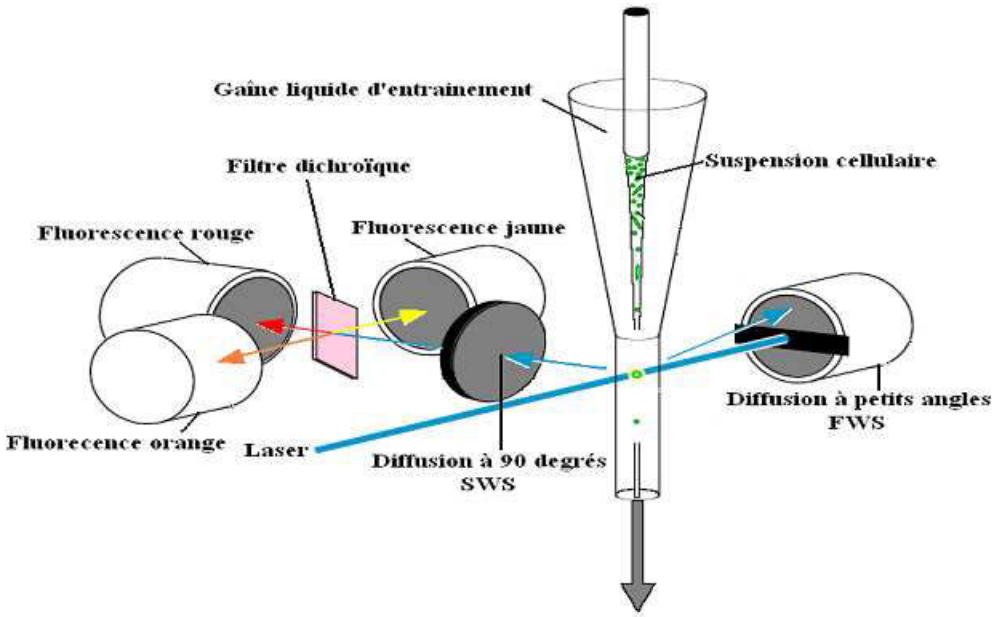


FIGURE 5.2 – Schéma fonctionnel de la cytomètrie en flux.

Enfin, lors du passage dans le faisceau du laser, des pigments tels que la chlorophylle sont excités et réémettent une fluorescence d'une longueur d'onde différente (cf. Figure 5.2). Les signaux optiques émis par fluorescence (et récupérés à 90°) sont alors séparés par des filtres optiques (et plus particulièrement, par des miroirs dichroïques) puis collectés par des photomultiplicateurs. Trois émissions de fluorescence sont détectées par le cytomètre, comme montré sur la figure 5.3 :

- la fluorescence rouge FLR ($\lambda_{em} > 620$ nm), caractéristique des pigments chlorophylliens (recueillie à travers un miroir dichroïque et un filtre passe-haut) ;
- la fluorescence orange FLO (565 nm $< \lambda_{em} <$ 592 nm), recueillie à 90° grâce à un miroir dichroïque qui la transmet sur un photomultiplicateur via un filtre passe-bande ;
- la fluorescence jaune FLY (545 nm $< \lambda_{em} <$ 570 nm), recueillie à 90° grâce à un miroir dichroïque qui réfléchit la lumière sur un photomultiplicateur via un filtre passe-bande.

Les signaux sont ensuite numérisés, prétraités et stockés en temps réel dans un ordinateur

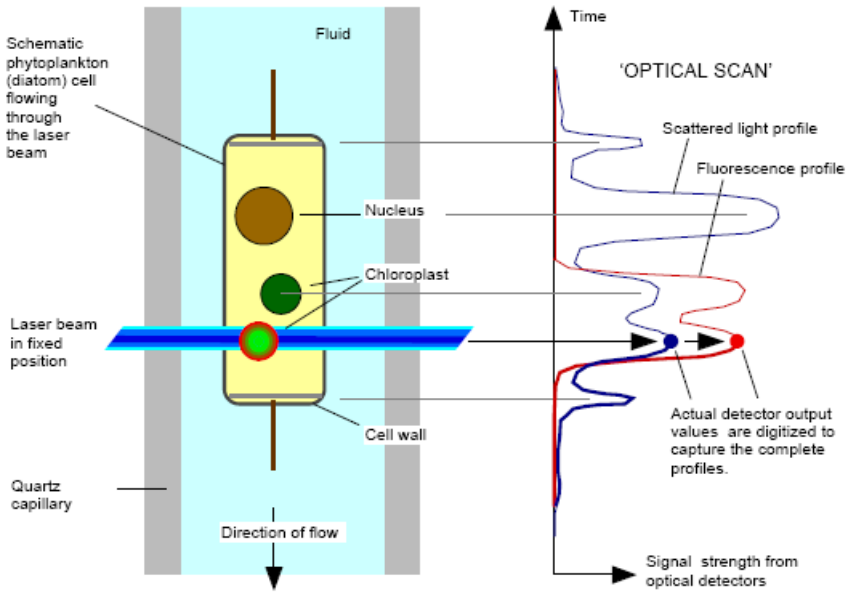


FIGURE 5.3 – Signaux envoyés par le cytomètre en flux (image issue de Cytobuoy©).

couplé au cytomètre. Il est donc possible d'établir des signatures caractéristiques de chaque type de cellules phytoplanctoniques rencontrées dans les eaux naturelles, par cytométrie en flux. Cette perspective semble intéressante et permettrait de suivre leur dynamique à haute résolution temporelle et spatiale en temps réel.

5.2.2 Logiciels de visualisation et d'étiquetage

Un logiciel d'analyse des données ("Cytoclus©") permet de calculer des statistiques monodimensionnelles (représentées par des histogrammes) et bidimensionnelles (représentées par des cytogrammes). Des attributs caractéristiques de chaque courbe sont alors obtenus : longueur, intégrale, hauteur, moyenne, inertie, centre de gravité, facteur de remplissage, asymétrie et nombre de pics. La longueur calculée correspond au temps de passage devant le faisceau lumineux du laser, et la hauteur correspond à une intensité électrique (en mV). Ces attributs caractéristiques, ainsi que la qualité pigmentaire d'une même cellule peuvent varier en fonction de la qualité lumineuse, de la température et de la quantité de sels nutritifs [Takabayashi et al., 2006].

L'utilisation naturelle de la cytométrie par les biologistes se résume alors à une discrimination manuelle et visuelle des cellules phytoplanctoniques grâce à ce logiciel. En effet, il permet d'obtenir une visualisation des cellules sur un plan, pour lequel les axes sont représentés par deux attributs caractéristiques des signaux.

Comme montré sur la figure 5.4, l'axe des abscisses représente les valeurs de l'intégrale du

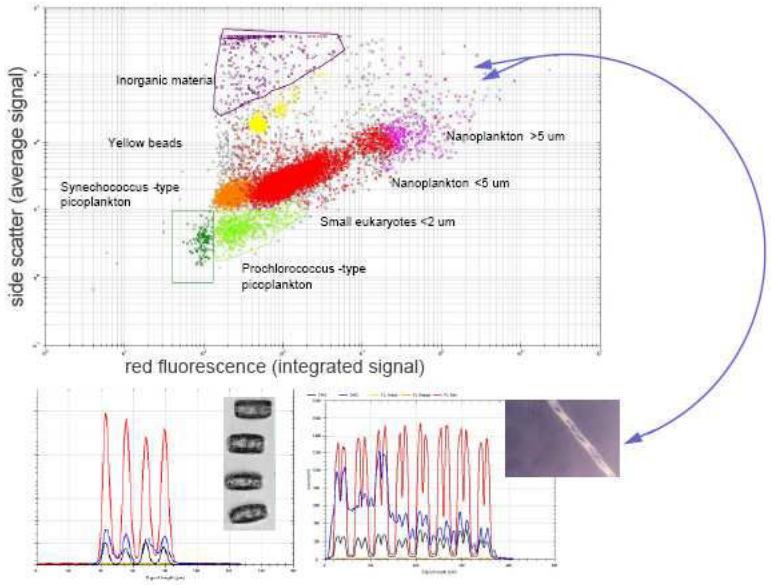


FIGURE 5.4 – Sélection manuelle des groupes taxonomiques selon les valeurs des attributs (image issue de Cytobuoy©).

signal de fluorescence rouge, et l'axe des ordonnées, les valeurs de la moyenne du signal de diffusion à 90° (SideWard Scatter). Le logiciel permet alors, grâce à cette visualisation plane, de réaliser des regroupements manuels et de comparer leurs caractéristiques aux observations microscopiques. Néanmoins, cela peut devenir fastidieux et s'avérer d'une précision incertaine lorsqu'il s'agit d'échantillons naturels.

Un logiciel de classification automatique, compatible avec le cytomètre en flux et comprenant plus de fonctionnalités, est en cours de développement ("EasyClus©" par la société "Thomas Rütten Projects"). Contrairement au "CytoClus©", ce logiciel permet d'obtenir des groupes de manière automatique. Il est utilisé, en grande partie, pour le prétraitement de données de calibration, mais également comme outil de classification supervisée et non supervisée, basée sur des rapports de fluorescences.

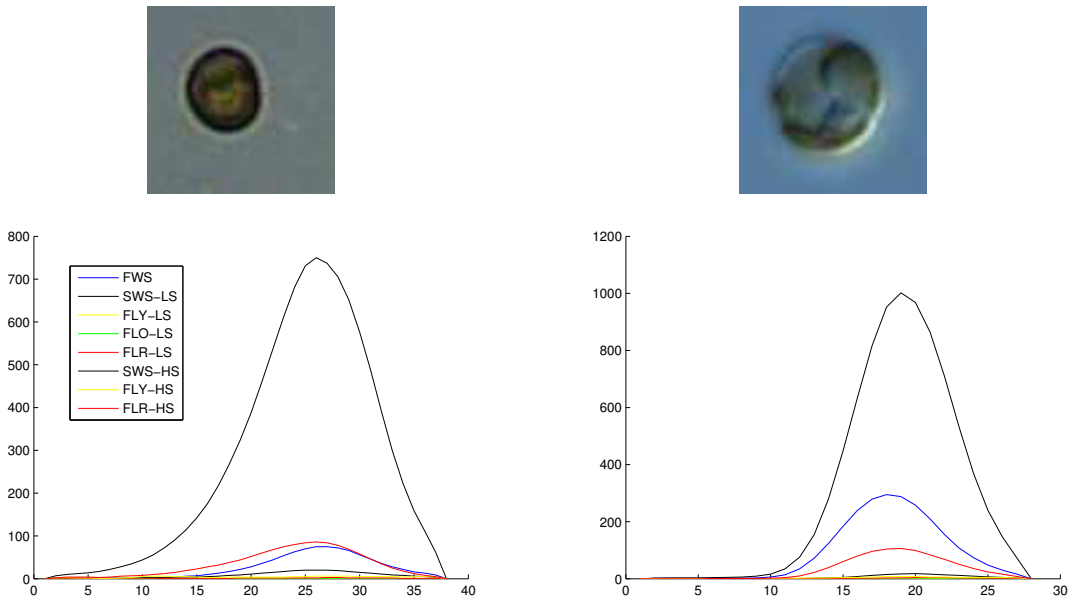
5.2.3 Variabilité des données cytométriques du phytoplancton

Variabilité inter-espèces

La principale difficulté de discrimination des cellules phytoplanctoniques réside dans le fait que des cellules appartenant à des espèces différentes peuvent présenter de fortes similitudes. Afin d'illustrer ce problème, nous proposons de visualiser les signaux cytométriques ainsi que les boîtes à moustaches des attributs caractéristiques pour deux cellules n'appartenant pas à la

même espèce.

La figure 5.5 montre les signaux cytométriques de deux espèces différentes : la première cellule (figure 5.5(a)) appartient à l'espèce *Emiliania huxleyi*, alors que la seconde (figure 5.5(b)) appartient à l'espèce *Phaeocystis globosa*. Nous pouvons noter une forte similitude des courbes en terme de longueurs, d'amplitudes mais également de formes globales.

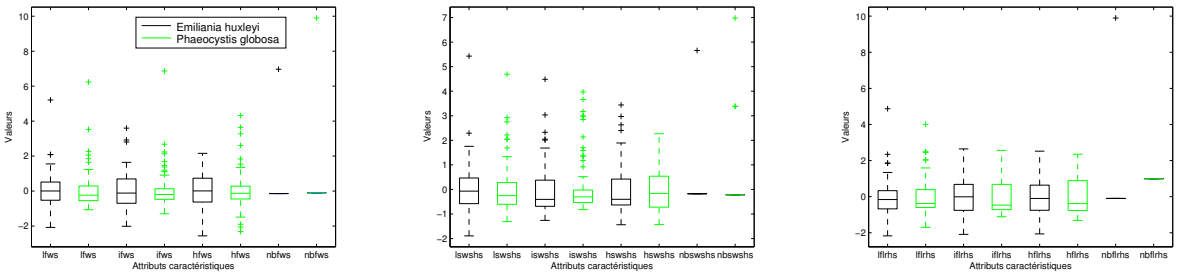


(a) *Emiliania huxleyi* (b) *Phaeocystis globosa*

FIGURE 5.5 – Illustration de la variabilité inter-espèce en cellules, grâce aux signaux profils cytométriques.

Les boîtes à moustaches de la figure 5.6, représentent la distribution statistique des attributs caractéristiques (centrés et réduits) suivants : longueur, intégrale, hauteur et nombre de pics. Nous choisissons de présenter ces distributions pour les signaux FWS (cf. figure 5.6(a)), SWS-HS (cf. figure 5.6(b)) et FLR-HS (cf. figure 5.6(c)) (HS : Haute Sensibilité du capteur). En effet, ces trois courbes sont fréquemment utilisées par les biologistes pour l'identification visuelle et manuelle des cellules.

Il est donc possible de montrer que les boîtes à moustaches de l'espèce *Emiliania huxleyi* (représentées en noir) et ceux de l'espèce *Phaeocystis globosa* (représentées en vert) présentent des chevauchements importants. La discrimination de ces deux espèces, en utilisant ces seuls attributs, semble donc très difficile.

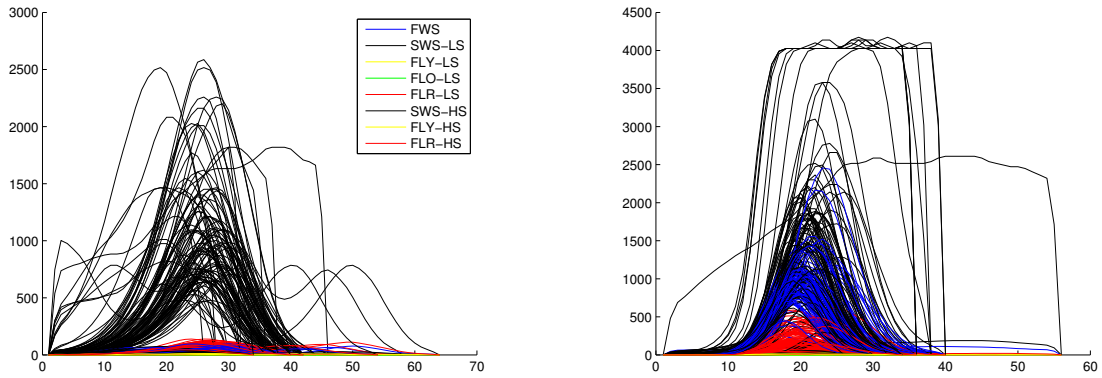


(a) Signal FWS (b) Signal SWS-HS (c) Signal FLR-HS

FIGURE 5.6 – Illustration de la variabilité inter-espèce en cellules, grâce aux attributs caractéristiques.

Variabilité intra-espèces

La seconde difficulté de discrimination est due à la variabilité importante des cellules d'une même espèce. Il existe une grande diversité de taille et de contenu pigmentaire (qualité et quantité), fonction de l'état physiologique des microalgues et des conditions du milieu. Cette difficulté est illustrée sur la figure 5.7.



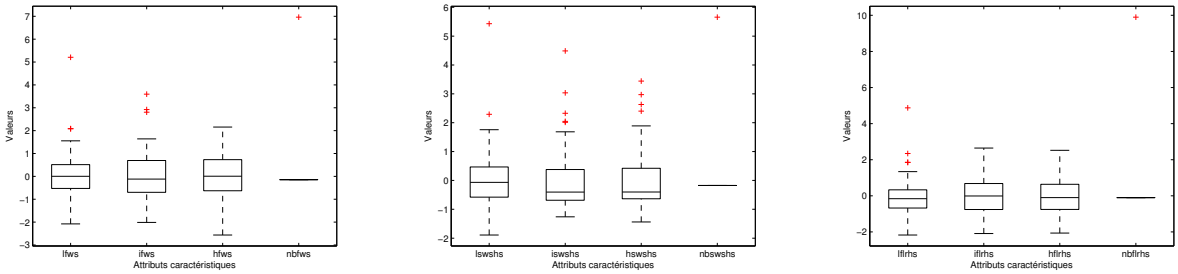
(a) *Emiliania huxleyi* (b) *Phaeocystis globosa*

FIGURE 5.7 – Illustration de la variabilité intra-espèce, grâce aux signaux profils cytométriques.

La figure 5.7(a) montre la variabilité des signaux cytométriques, par superposition des profils de chacune des cellules appartenant à l'espèce *Emiliania huxleyi*. De même, la figure 5.7(b) représente les signaux des cellules appartenant à l'espèce *Phaeocystis globosa*. Il est donc aisé de montrer que la variabilité des profils pour une même espèce est très importante. Nous pouvons, en particulier, noter que les fronts montants et descendants des courbes de forte intensité ne sont pas exactement synchrones.

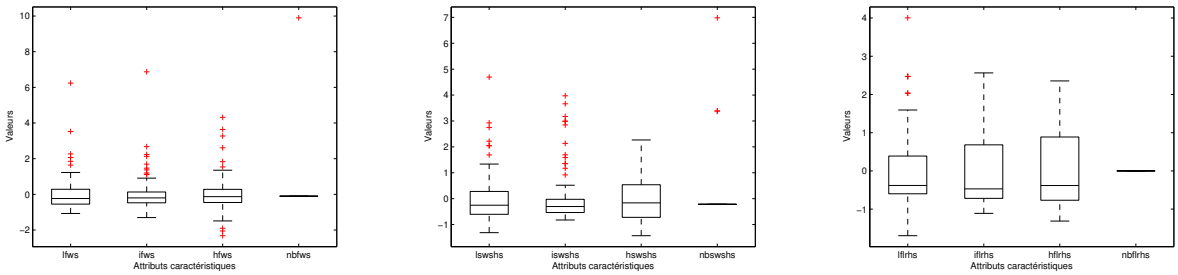
Les boîtes à moustaches, présentées sur les figures 5.8 (pour *Emiliania huxleyi*) et 5.9 (pour *Phaeocystis globosa*), montrent la taille importante de l'intervalle de valeurs pour chacun des

attributs caractéristiques des courbes FWS, SWS-HS et FLR-HS. En effet, l'écart entre les valeurs minimales et maximales est élevé (notamment pour les longueurs, les intégrales et les hauteurs). De plus, les croix rouges, représentant les valeurs extrêmes, permettent de mettre en évidence la diversité importante des profils des cellules d'une même espèce.



(a) Signal FWS (b) Signal SWS-HS (c) Signal FLR-HS

FIGURE 5.8 – Illustration de la variabilité intra-espèce pour *Emiliania huxleyi*, grâce aux attributs caractéristiques.

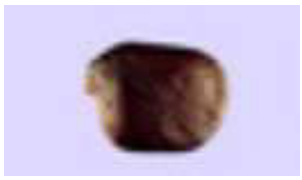


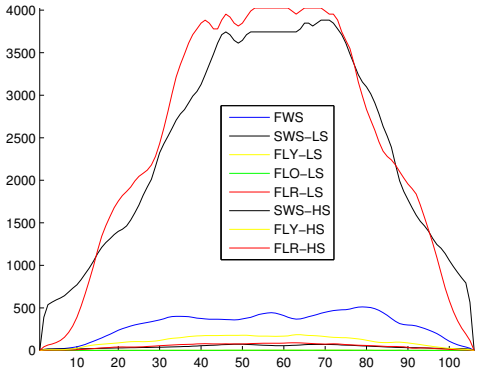
(a) Signal FWS (b) Signal SWS-HS (c) Signal FLR-HS

FIGURE 5.9 – Illustration de la variabilité intra-espèce pour *Phaeocystis globosa*, grâce aux attributs caractéristiques.

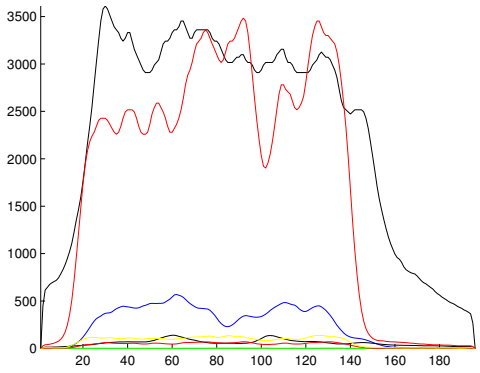
Variabilité des colonies

Certaines espèces de cellules phytoplanctoniques peuvent se trouver en colonies. Les cellules s'associent alors afin de se protéger de la prédation. Les figures 5.10(a) à 5.10(d) illustrent des exemples de profils pour l'espèce *Lauderia annulata*. Nous avons sélectionné ici, deux cas où les cellules sont seules et deux autres cas où plusieurs cellules se trouvent en colonie. Pour des cellules d'une même espèce, nous pouvons constater une variabilité non négligeable des profils intra-espèce (figures 5.10(a) et (b)). Les colonies, quant à elles, peuvent comporter un nombre de cellules très variable pouvant conduire à des profils différents (comme en attestent les figures 5.10(c) et (d)).

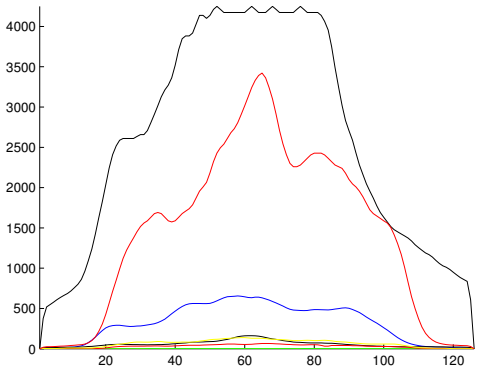




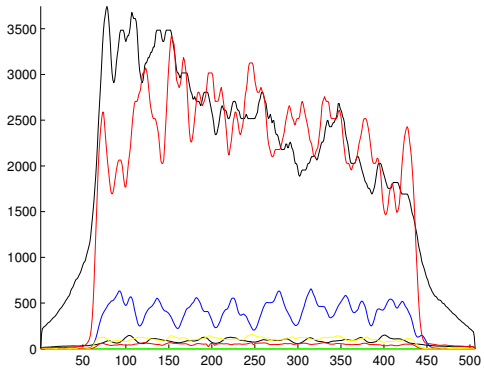
(a) Cellule 1



(c) Colonie 1



(b) Cellule 2



(d) Colonie 2

FIGURE 5.10 – Illustration de la variabilité intra-espèce en cellules et en colonies.

Il est donc possible de montrer la variabilité non négligeable des colonies d'une même espèce, selon le nombre de cellules qu'elles contiennent. Le regroupement des cellules seules et des colonies pour une même espèce apparaît donc difficile. L'alternative la plus réalisable en pratique consiste donc à rechercher des sous-groupes appartenant à la même espèce.

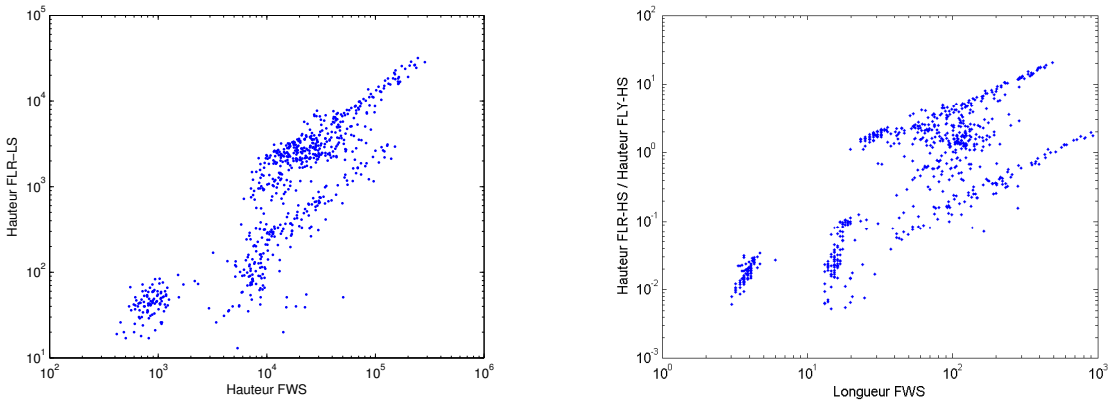
Ces différents cas de figure permettent de mettre en évidence la difficulté de discrimination des espèces phytoplanctoniques mais également la difficulté de regroupement des cellules appartenant à une même espèce.

5.3 Construction de la base de données

Dans cette section, nous présentons deux méthodes d'utilisation de l'information contenue dans les courbes cytométriques. La première est classique et consiste à synthétiser l'information recueillie par cytométrie grâce à un principe d'extraction de caractéristiques des signaux. La seconde est celle que nous proposons. Elle est basée sur l'approche naturelle des biologistes qui consiste à séparer ou regrouper des cellules à partir de la mise en correspondance visuelle des courbes cytométriques. Il s'agit donc d'une comparaison de deux ensembles de signaux bruts afin de déterminer le degré de similitude entre les deux cellules correspondantes.

5.3.1 Composition de la base utilisée

La base de données à disposition est issue d'un échantillon de cellules provenant de culture en laboratoire (LOG), et pour lequel les particules sont réparties en sept espèces phytoplanctoniques distinctes : *Chaetoceros socialis*, *Emiliania huxleyi*, *Lauderia annulata*, *Leptocylindrus minimus*, *Phaeocystis globosa*, *Skeletonema costatum* et *Thalassiosira rotula*. Chacune des espèces est représentée par 100 cellules, qui ont été préalablement étiquetées par des biologistes grâce à l'utilisation du logiciel "CytoClus©". En effet, des regroupements manuels ont été effectués à partir d'une ou de plusieurs représentations bidimensionnelles (par exemple, la longueur du signal FWS en abscisse et le rapport du maximum du signal FLR-HS et du maximum du signal FLY-HS en ordonné, comme représenté sur la figure 5.11(b)). L'analyse d'une faible quantité de cellules présentes dans chacun des groupes a alors permis d'affecter un nom d'espèce à chacun d'entre eux (par extrapolation). Il existe donc des imprécisions dans l'étiquetage, dues à l'absence de post-vérification par microscopie optique.



(a) Plan 1 (b) Plan 2

FIGURE 5.11 – Visualisations planes de l'ensemble des cellules composant l'échantillon de culture.

5.3.2 Base de données "attributs"

Afin de construire la base de données "attributs", nous utilisons un processus d'extraction d'attributs caractéristiques similaire à celui proposé par le logiciel "CytoClus©". Il consiste à définir un ensemble d'attributs afin d'obtenir des données numériques susceptibles de pouvoir discriminer les différentes espèces phytoplanctoniques. Ici, nous nous intéressons plus précisément à quatre attributs par signal, et qui sont représentés sur la figure 5.12.

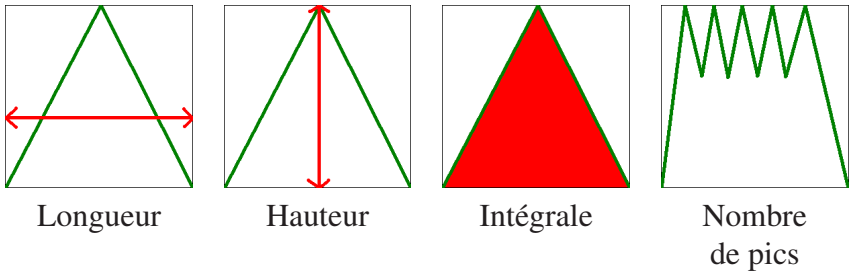


FIGURE 5.12 – Attributs extraits des signaux cytomètriques.

La matrice de données X est donc composée de N cellules décrites par quatre attributs par signal. Sachant que la cytométrie en flux permet d'obtenir huit signaux temporels, il est donc possible d'avoir à notre disposition : 8 signaux $\times$ 4 attributs caractéristiques $=$ 32 attributs. La matrice obtenue est donc de dimensions $N \times 32$. Pour nos expérimentations, nous choisissons de calculer les logarithmes de ces attributs afin d'éviter les problèmes liés à l'échelle linéaire, dus à la gamme étendue des valeurs.

5.3.3 Base de données "similarités" (ou données brutes)

Nous disposons d'une base de signaux profils des cellules phytoplanctoniques où chacune d'entre elles est représentée par huit signaux. Il est à noter que chaque cellule ayant une taille différente, les séquences temporelles sont de longueurs différentes.

Comparaison de séquences temporelles

Afin de construire la base de données "similarités", nous proposons d'utiliser la méthode d'appariement élastique (notée DTW) qui est une méthode de comparaison entre deux séquences temporelles, de longueurs différentes. La mesure obtenue traduit alors la quantité de distorsion géométrique nécessaire à la superposition des deux séquences, en tolérant des déformations temporelles locales. Pour cela, elle cherche à faire correspondre au mieux les points de l'une avec ceux de l'autre séquence, puis définit leur dissimilarité comme la moyenne des

distances entre points appariés. La souplesse de l'algorithme provient de sa capacité à apparier des points décalés dans le temps.

Plus précisément, notons $X = (x_i), i = 1, \ldots, n_x$ et $Y = (y_j), j = 1, \ldots, n_y$ les deux signaux à comparer, avec i et j leurs indices temporels. L'algorithme permet d'obtenir un appariement $App = (i_l, j_l), l = 1, \ldots, nl$ (matrice de mise en correspondance représentée par l'équation 5.1) entre les points des signaux X et Y, soumis à certaines contraintes temporelles :

$$App = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ \ldots & \ldots \\ i & j \\ \ldots & \ldots \\ n_x & n_y \end{pmatrix} \tag{5.1}$$

L'appariement optimal minimise la moyenne pondérée des distances entre points appariés :

$$J(X,Y,App) = \frac{\sum_{l=1}^{nl} d(x_{il}, y_{jl}).\eta(l)}{\sum_{l=1}^{nl} \eta(l)}, \tag{5.2}$$

avec $\eta(l)$ le poids associé à chaque paire l de l'appariement. Plusieurs conditions d'appariement sont imposées pour l'algorithme proposé :

1. Contrainte aux limites : les premiers et derniers points des signaux sont appariés : $(1,1) \in App$ et $(n_x, n_y) \in App$ (comme montré dans la matrice 5.1) ;
2. Contrainte de continuité : tout point est apparié au minimum une fois ;
3. Contrainte de monotonie : deux points d'un signal donné sont nécessairement appariés à des points identiquement ordonnés temporellement : si $(i_a, i_b) \in App$ et $(i_c, i_d) \in App$, alors $c > a$ implique $d \geq b$.

S'appuyant sur ces contraintes, l'algorithme considère chaque appariement comme un chemin dans l'espace bidimensionnel des couples $(i, j), i = 1, \ldots, n_x, j = 1, \ldots, n_y$, c'est-à-dire l'ensemble des correspondances possibles entre les points des signaux X et Y. Il cherche alors l'appariement optimal partant de la correspondance initiale $(1,1)$ et allant jusqu'à la correspondance finale (n_x, n_y) (condition 1), en assurant la minimisation du coût J (moyenne pondérée des distances entre les points appariés), comme illustré sur la figure 5.13. L'optimisation de ce critère de coût a été ramenée à la minimisation de la distance cumulée $\sum_{l=1}^{nl} d(x_{il}, y_{jl}).\eta(l)$,

grâce au choix de poids dit "symétrique" proposé par Sakoe et Chiba [Sakoe and Chiba, 1978]. Soit (i_l, j_l) la paire l de l'appariement App :

- $\eta(l) = 2$ si $l = 1$, ou si $(i_{(l-1)}, j_{(l-1)}) = (i_l - 1, j_l - 1)$ (passage par une diagonale) ;
- $\eta(l) = 1$ sinon (passage vertical ou horizontal).

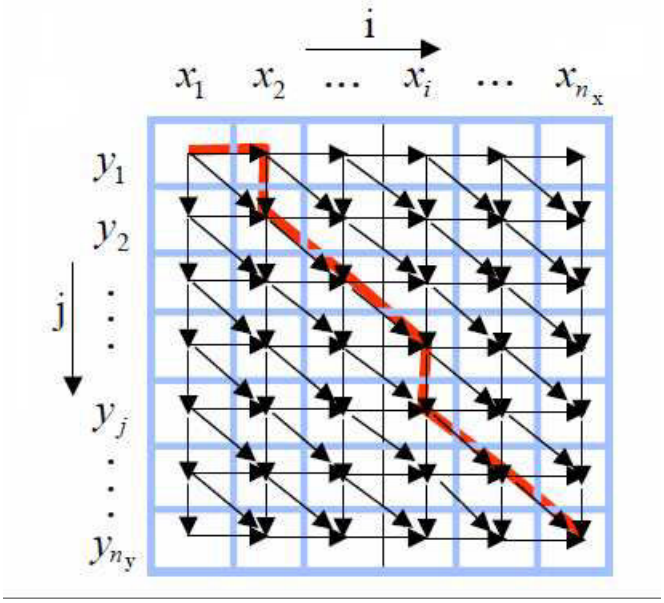


FIGURE 5.13 – Calcul du chemin dans l'espace bidimensionnel des couples (i, j).

Cependant, cette version de l'algorithme autorise des associations extrêmement souples. En effet, si les premiers points de chaque signal sont nécessairement appariés, tout comme les derniers, le premier d'un signal peut être apparié au dernier de l'autre. Afin d'autoriser des déformations temporelles locales, il est possible de définir une méthode plus restrictive, qui limite les associations à l'aide d'une fenêtre temporelle, dont la taille est définie comme un pourcentage a de la durée totale :

$$\forall (i,j) \in App,\ (i,j) \in \{(i,j_i)\,;\, i \in \{1,\ldots,n_x\},\ j_i \in [{j_i}^* - a.n_y, {j_i}^* + a.n_y]\}, \tag{5.3}$$

avec ${j_i}^*$ l'indice de correspondance linéaire défini tel que : ${j_i}^* = E(1 + \frac{(i-1)}{(n_x-1)}(n_y - 1))$ où E représente la fonction "arrondi à l'entier le plus proche". La mesure fournie par l'algorithme est donc une distance moyenne, dépendante des intensités des deux signaux.

Dans l'objectif d'obtenir une mesure de dissimilarité bornée entre 0 et 1, pour des signaux positifs, nous proposons alors de remplacer la distance d par une dissimilarité w, construite comme un rapport de distances borné [Caillault et al., 2009] :

$$Diss(x_{il}, y_{jl}) = \frac{d(x_{il}, y_{jl})}{max\{d(x_{il}, 0), d(0, y_{jl})\}}. \tag{5.4}$$

La distance entre les points appariés des courbes est donc divisée par la distance maximale de ces points avec l'ordonnée nulle, comme illustré sur la figure 5.14. Cette dissimilarité est propice à la comparaison de signaux positifs de faibles valeurs, contrairement à l'algorithme de Sakoe et Chiba [Sakoe and Chiba, 1978]. Cette fonction de dissimilarité respecte les propriétés suivantes :

1. $Diss_{ij} \in [0,1]$ (contrainte de normalisation),
2. $Diss_{ij} = 0$ si et seulement si $x_i = x_j$,
3. $Diss_{ij} = Diss_{ji}$ (contrainte de symétrie).

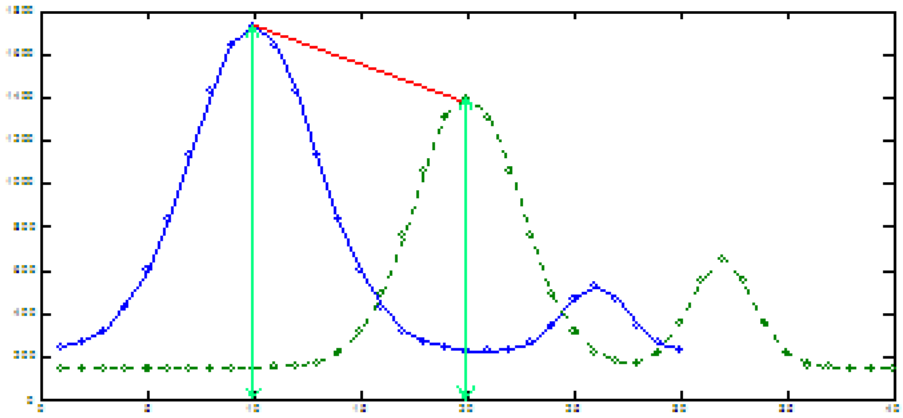


FIGURE 5.14 – Illustration de la mesure de similarité pour deux profils de courbes.

Extension aux séquences temporelles multidimensionnelles

La cytométrie en flux offrant la possibilité d'obtenir huit signaux temporels par cellule, un appariement conjoint de signaux monodimensionnels semble être une méthode intéressante et pertinente pour définir une mesure de similarité entre cellules. Nous considérons donc que les déformations temporelles doivent se faire conjointement pour chacune des courbes caractéristiques d'une même cellule.

Soient $\overline{X} = \{(\overline{x}_i, i = 1, \ldots, n_x)\}$ et $\overline{Y} = \{(\overline{y}_j, j = 1, \ldots, n_y)\}$, deux ensembles de n_c courbes cytométriques, avec $\overline{x}_i = \{(x_{ic}), c = 1, \ldots, n_c\}$ et $\overline{y}_j = \{(y_{jc}), c = 1, \ldots, n_c\}$. Il s'agit alors de

cumuler les dissimilarités sur les n_c courbes. La mesure de similarité conjointe entre ces deux ensembles est alors définie comme étant égale à :

$$SimDTW(\overline{x}_i, \overline{y}_j) = 1 - \frac{1}{n_c} \sum_{c=1}^{n_c} Diss(x_{ic}, y_{jc}). \tag{5.5}$$

La similarité *SimDTW*, traduisant la ressemblance de deux ensembles de signaux cytométriques de deux cellules différentes, est alors définie comme étant égale au coût J associé à la distance d. Cependant, dans le cadre de la cytométrie en flux, les particules peuvent passer devant le faisceau du laser, dans un sens ou dans l'autre. Nous choisissons donc de calculer la similarité entre les signaux des cellules x_1 et x_2, ainsi que la similarité entre les signaux retournés de x_1 et ceux de x_2. La valeur conservée est alors celle maximale. De plus, il est important de noter que cette mesure n'est pas robuste à la rotation. Néanmoins, pour notre application, la vitesse du flux d'entraînement des cellules étant élevée, celles-ci sont guidées de manière horizontale devant le faisceau du laser. Il est donc possible de négliger ce fait.

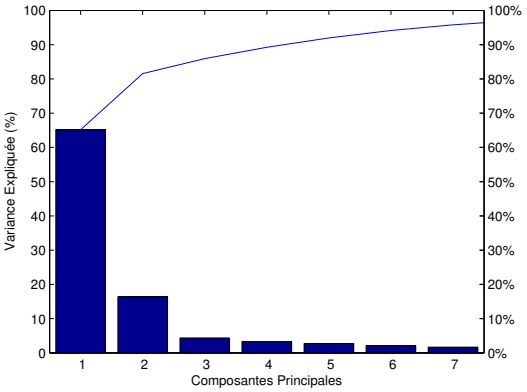
La base de données "similarités" est donc construite de la manière suivante : chaque cellule est décrite par un vecteur de similarités de dimensions $N \times 1$. La matrice obtenue est donc de dimensions $N \times N$ (avec N le nombre total de cellules).

5.4 Expérimentations sur la base "attributs"

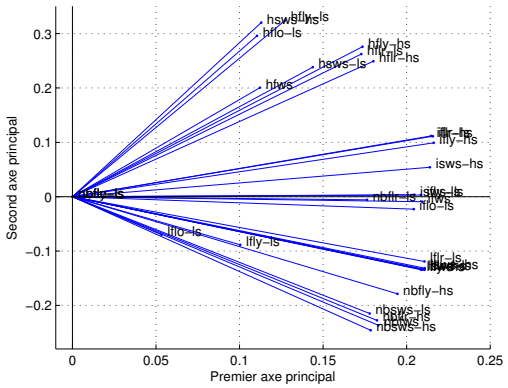
Des expérimentations ont été effectuées sur la base "attributs". Celle-ci est composée de 700×32 attributs caractéristiques, où chacune des sept espèces est représentée par cent cellules.

5.4.1 Visualisations planes

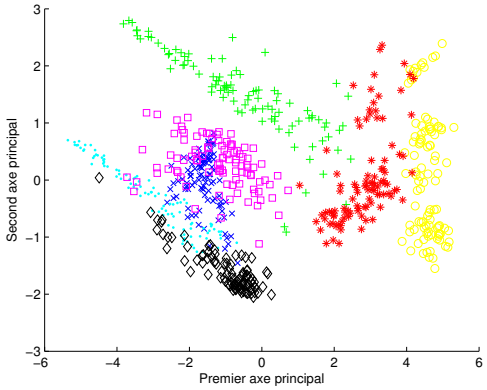
Les techniques non contraintes d'analyse en composantes principales et de projection préservant la structure locale (notée LPP [He and Niyogi, 2002]) sont appliquées sur la base composée des logarithmes des valeurs originales des attributs extraits des signaux cytométriques (afin d'éviter les problèmes liés à l'échelle linéaire). La figure 5.15 permet alors d'illustrer les degrés de corrélation entre attributs, mais également de fournir une visualisation en deux dimensions des résultats obtenus par l'analyse en composantes principales (avec 82% d'information conservée, cf. figure 5.15(c)) et par la méthode de projection préservant la structure locale des données (cf. figure 5.15(d)). Pour cette dernière, la matrice de similarités est construite à partir d'un noyau gaussien, pour lequel le paramètre de dispersion est calculé de manière locale (avec un nombre de voisins égal à 7) [Zelnik-Manor and Perona, 2004].



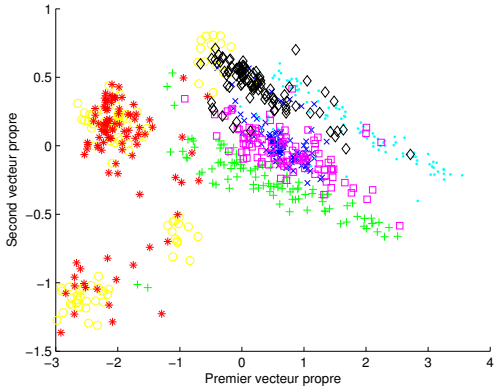
(a) Variance expliquée



(b) Corrélations



(c) ACP - Visualisation 2D



(d) LPP - Visualisation 2D

FIGURE 5.15 – Résultats obtenus par ACP et LPP (visualisation 2D), sur la base "attributs".

La figure 5.15(a) montre que les dix premiers axes principaux permettent d'expliquer 98% de la variance des données (dont 65% pour le premier axe et 17% pour le second) pour la base "attributs". Il est donc possible de passer de 32 à 10 dimensions, en perdant moins de 2% de l'information originale. De plus, en représentant les corrélations entre les attributs originaux grâce à un cercle unité (cf. figure 5.15(b)), nous pouvons remarquer qu'un nombre important de ces caractéristiques sont très fortement corrélées (notamment les valeurs des hauteurs des signaux, ainsi que les nombres de pics).

Les résultats présentés sur les figures 5.15(c) et 5.15(d) illustrent la difficulté de discrimination des cellules en utilisant les attributs extraits. En effet, l'ACP et le LPP montrent qu'il existe des chevauchements entre plusieurs groupes d'individus : par exemple, pour les classes jaune (*Emiliania huxleyi*, représentée par des cercles) et rouge (*Phaeocystis globosa*, représentée par des étoiles) dans l'espace de projection obtenu par LPP, ou encore les classes rose (*Skeletonema*

costatum, représentée par des carrés) et bleu foncé (*Chaetoceros socialis*, représentée par des croix) dans l'espace de projection obtenue par ACP.

Les résultats obtenus par LPP montrent également, qu'il existe des sous-groupes de cellules appartenant à une même espèce (cf. figure 5.15(d)). La classe jaune (*Emiliania huxleyi*, représentée par des cercles) est, par exemple, représentée ici par quatre groupes distincts et relativement distants dans l'espace de projection construit.

5.4.2 Classification supervisée

Dans un contexte supervisé, nous utilisons la connaissance d'appartenance des cellules aux différentes espèces, afin de définir des domaines pour chaque classe qui permettraient de classifier des cellules non étiquetées. Pour cela, nous divisons la base originale selon la technique de validation croisée, en quatre sous-ensembles de 25×7 cellules chacun, qui jouent tour à tour le rôle de base d'apprentissage ou de prototypes. Afin d'évaluer l'intérêt de la classification grâce à l'extraction d'attributs, nous testons trois méthodes de classification :

- **un perceptron multi-couches** (noté MLP), avec une couche cachée et utilisant une fonction de transfert de type sigmoïde. La répartition des neurones pour la base "attributs" est la suivante : 32/19/7 (entrées/couche cachée/sorties).
- **l'algorithme du plus proche voisin** (noté 1-PPV).
- **une machine à vecteurs supports** (noté SVM), avec un noyau polynomial d'ordre 1.

Méthodes	Base 1	Base 2	Base 3	Base 4	Moyenne
MLP	**0.986**	**0.973**	**0.983**	**0.977**	**0.979**
1-PPV	0.970	0.954	0.958	0.960	0.960
SVM	0.952	0.941	0.947	0.962	0.950

TABLE 5.1 – Indices de Rand obtenus par classification supervisée et validation croisée, sur la base "attributs".

Méthodes	Base 1	Base 2	Base 3	Base 4	Moyenne
MLP	**0.973**	**0.951**	**0.962**	**0.955**	**0.960**
1-PPV	0.939	0.908	0.939	0.929	0.928
SVM	0.909	0.875	0.895	0.925	0.901

TABLE 5.2 – F-mesures obtenus par classification supervisée et validation croisée, sur la base "attributs".

Nous constatons, grâce au tableau 5.1, que les méthodes de classification supervisée sur la base "attributs" permettent d'obtenir des résultats moyens très satisfaisants. En particulier, le

perceptron multi-couches à une couche cachée offre des scores très élevés (indice de Rand égal à 0.979 et F-mesure égale à 0.960, ce qui représente 92.7% de bonne reconnaissance). Ceci implique que l'information contenue dans les attributs extraits des signaux est suffisamment discriminante dans un cadre supervisé. Ce travail préliminaire de validation croisée des résultats permet donc de montrer la pertinence des algorithmes utilisés au même titre que la méthode d'extraction de l'information.

Néanmoins, il est nécessaire de rappeler que cette classification a été réalisée pour un échantillon de culture ce qui implique que la base de données est relativement "propre" (présence de cellules vivantes présentant une morphologie quasi-parfaite et absence de bruit et de cellules dégradées).

5.4.3 Classification semi-supervisée avec contraintes

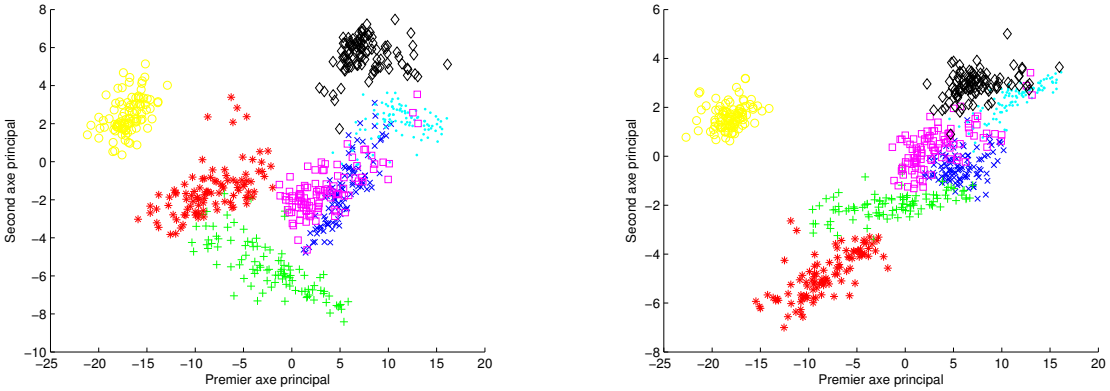
Comme présenté dans la section 5.2.3, il existe une forte variabilité des profils cytométriques intra et inter-espèces, engendrant des difficultés de discrimination. Il paraît donc justifié d'apporter de l'information *a priori* sous forme de contraintes de comparaison par paires d'objets. Ces dernières sont générées selon le processus décrit dans la section 4.3.3 du chapitre 4 (ici, la totalité des contraintes générées est utilisée).

Visualisation des données par projection sous contraintes

La figure 5.16 montre une visualisation en deux dimensions de l'ACP contrainte (2% et 5% de contraintes), pour la base "attributs" (cf. figures 5.16(a) et 5.16(b)). Nous pouvons donc affirmer que, dans l'espace de projection construit à partir des deux premières composantes principales, et pour la base testée, l'ajout de contraintes de comparaison par paires d'objets permet une discrimination plus aisée des espèces phytoplanctoniques. En effet, les classes bleue (*Chaetoceros socialis*, représentée par des triangles) et rose (*Skeletonema costatum*, représentée par des carrés) se chevauchent lorsqu'aucune contrainte n'est introduite. Ici, le chevauchement est moins important (notamment avec 5% d'étiquettes connues). De plus, cet espace de projection est capable de prendre en considération à la fois, la structure originale des données mais également les connaissances *a priori* fournies par les experts.

Résultats obtenus par les algorithmes semi-supervisés

La figure 5.17 montre les résultats obtenus par les algorithmes **SL-$\overline{L}_2$**, **FCSC-θSP** et **SSSC** présentés au chapitre 4, en termes d'indice de Rand, de valeurs de coupe *MNCut*, mais égale-



(a) ACP 2% - Visualisation 2D (b) ACP 5% - Visualisation 2D

FIGURE 5.16 – Résultats obtenus par ACP pour 2 et 5% de contraintes (visualisation 2D), sur la base "attributs".

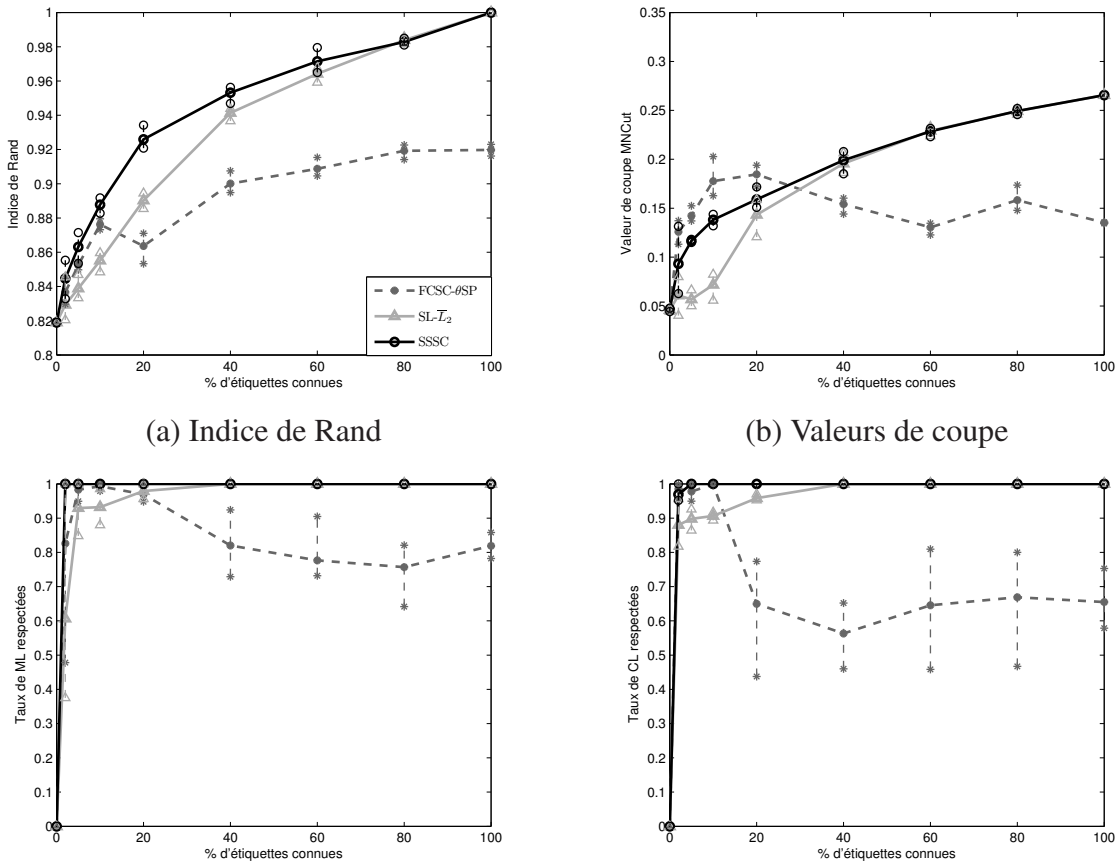
ment de pourcentages de contraintes des deux types respectées :

- Pour un faible nombre d'étiquettes connues, la méthode proposée et appliquée sur la base "attributs", permet d'obtenir une amélioration nette des performances de partitionnement (pour 5% d'étiquettes connues, l'indice de Rand est amélioré de 0.044, et la F-mesure, de 0.019). De plus, les taux de respect des contraintes des deux types sont très élevés, contrairement aux algorithmes **SL-$\overline{L}_2$** et **FCSC-θSP**.

- Plus le pourcentage d'étiquettes connues augmente, plus les valeurs de coupe *MNCut* augmentent. A partir de 20% d'étiquettes connues, l'algorithme **FCSC-θSP** obtient les valeurs de coupe les plus faibles. Néanmoins, ses pourcentages de contraintes respectées, de type "Must-Link" et "Cannot-Link", sont également faibles alors que pour **SSSC**, ces taux sont très élevés. La méthode proposée permet donc d'obtenir, à la fois des taux élevés en respect des contraintes ainsi que des valeurs satisfaisantes de coupe *MNCut* (inférieures à 0.3).

Le tableau 5.3 présente quelques indicateurs de performance pour la méthode d'extraction de l'information utilisée, pour la base de données composée de cellules phytoplanctoniques issues d'un échantillon de culture. L'étiquetage des cellules étant une tâche complexe et coûteuse en temps, il est indispensable que l'algorithme proposé soit performant avec une faible quantité d'informations *a priori*. Pour cette raison, nous choisissons de présenter les indicateurs de performance pour 2% et 5% d'étiquettes connues.

Nous pouvons noter une augmentation permanente des résultats obtenus par la méthode proposée, fonction du pourcentage d'étiquettes connues considéré. L'algorithme permet également



(a) Indice de Rand

(b) Valeurs de coupe

(c) Taux de *ML* respectées

(d) Taux de *CL* respectées

FIGURE 5.17 – Indices de Rand, valeurs de coupe et taux de contraintes "Must-Link" et "Cannot-Link" respectées, en fonction du pourcentage d'étiquettes connues, sur la base "attributs".

d'obtenir, dans tous les cas, les indices de Rand et les totaux de contraintes respectées les plus élevés. Pour 5% d'étiquettes connues, il est possible de constater que la totalité des contraintes des deux types ("Must-Link" et "Cannot-Link") est respectée pour **SSSC** (100.0%). La valeur de coupe *MNCut* obtenue est alors plus élevée que pour **SL-$\overline{L}_2$** (0.117 contre 0.057), mais plus faible que pour **FCSC-θSP** (0.117 contre 0.142).

5.5 Expérimentations sur la base "similarités"

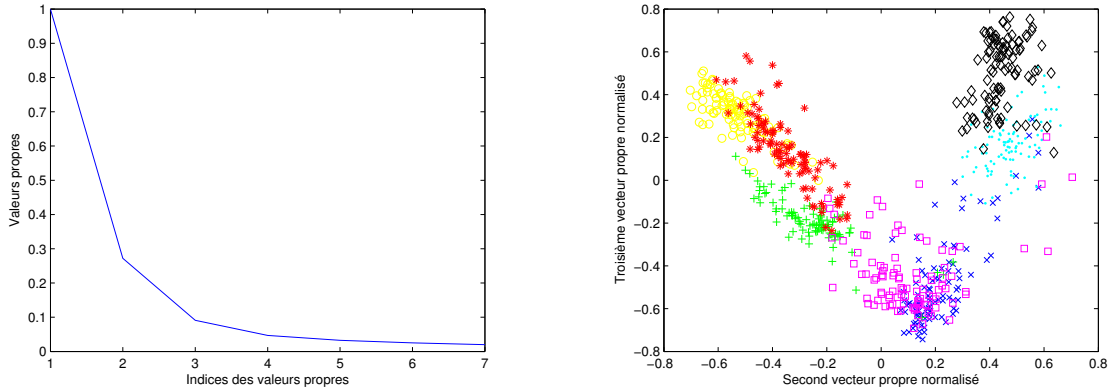
5.5.1 Visualisation par projection dans l'espace spectral non contraint

L'algorithme de classification spectrale proposée par Ng et al. [Ng et al., 2002], est appliqué sur la base résultant des appariements élastiques. La figure 5.18 fournit alors une visualisation en deux dimensions des résultats obtenus par projection dans l'espace spectral construit à partir

% d'étiquettes connues	Méthodes	*% ML*	*% CL*	% Total	J_{MNCut}	Indice de Rand	F-mesure
0	SL-$\overline{L}_2$	/	/	/	**0.046**	**0.819**	**0.790**
	FCSC-θSP	/	/	/	**0.046**	**0.819**	**0.790**
	SSSC	/	/	/	**0.046**	**0.819**	**0.790**
2	SL-$\overline{L}_2$	60.6	88.0	74.3	**0.061**	0.829	0.792
	FCSC-θSP	82.6	**98.5**	90.5	0.126	0.837	0.793
	SSSC	**100.0**	96.9	**98.8**	0.093	**0.845**	**0.796**
5	SL-$\overline{L}_2$	92.9	89.8	91.3	**0.057**	0.839	0.793
	FCSC-θSP	98.3	97.8	98.0	0.142	0.852	0.804
	SSSC	**100.0**	**100.0**	**100.0**	0.117	**0.863**	**0.809**

TABLE 5.3 – Scores de performances sur la base de cellules issues d'un échantillon de culture ($K = 7$) avec différents pourcentages d'étiquettes connues.

des vecteurs propres de la matrice Laplacienne $\overline{L}_2$.



(a) Valeurs propres triées (b) Espace spectral

FIGURE 5.18 – Projections des points sur la sphère unité, grâce aux différents vecteurs propres de la matrice Laplacienne normalisée symétrique (second et troisième vecteurs propres).

La figure 5.18(a) montre les K plus grandes valeurs propres de la matrice Laplacienne, associées aux K vecteurs propres permettant de construire l'espace spectral. La figure 5.18(b), quant à elle, permet d'obtenir une visualisation plane des différentes classes par projection des objets sur le second et le troisième vecteur propre. Cette figure permet d'illustrer la difficulté de discrimination des cellules en utilisant les appariements élastiques des signaux. En effet, des chevauchements existent entre différentes espèces, notamment entre la classe bleue (*Chaetoceros socialis*, représentée par des croix) et rose (*Skeletonema costatum*, représentée par des carrés). Ceci s'explique par la similarité importante entre les signaux profils caractéristiques de

ces espèces.

5.5.2 Classification supervisée

Les expérimentations réalisées dans un contexte supervisé sont basées sur le même protocole expérimental que pour la base "attributs". En effet, nous divisons la base originale selon la technique de validation croisée, en quatre sous-ensembles de 25×7 cellules chacun, qui jouent tour à tour le rôle de base d'apprentissage ou de prototypes. Les méthodes de classification supervisée testées sont alors les suivantes :

- **un perceptron multi-couches**, avec une couche cachée et utilisant une fonction de transfert de type sigmoïde. La répartition des neurones pour la base "similarités" est la suivante : 175/91/7 (entrées/couche cachée/sorties).
- **l'algorithme du plus proche voisin**.
- **une machine à vecteurs supports**, avec un noyau polynomial d'ordre 1.

L'idée originale de la méthode [Caillault et al., 2009] consiste donc à proposer les lignes de la matrice composée de valeurs de similarités entre cellules, en entrée de chacun des classifieurs. En effet, chacune des cellules est décrite par un vecteur similarités (de dimension $N \times 1$, avec N étant égal au nombre total d'objets), jouant le rôle de vecteur attributs.

Méthodes	Base 1	Base 2	Base 3	Base 4	Moyenne
MLP	0.991	**0.984**	**0.984**	0.981	**0.985**
1-PPV	0.991	0.972	0.979	**0.984**	0.982
SVM	**0.993**	0.976	0.976	0.979	0.981

TABLE 5.4 – Indices de Rand obtenus par classification supervisée et validation croisée, sur la base "similarités".

Méthodes	Base 1	Base 2	Base 3	Base 4	Moyenne
MLP	0.985	**0.972**	**0.972**	0.970	**0.975**
1-PPV	0.985	0.951	0.962	**0.980**	0.969
SVM	**0.988**	0.956	0.956	0.962	0.965

TABLE 5.5 – F-mesures obtenus par classification supervisée et validation croisée, sur la base "similarités".

Le tableau 5.4 présente les résultats obtenus pour la base "similarités". De la même manière que pour les expérimentations précédentes, le perceptron multi-couches permet d'obtenir les meilleurs scores de classification (indice de Rand égal à 0.985 et F-mesure égale à 0.975, ce qui représente 96.7% de bonne reconnaissance). De plus, il est possible d'observer une nette

amélioration des résultats par rapport à ceux obtenus grâce à la base constituée des attributs.

Dans le cadre supervisé, la méthode d'appariement élastique des signaux bruts apparaît plus performante que celle basée attributs. En effet, l'information semble plus complète (car les attributs caractéristiques synthétisent l'information contenue dans les signaux). Cette méthode de classification, basée sur les données brutes, est donc plus pertinente que celle basée sur les attributs caractéristiques. Ceci est confirmé par les résultats obtenus sur trois types de classifieur : MLP, 1-PPV et SVM (cf. tables 5.4 et 5.5).

5.5.3 Classification semi-supervisée avec contraintes

Dans cette section, les ensembles de contraintes de comparaison utilisés sont ceux générés dans le cas des expérimentations effectuées sur la base "attributs".

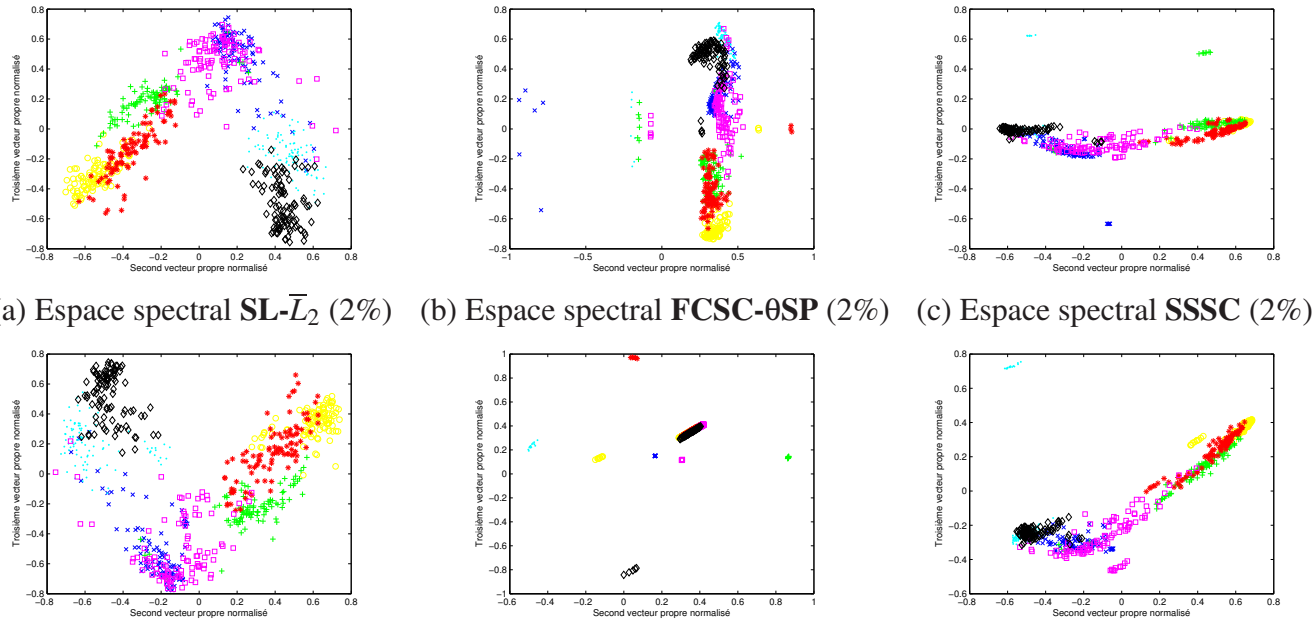
Visualisation des données par projection sous contraintes

Afin de montrer l'impact des contraintes de comparaison par paires d'objets sur la représentation des données en groupes, nous proposons une visualisation en deux dimensions de l'espace spectral contraint (2% et 5% de contraintes), pour la base "similarités" et pour les algorithmes **SL-$\overline{L}_2$**, **FCSC-θSP** et **SSSC** (cf. figure 5.19). Visuellement, il est aisé de constater que les espaces de projection obtenus par les algorithmes **FCSC-θSP** (cf. figures 5.19(b) et (e)) et **SSSC** (cf. figures 5.19(c) et (f)) permettent de respecter la structure en groupes des données ainsi que les contraintes de comparaison par paires d'objets (contrairement à ceux obtenus par l'algorithme **SL-$\overline{L}_2$**, représentés sur les figures 5.19(a) et (d)). Pour ces méthodes, les points les plus éloignés représentent les cellules contraintes. Cependant, dans ce plan, la discrimination des groupes de cellules non contraintes par la méthode **FCSC-θSP** (avec 5% d'étiquettes connues) apparaît plus difficile que pour les autres algorithmes (les données sont compactées dans l'intervalle [0.2-0.4 ;0.3-0.5]).

Résultats obtenus par les algorithmes semi-supervisés

La figure 5.20 montre les résultats obtenus par les algorithmes **SL-$\overline{L}_2$**, **FCSC-θSP** et **SSSC**. Les scores de performance sont présentés en termes d'indices de Rand, de valeurs de coupe *MNCut*, mais également de proportions de contraintes des deux types respectées :

– Pour un faible nombre d'étiquettes connues, la méthode proposée et appliquée sur la base "similarités", permet d'obtenir une faible amélioration des performances de groupement. Ceci s'explique principalement par le score très élevé de la classification spectrale non



(a) Espace spectral **SL-**$\overline{L}_2$ (2%) (b) Espace spectral **FCSC-θSP** (2%) (c) Espace spectral **SSSC** (2%)

(d) Espace spectral **SL-**$\overline{L}_2$ (5%) (e) Espace spectral **FCSC-θSP** (5%) (f) Espace spectral **SSSC** (5%)

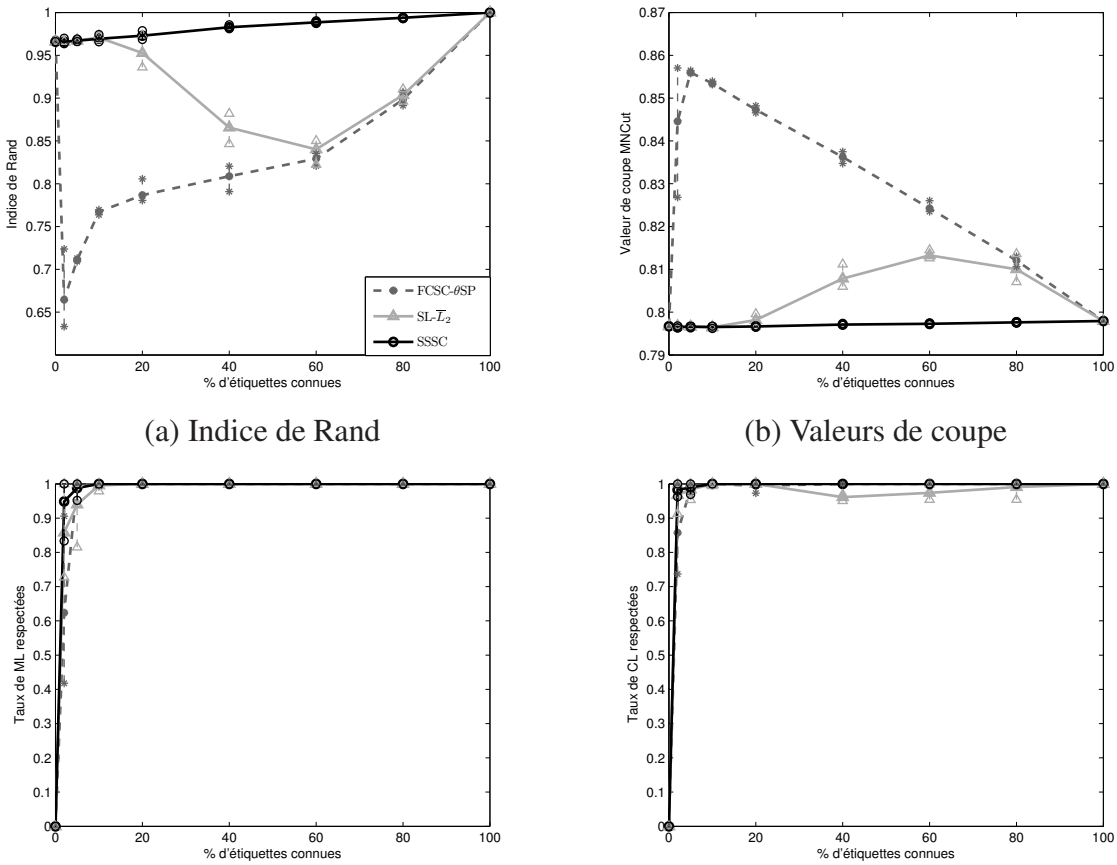
FIGURE 5.19 – Projections des points sur la sphère unité, grâce aux différents vecteurs propres de la matrice Laplacienne normalisée symétrique, sur la base "similarités" (2% et 5% de contraintes).

supervisée (correspondant à l'abscisse 0%), mais également par l'aspect aléatoire du processus de génération des contraintes, pouvant engendrer un apport d'informations non utiles. Cependant, l'algorithme proposé permet d'obtenir une partition plus proche de celle optimale que ses concurrentes.

– A partir de 10% d'étiquettes connues, nous remarquons que les taux de respect des contraintes de comparaison des deux types restent très élevés (contrairement à **SL-**$\overline{L}_2$ pour laquelle le taux de contraintes "Cannot-Link" respectées diminue). De plus, **SSSC** permet d'obtenir des valeurs de coupe *MNCut* très satisfaisantes. La méthode d'extraction de l'information, basée sur les données brutes, ainsi que l'algorithme de classification semi-supervisée proposé, permettent donc d'obtenir des scores de performance élevés, conduisant à la partition désirée.

Le tableau 5.6 présente quelques indicateurs de performance pour la méthode d'extraction de l'information par appariement élastique, pour la base de données composée de cellules phytoplanctoniques issues d'un échantillon de culture.

Pour l'algorithme **SSSC**, comme démontré grâce aux figures 5.20(a) à (d), les indices de Rand obtenus sur la base "similarités" sont meilleurs que ceux obtenus sur la base "attributs"



(a) Indice de Rand

(b) Valeurs de coupe

(c) Taux de *ML* respectées

(d) Taux de *CL* respectées

FIGURE 5.20 – Indices de Rand, valeurs de coupe et taux de contraintes "Must-Link" et "Cannot-Link" respectées, en fonction du pourcentage d'étiquettes connues, sur la base de culture.

(lorsqu'aucune étiquette de classe n'est connue, l'indice de Rand obtenu sur cette dernière est égal à 0.819 contre 0.966). A partir de 5% d'étiquettes connues, une faible augmentation des scores de performance est observée. En effet, la valeur de l'indice de Rand mais également celle de F-mesure sont améliorées (pour l'indice de Rand : +0.002, et pour la F-mesure : +0.006). De plus, pour ce même pourcentage, bien que le total de contraintes respectées soit plus élevé pour **FCSC-θSP**, la valeur de coupe ainsi que l'indice de Rand et la F-mesure restent meilleurs pour la méthode proposée.

Le tableau 5.7 montre que la quantité de cellules non reconnues par notre méthode, mais reconnues par l'algorithme **SL-$\overline{L}_2$** est sensiblement la même que pour l'inverse. Nous décidons de visualiser les partitions obtenues par les différents algorithmes (avec 2% d'étiquettes connues) afin d'identifier ces cellules.

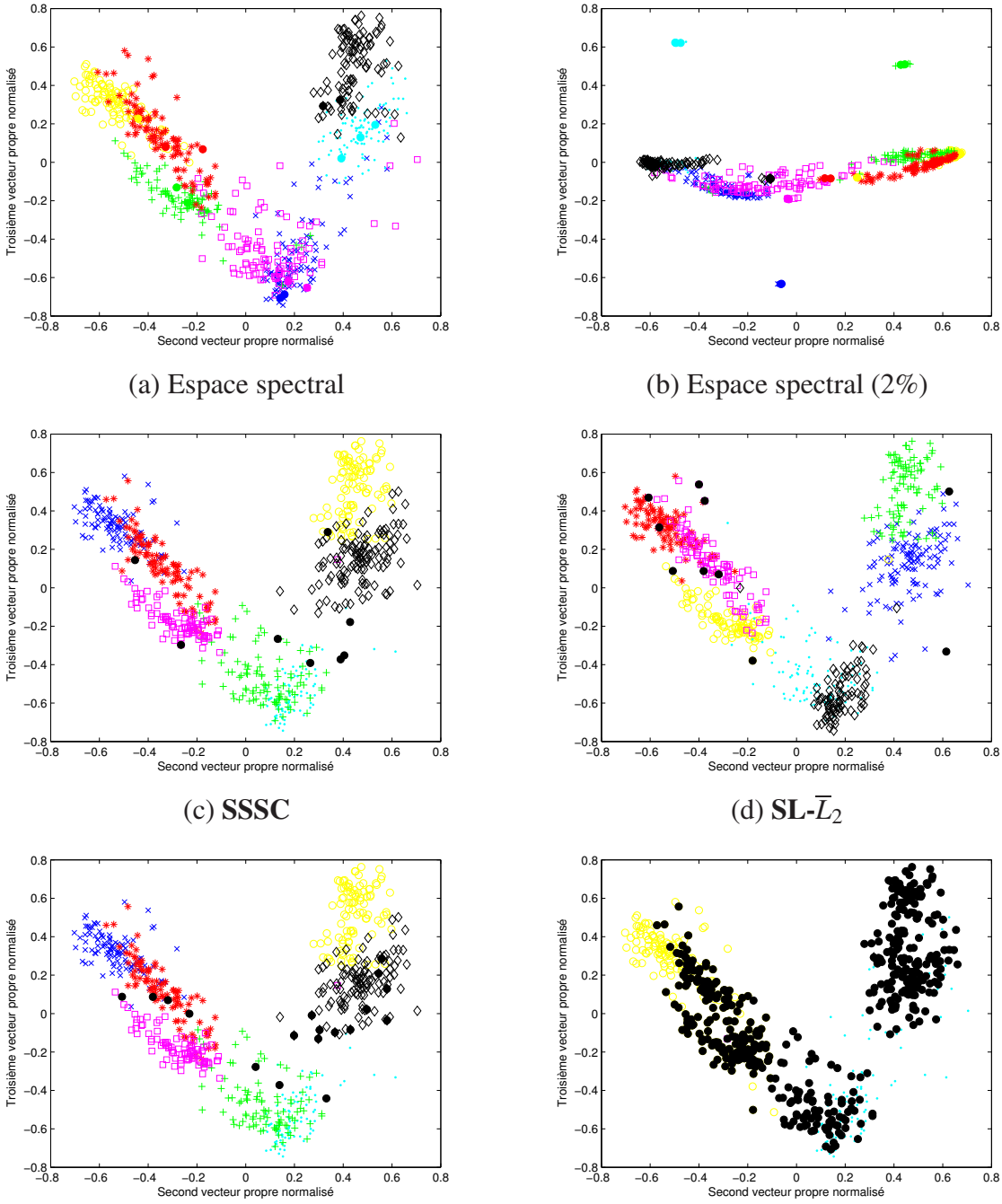
% d'étiquettes connues	Méthodes	% *ML*	% *CL*	% Total	J_{MNCut}	Indice de Rand	F-mesure
0	SL-$\overline{L}_2$	/	/	/	**0.797**	**0.966**	**0.924**
	FCSC-θSP	/	/	/	**0.797**	**0.966**	**0.924**
	SSSC	/	/	/	**0.797**	**0.966**	**0.924**
2	SL-$\overline{L}_2$	85.8	96.7	91.3	**0.797**	**0.966**	**0.928**
	FCSC-θSP	62.4	85.7	74.1	0.845	0.665	0.676
	SSSC	**94.9**	**98.2**	**96.6**	**0.797**	**0.966**	**0.928**
5	SL-$\overline{L}_2$	94.0	98.4	96.2	**0.797**	0.967	0.929
	FCSC-θSP	**100.0**	98.8	**99.4**	0.856	0.711	0.734
	SSSC	98.8	**99.0**	98.9	**0.797**	**0.968**	**0.930**

TABLE 5.6 – Scores de performances sur la base de cellules issues d'un échantillon de culture ($K = 7$) avec différents pourcentages d'étiquettes connues.

2%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	92.3%	1.4%	27.4%	66.3%
	Non Reconnus	1.2%	5.1%	2.6%	3.7%

TABLE 5.7 – Comparaison de l'algorithme **SSSC** avec **SL-$\overline{L}_2$** et **FCSC-θSP**, pour la base provenant d'un échantillon de culture (2% d'étiquettes connues).

La partition représentée sur les figures 5.21(a) et (b) est la "vérité terrain". Sur ces figures, les objets contraints sont représentés par des ronds pleins (colorés selon le groupe d'appartenance). Les ronds noirs pleins, sur la figure 5.21(c), sont les objets non reconnus par **SSSC** et reconnus par **SL-$\overline{L}_2$** (inversement pour la figure (d)). De même, les ronds noirs pleins sur la figure (e), sont les objets non reconnus par **SSSC** et reconnus par **FCSC-θSP** (inversement pour la figure (f)). Comme montré sur les figures 5.21(c) et (d), les cellules non reconnues par une méthode mais reconnues par l'autre, ne sont pas les mêmes. De plus, la majorité de ces erreurs concernent des objets litigieux (points isolés ou distants des classes d'appartenance). La figure 5.21(f) montrent que l'algorithme proposé permet de reconnaître un nombre important de cellules non reconnues par la méthode **FCSC-θSP**. Ceci s'explique par le fait que cette dernière tend à isoler les cellules contraintes (groupes ne contenant qu'une cellule). La plupart des classes réelles sont alors fusionnées.



(a) Espace spectral

(b) Espace spectral (2%)

(c) **SSSC**

(d) **SL-$\overline{L}_2$**

(e) **SSSC**

(f) **FCSC-θSP**

FIGURE 5.21 – Visualisations planes des partitions des objets.

5.6 Conclusion

Nous avons appliqué et comparé différentes méthodes de visualisation et de classification supervisée et semi-supervisée, afin d'identifier des cellules phytoplanctoniques. La base de don-

nées à disposition est composée de cellules provenant d'un échantillon de culture, réalisée en laboratoire (LOG). L'étiquetage est rendu possible grâce à une visualisation plane issue du logiciel associé au cytométre en flux.

Il est d'usage, dans la cytométrie en flux, d'extraire l'information contenue dans les signaux cytométriques de chacune des cellules, afin de pouvoir comparer ces dernières entre elles. La méthode "classique" consiste en l'extraction d'attributs caractéristiques (tels que la longueur, la hauteur, l'intégrale et le nombre de pics). Nous avons proposé une alternative s'appuyant sur la méthode d'appariement élastique de séquences temporelles (DTW) proposée par Sakoe et Chiba [Sakoe and Chiba, 1978]. L'algorithme proposé permet un appariement élastique conjoint des courbes cytométriques en tolérant des déformations temporelles locales. La mesure obtenue reflète alors la similarité entre profils de cellules.

Les différents classifieurs utilisés dans ce chapitre (et proposés dans les chapitres 3 et 4), sont évalués dans un contexte supervisé (grâce à une validation croisée des résultats), mais également dans un contexte semi-supervisé (pour lequel les contraintes de comparaison sont définies grâce aux étiquettes de classes fournies par l'expert).

Les scores de performance obtenus permettent donc de mettre en avant l'intérêt des algorithmes. Cependant, la base de données utilisée est une base qualifiée de "propre" car elle ne contient que des cellules présentant une morphologie quasi-parfaite et non dégradée. C'est pourquoi, il est intéressant de tester ces méthodes sur une base de cellules provenant d'un échantillon du milieu naturel et contenant des particules à des stades de vie différents. Les résultats obtenus par classification non supervisée et semi-supervisée sont présentés dans le chapitre suivant.

Chapitre 6

Recherche de groupes de cellules phytoplanctoniques dans un échantillon marin

6.1 Introduction

Dans l'environnement marin, les variations climatiques récurrentes peuvent avoir des répercussions sur l'état physiologique des cellules qui se traduit par une modification de leur morphologie, voire une dégradation de leur contenu pigmentaire. Dans ce sens, les profils des cellules, obtenus suite à l'analyse cytométrique d'échantillons marins, montrent une plus grande variabilité intra-espèces et inter-espèces, par rapport aux cellules provenant d'un échantillon de culture. La difficulté d'analyse d'échantillons marins réside dans le fait que les signaux sont très bruités et le nombre des espèces est inconnu.

Dans ce chapitre, nous proposons une démarche méthodologique pour la recherche automatique de groupes de cellules dans un échantillon marin. Nous partons de l'analyse exploratoire de l'expert biologiste. Nous présentons ensuite des techniques classiques d'analyse automatique des données. Celles-ci sont confrontées aux techniques récentes de classification et de visualisation par approches spectrales. Pour améliorer les résultats d'analyse, nous introduisons des connaissances supplémentaires de comparaison de profils, faciles à générer.

Après avoir décrit la base de données expérimentale et la démarche experte d'analyse, nous caractérisons les données par différentes techniques de visualisation pour, ensuite, traiter le problème de recherche du nombre d'espèces. Nous comparons les résultats de classification

non supervisée classique et récente sur la base de données ”attributs” et sur la base ”similarités”. Aussi, nous proposons d'améliorer les performances des classifieurs par ajout de connaissances type contraintes de comparaison.

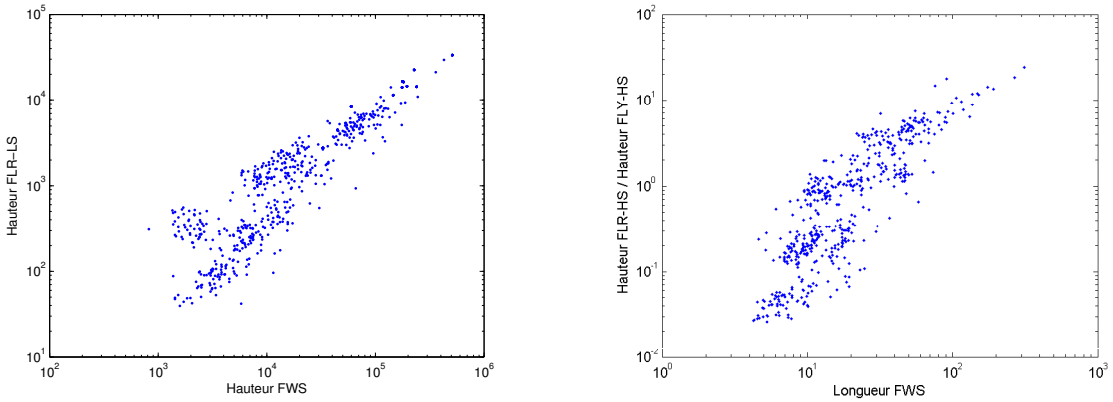
6.2 Démarche experte d'analyse des données

Les biologistes du laboratoire LOG de la Maison de la Recherche en Environnement Naturel (MREN), ont menés des campagnes de prélévement d'échantillons marins en Manche orientale. Ces campagnes ont été effectuées à une fréquence hebdomadaire et essentiellement en période printanière, période productive du phytoplancton. Leur objectif est alors de réaliser une analyse descriptive des communautés phytoplanctoniques, aussi bien en termes d'abondance que de biomasse et de composition, pour la surveillance du milieu marin.

Le cytomètre en flux utilisé pour les expérimentations permet de régler un seuil de détection (”trigger level”) pour les différents signaux cytométriques. En fixant une valeur de seuil élevée pour le signal de fluorescence rouge (caractéristique de la présence de chlorophylle dans la cellule), il est possible d'éliminer les déchets présents dans l'échantillon d'eau. Cependant, ceci peut engendrer le risque d'éliminer des cellules dégradées ou vieilles (peu fluorescentes).

Le logiciel ”CytoClus©” caractérise chaque profil de cellule par un ensemble de 32 attributs. L'expert biologiste considère, par expérience, que certains attributs sont plus pertinents que d'autres. Il visualise sur quelques plans, définis par les attributs retenus, l'ensemble des cellules représentant l'échantillon (par exemple, le plan représenté sur la figure 6.1(a) est défini par la hauteur du signal de diffusion à petits angles FWS et la hauteur du signal de fluorescence rouge FLR-LS). Il repère les groupes denses et les délimite manuellement. Cette classification visuelle interactive nécessite un savoir faire et une connaissance plus ou moins approximative de la position relative des classes, voire de leurs domaines respectifs. Pour les cellules jugées prototypes de classes, il examine la forme des profils pour en déduire le type d'espèce. Ensuite, le logiciel donne le nombre de cellules dans le domaine délimité par l'expert. Il est évident que cette démarche nécessite une certaine dextérité et une grande connaissance du domaine, notamment, les profils des espèces qui sont censées être présentes dans la base. Aussi, l'étiquetage étant réalisé par extrapolation, il présente des imprécisions.

Une analyse fine par l'expert a permis de mettre en évidence des espèces pouvant se trouver sous forme de cellules seules ou de colonies comme le montre le tableau 6.1.



(a) Plan 1 (b) Plan 2

FIGURE 6.1 – Visualisations planes de l'ensemble des cellules (600 cellules) composant l'échantillon issu du milieu naturel.

Espèces	Cellules seules	Colonies	Inconnu
Asterionella glacialis	X	X	
Lauderia annulata	X	X	
Skeletonema costatum	X	X	
Thalassiosira rotula	X	X	
Guinardia delicatula		X	
Cylindrothetca closterium	X		
Cryptophycées			X
Dinoflagellés			X

TABLE 6.1 – États des espèces présentes dans la base de données naturelles.

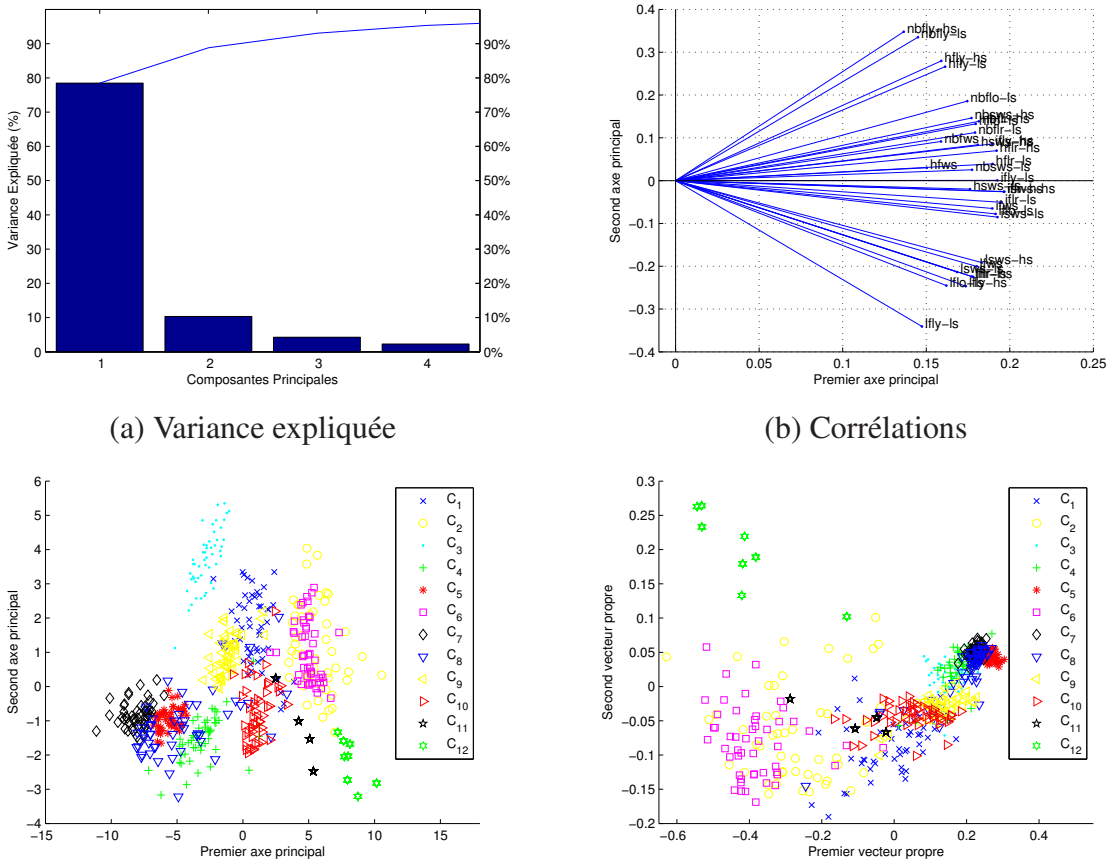
Au total, 12 groupes ont donc été identifiés. Nous considérons les résultats de l'analyse experte comme "vérité-terrain" malgré son manque de précision. Dans la suite, nous procédons à des visualisations planes des données pour la base "attributs".

6.3 Expérimentations sur la base "attributs"

Nous procédons, tout d'abord, par une visualisation plane des données en considérant le nombre de groupes $K = 12$ estimé par les biologistes. Ensuite, nous appliquons des méthodes de partitionnement automatique en choisissant différentes valeurs de K (déterminées automatiquement). Cette même démarche est employée en ajoutant un faible pourcentage (2% et 4%) de contraintes et en fixant le nombre de groupes K à 12. Pour l'ensemble des expérimentations, les résultats sont comparés avec la "vérité terrain".

6.3.1 Visualisations planes par ACP et LPP

Les techniques d'analyse en composantes principales et de projection préservant la structure locale (LPP) sont appliquées sur la base "attributs", composée des logarithmes des valeurs originales (cf. figure 6.2). Pour le cas LPP, la matrice de similarités est construite à partir d'un noyau gaussien, pour lequel le paramètre de dispersion est calculé de manière locale (avec un nombre de voisins égal à 7) [Zelnik-Manor and Perona, 2004]. Afin de mettre en évidence la difficulté de discrimination, nous représentons sur ces figures, l'étiquetage "vérité-terrain" fourni par les biologistes.



(a) Variance expliquée

(b) Corrélations

(c) ACP - Visualisation 2D

(d) LPP - Visualisation 2D

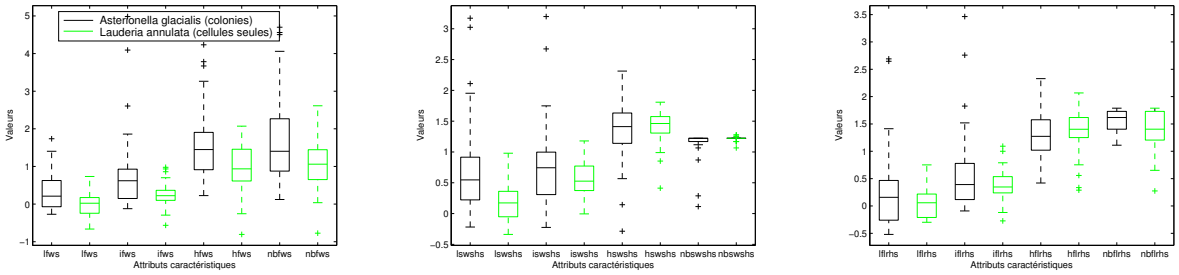
FIGURE 6.2 – Résultats obtenus par ACP et LPP (visualisation 2D), sur la base "attributs".

La figure 6.2(a) montre la variance expliquée. Les deux premiers axes expliquent 78% et 10% de la variance, respectivement. Les dix premiers axes principaux expliquent 99% de la variance des données. Il est donc possible de passer de 32 à 10 dimensions, en ne perdant qu'un

pourcent de l'information originale.

La figure 6.2(b) montre le cercle de corrélations. Nous pouvons alors remarquer que certains attributs sont corrélés (en particulier, les hauteurs et les nombres de pics des signaux).

Les résultats présentés sur les figures 6.2(c) et 6.2(d) illustrent la difficulté de discrimination des cellules en utilisant les attributs extraits. En effet, dans un contexte non supervisé, il est très difficile de repérer des groupes dans les différents espaces de projection obtenus par les méthodes testées (à l'exception de la classe composée des points bleus et représentant l'espèce appartenant au groupe taxonomique des Cryptophycées). De plus, il semble exister des chevauchements importants entre les différents groupes de cellules : par exemple, entre les triangles bleus (*Skeletonema costatum* en colonies, C_8), les losanges noirs (*Skeletonema costatum* en cellules seules, C_7) et les croix rouges (l'espèce appartenant aux Dinoflagellés, C_5), mais également entre les cercles jaunes (*Asterionella glacialis* en colonies, C_2) et les carrés roses (*Lauderia annulata* en cellules seules, C_6).



(a) Signal FWS (b) Signal SWS-HS (c) Signal FLR-HS

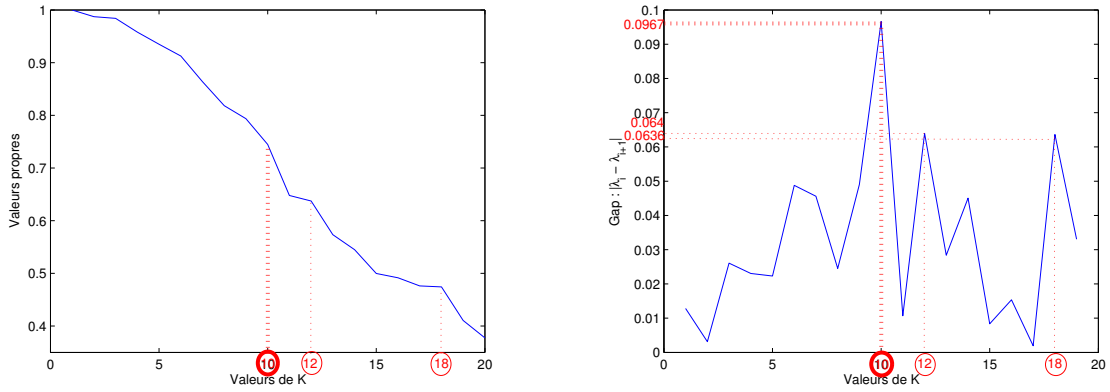
FIGURE 6.3 – Illustration de la variabilité entre les espèces *Asterionella glacialis* (colonies) et *Lauderia annulata* (cellules seules), grâce aux attributs caractéristiques.

Nous choisissons de prendre pour exemples, les espèces *Asterionella glacialis* (colonies) et *Lauderia annulata* (cellules seules). Les boîtes à moustaches ("boxplots") associées aux attributs et illustrées sur la figure 6.3 (pour les signaux FWS, SWS-HS et FLR-HS), reflétent le fort chevauchement entre espèces. Ceci permet alors d'expliquer la difficulté de discrimination de celles-ci dans l'espace de projection de l'ACP (cf. figure 6.2(c)).

6.3.2 Estimation automatique du nombre de groupes

Les données à disposition étant issues d'un échantillon du milieu naturel, le nombre d'espèces (ou groupes) n'est pas connu *a priori*. Nous avons présenté dans le chapitre 2 plusieurs

méthodes d'estimation automatique du nombre K. Nous choisissons d'appliquer les deux les plus connues : le saut de gap (Gap=$|\lambda_k - \lambda_{k+1}|$) proposée par [Shortreed and Meila, 2005] et la fonction de modularité Ψ proposée par White et Smyth [White and Smyth, 2005], sur la matrice Laplacienne L_{Ng} correspondant aux dix premiers axes principaux issus de l'ACP.



(a) Valeurs propres triées (b) Calcul du gap

FIGURE 6.4 – Résultats obtenus par le calcul du gap, pour l'estimation du nombre de groupes recherchés K, sur la base "attributs".

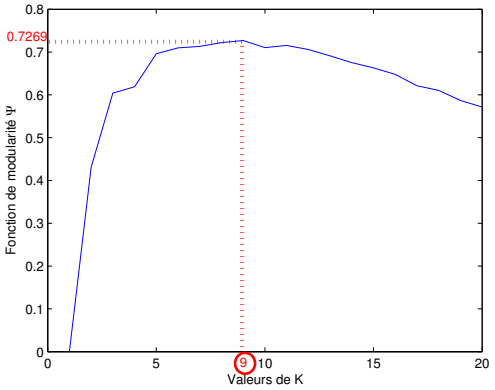


FIGURE 6.5 – Résultats obtenus par la fonction modularité, pour l'estimation du nombre de groupes recherchés K, sur la base "attributs".

La figure 6.4 présente les résultats obtenus par la méthode de calcul du gap pour l'estimation automatique du nombre de groupes K. Nous avons décidé de chercher la valeur du nombre K dans l'intervalle $[0,20]$. En effet, au-delà de 20, les valeurs propres ne sont pas significatives ($\lambda < 0.4$). La figure 6.4(a) montre les valeurs propres. Elle permet de rechercher un "saut" dans ces valeurs propres triées par ordre décroissant et montre ce saut pour la valeur de $K = 10$. La

figure 6.4(b) montre les résultats de calcul du gap et le maximum obtenu à $K = 10$. Cependant, sur cette figure, deux autres valeurs peuvent être mises en évidence : $K = 12$ et $K = 18$.

La fonction de modularité Ψ montre, quant à elle, un maximum à $K = 9$ (cf. figure 6.5). Cette valeur est plus faible que celle obtenue par la méthode de calcul du gap.

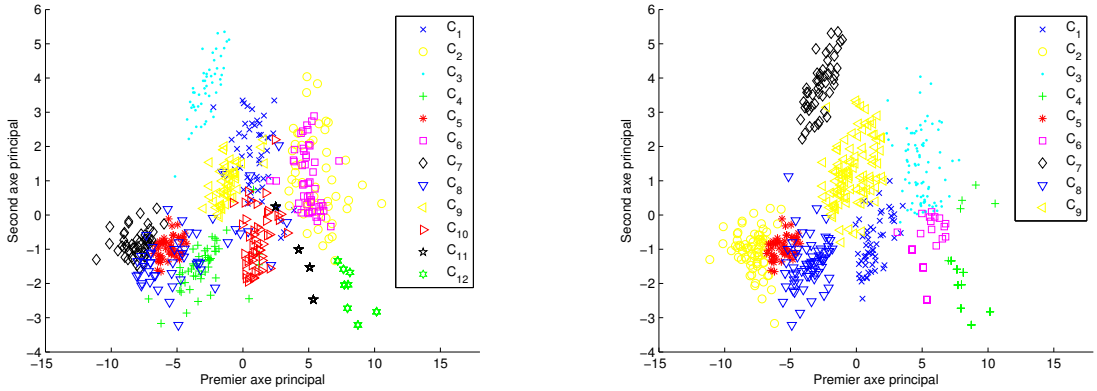
Dans la suite, nous étudions les résultats de partitionnement automatique en considérant les cas $K = 9$ et $K = 10$. Nous examinons aussi les résultats de partitionnement pour le cas $K = 12$ obtenu par méthode d'exploration experte des biologistes, et pour le cas $K = 18$ afin de rechercher la présence éventuelle de groupes fonctionnels représentant une même espèce (appelés sous-groupes).

6.3.3 Classification non supervisée

Nous appliquons des méthodes de classification non-supervisée, dans le but de montrer la difficulté de discrimination linéaire des espèces phytoplanctoniques, sur la base composée des attributs extraits des courbes cytométriques. Nous utilisons pour cela trois algorithmes dont deux sont classiques et un plus récent :

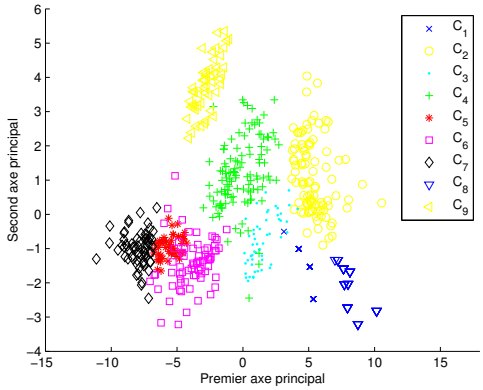
- **la classification ascendante hiérarchique** (notée CAH), utilisant une distance euclidienne, pour calculer la proximité des cellules dans l'espace d'origine. Cet algorithme s'arrête lorsque le nombre de groupes obtenu est égal à celui recherché.
- **l'algorithme des K-moyennes** (noté KM), utilisant également une distance euclidienne. Les différents centres des groupes sont initialisés de manière aléatoire, parmi les objets existants.
- **la classification spectrale** (noté SC). Le paramètre de dispersion σ est calculé pour s'adapter de manière locale [Zelnik-Manor and Perona, 2004]. L'algorithme utilisé est celui proposé par Ng et al. [Ng et al., 2002].

Les résultats obtenus par ces algorithmes sont présentés après réduction de la dimension grâce à l'application de la méthode d'analyse en composantes principales. Ceci permet alors d'éliminer les bruits et les attributs redondants et corrélés. Pour nos expérimentations, nous choisissons de ne conserver que les dix premiers axes principaux (permettant d'expliquer 99% de la variance des données), et de visualiser la partition obtenue dans le plan formé par les deux premiers axes principaux.

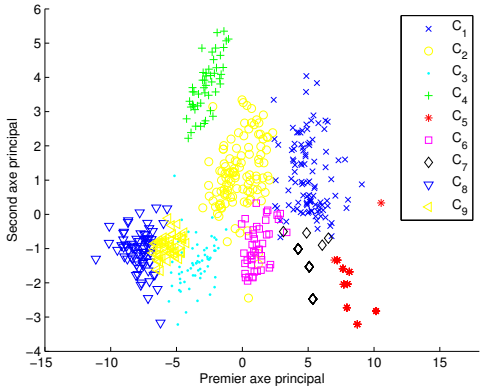


(a) Vérité terrain ($K = 9$)

(b) K-moyennes ($K = 9$)



(c) Classification hiérarchique ($K = 9$)



(d) Classification spectrale ($K = 9$)

FIGURE 6.6 – Visualisation 2D des résultats obtenus par les différents algorithmes sur la base ”attributs” ($K = 9$).

Hypothèse $K = 9$

Comme montré sur la figure 6.6, les algorithmes de classification non supervisée, testés à $K = 9$, ne pemettent pas de discriminer les classes présentant des chevauchements importants (*Asterionella glacialis* en cellules seules représentée par des croix bleus et *Thalassiosira rotula* en cellules seules représentée par des triangles jaunes, mais également *Asterionella glacialis* en colonies représentée par des cercles jaunes et *Lauderia annulata* en cellules seules représentée par des carrés roses. En effet, les résultats obtenus par les différentes méthodes (cf. figures (b) à (d)) montrent une fusion de ces groupes (triangles jaunes et points bleus pour les K-moyennes, ’+’ verts et cercles jaunes pour la classification hiérarchique, et cercles jaunes et croix bleus pour la classification spectrale).

Le tableau 6.2 présente quelques indices de performance pour les différents algorithmes à

Méthodes	Indice de Rand	F-mesure	% de reconnaissance
KM	0.795	0.608	76.0
CAH	0.812	0.624	78.5
SC	**0.815**	**0.628**	**78.8**

TABLE 6.2 – Indices de Rand, F-mesures et pourcentages de bonne reconnaissance obtenus pour les différents algorithmes de classification non supervisée, sur la base "attributs" ($K = 9$).

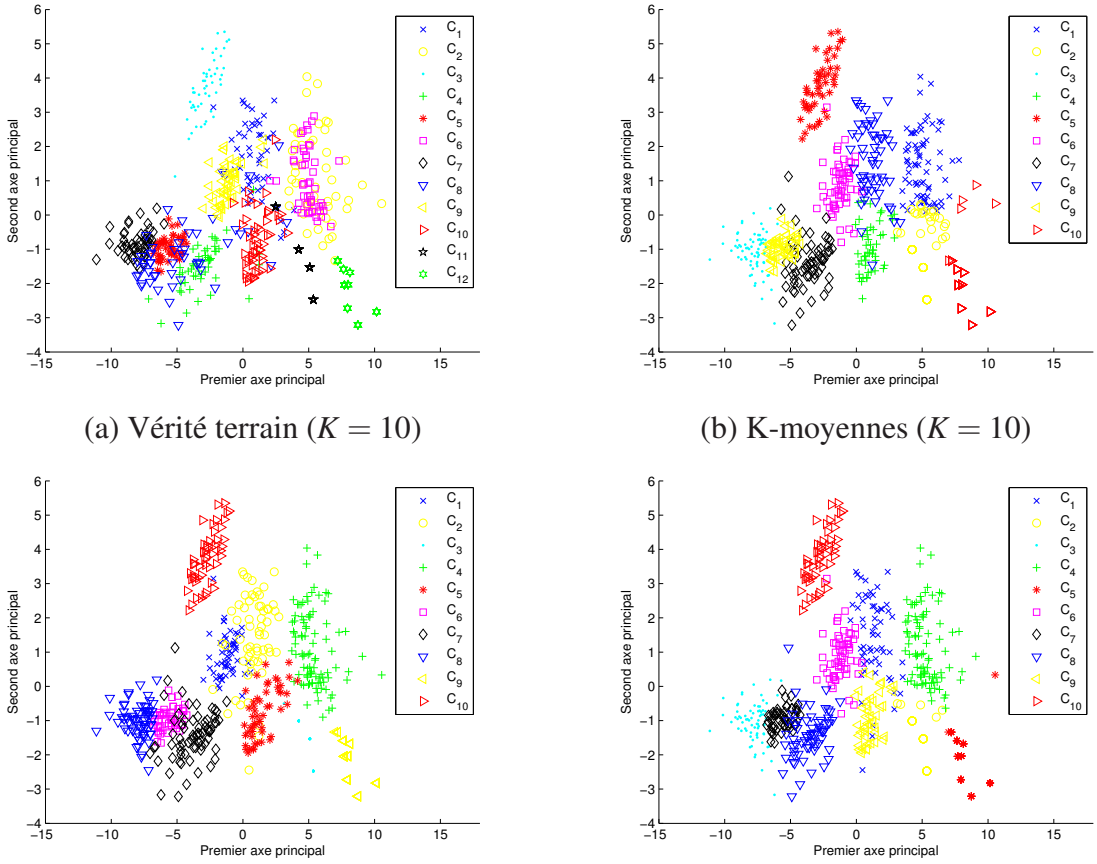
$K = 9$. Ces scores sont obtenus par comparaison avec la partition "vérité-terrain" fournie par les biologistes, mais après fusion des classes présentant un fort chevauchement : par exemple, comme illustré par la matrice de confusion de la table C.2-Annexe C, le groupe C_9 contient 86% de la classe 1 et 98% de la classe 9 pour l'algorithme des K-moyennes. Ici, nous procédons à trois fusions afin de pouvoir calculer les scores de performance des méthodes : *Asterionella glacialis* en cellules seules (croix bleues, C_1) avec *Thalassiosira rotula* en cellules seules (triangles jaunes, C_9), *Asterionella glacialis* en colonies (cercles jaunes, C_2) avec *Lauderia annulata* en cellules seules (carrés roses, C_6), et *Cylindrothetca closterium* ('+' verts, C_4) avec *Skeletonema costatum* en colonies (triangles bleus, C_8). Nous constatons alors que les scores les plus élevés sont obtenus par l'algorithme de classification spectrale (78.8% de bonne reconnaissance contre 76.0% pour les K-moyennes et 78.5% pour la classification hiérarchique).

Hypothèse $K = 10$

Dans le cas où le nombre de groupes recherchés est égal à 10 (cf. figure 6.7), nous pouvons noter que les trois algorithmes testés parviennent à séparer les classes bleue (représentée par des croix, C_1) et jaune (représentée par des triangles, C_9), contrairement au cas $K = 9$ (points bleus et triangles jaunes pour les K-moyennes, points bleus et '+' verts pour la classification hiérarchique, et triangles rouges et cercles jaunes pour la classification spectrale). Cependant, les partitions obtenues par ces trois méthodes (cf. figures 6.7(b), (c) et (d)), montrent que la fusion des classes jaune (représentée par des cercles, C_2) et rose (représentée par des carrés, C_6) persiste.

Méthodes	Indice de Rand	F-mesure	% de reconnaissance
KM	0.811	0.694	74.0
CAH	0.821	0.708	77.0
SC	**0.823**	**0.718**	**77.3**

TABLE 6.3 – Indices de Rand, F-mesures et pourcentages de bonne reconnaissance obtenus pour les différents algorithmes de classification non supervisée, sur la base "attributs" ($K = 10$).



(a) Vérité terrain ($K = 10$)

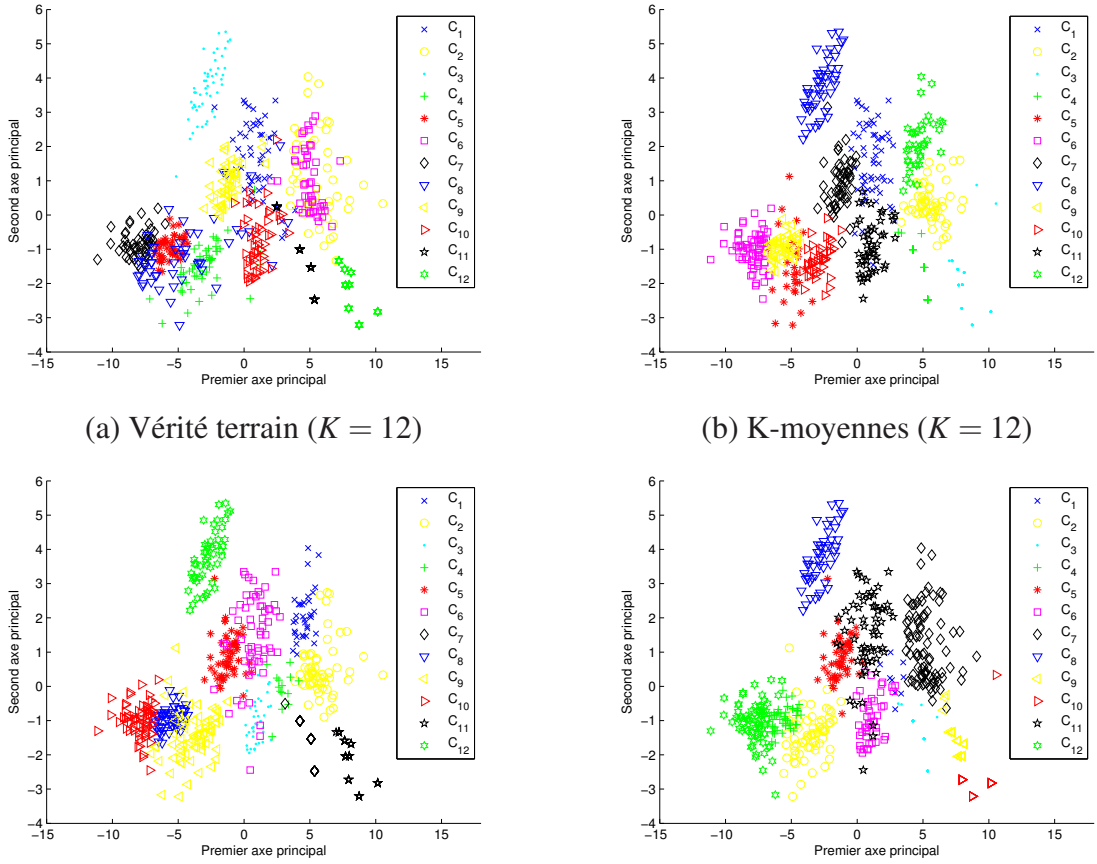
(b) K-moyennes ($K = 10$)

(c) Classification hiérarchique ($K = 10$)

(d) Classification spectrale ($K = 10$)

FIGURE 6.7 – Visualisation 2D des résultats obtenus par les différents algorithmes sur la base ”attributs” ($K = 10$).

Le tableau 6.3 présente les scores de performance ainsi obtenus. Ici, nous décidons, après analyse des matrices de confusion (cf. tableaux C.3, C.7 et C.11 en Annexe C), de fusionner les classes représentant l'espèce *Asterionella glacialis* en colonies (cercles jaunes, C_2) avec l'espèce *Lauderia annulata* en cellules seules (carrés roses, C_6), ainsi que l'espèce *Skeletonema costatum* en cellules seules (losanges noirs, C_7) avec l'espèce *Skeletonema costatum* en colonies (triangles bleus, C_8). Nous pouvons alors noter que l'algorithme de classification spectrale permet, une fois de plus, d'obtenir les meilleurs scores de performance en termes d'indices de Rand (0.823), de F-mesures (0.718) et de pourcentages de bonne reconnaissance (77.3%), après fusion des espèces précédemment citées.



(a) Vérité terrain ($K = 12$)

(b) K-moyennes ($K = 12$)

(c) Classification hiérarchique ($K = 12$)

(d) Classification spectrale ($K = 12$)

FIGURE 6.8 – Visualisation 2D des résultats obtenus par les différents algorithmes sur la base "attributs" ($K = 12$).

Hypothèse $K = 12$

La figure 6.8 montre les partitions obtenues pour les différentes méthodes testées à $K = 12$. Nous remarquons que, contrairement à la classification spectrale, l'algorithme de classification hiérarchique, ainsi que celui des K-moyennes, produisent une mauvaise partition de deux espèces de cellules (*Asterionella glacialis* en colonies et *Lauderia annulata* en cellules seules) : groupes vert (représenté par des étoiles) et jaune (représenté par des cercles) pour les K-moyennes, et groupes bleu (représenté par des croix) et jaune (représenté par des cercles) pour la classification hiérarchique. En revanche, l'algorithme de classification spectrale scinde la classe verte (représentée par des étoiles, C_{12}) en deux groupes (triangles jaunes et triangles rouges). Le nombre de groupes recherchés étant égal à celui fourni par les biologistes, il est possible de comparer directement les résultats obtenus, en termes d'indice de Rand et de F-mesures.

Méthodes	Indice de Rand	F-mesure	% de reconnaissance
KM	0.810	0.712	78.7
CAH	0.802	0.706	76.7
SC	**0.817**	**0.714**	**81.5**

TABLE 6.4 – Indices de Rand, F-mesures et pourcentages de bonne reconnaissance obtenus pour les différents algorithmes de classification non supervisée, sur la base "attributs" ($K = 12$).

Nous constatons, grâce au tableau 6.4, que l'algorithme de classification spectrale appliqué à la base "attributs", permet d'obtenir des scores de performance satisfaisants (indice de Rand égal à 0.817 et F-mesure égale à 0.714, ce qui représente 81.5% de bonne reconnaissance). Les erreurs de classification sont notamment dues au chevauchement important de certaines espèces de cellules phytoplanctoniques. Dans un contexte totalement non supervisé, il semble donc difficile de discriminer correctement toutes les espèces phytoplanctoniques.

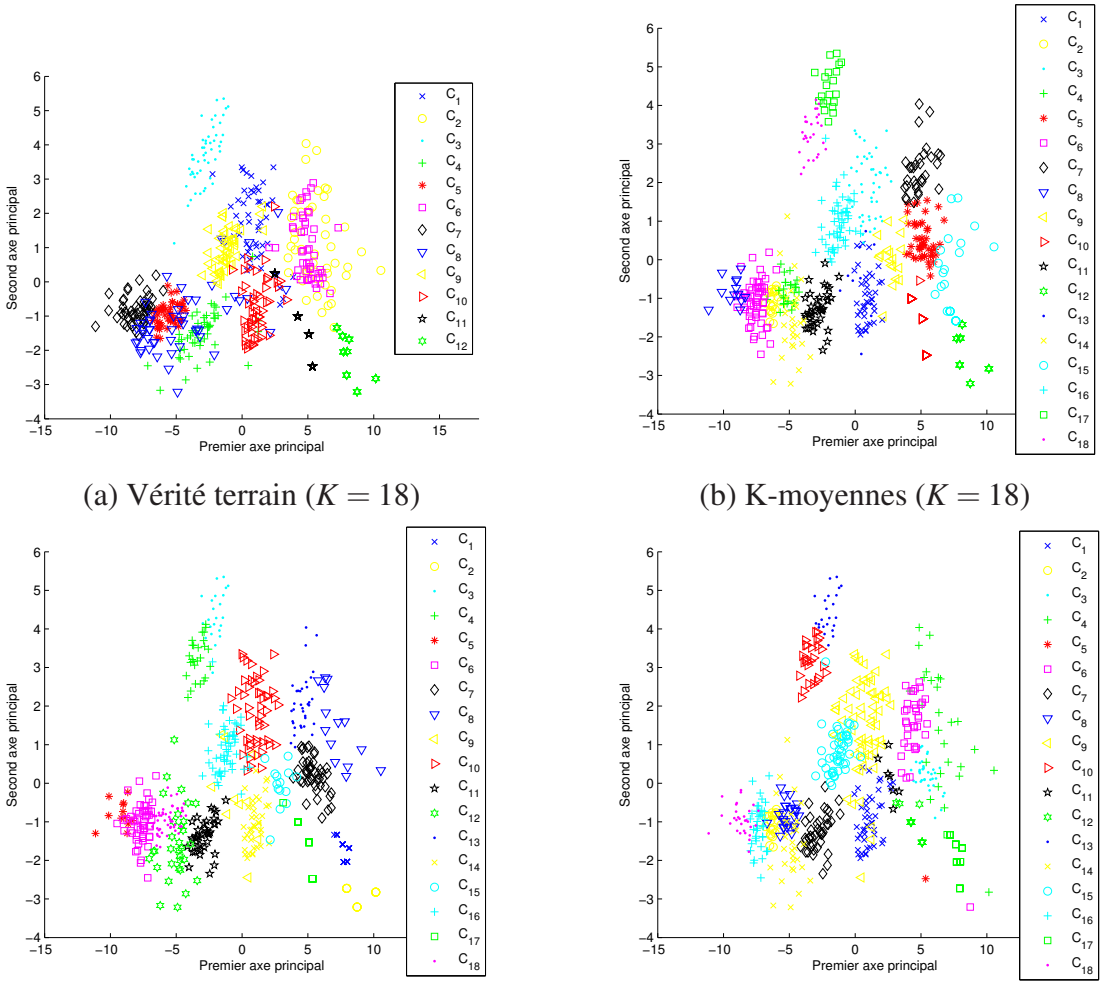
Hypothèse $K = 18$

Nous choisissons, enfin, de tester les méthodes de classification non supervisée dans le cas $K = 18$ afin d'étudier la présence éventuelle de sous-groupes appartenant à la même espèce.

La figure 6.9 montre que le groupe bleu (représenté par des points) de la figure 6.9(a) peut être divisé en deux sous-groupes de cellules (carrés verts et points roses pour l'algorithme des K-moyennes, '+' verts et points bleus pour la classification hiérarchique, triangles rouges et points bleus pour la classification spectrale). Cependant, la valeur élevée du nombre de groupes recherchés ($K = 18$), tend à former des groupes contenant un faible nombre de cellules qui appartiennent à des espèces différentes (cf. tableaux C.5, C.9 et C.13 en Annexe C). Par exemple, pour l'algorithme des K-moyennes, le groupe C_9 (triangles jaunes) est composé de 17 cellules appartenant à 5 espèces différentes ; pour la classification hiérarchique, le groupe C_{15} (cercles bleus) est composé de 26 cellules appartenant à 6 espèces différentes ; pour la classification spectrale, le groupe C_{11} (étoiles noirs) est composé de 19 cellules appartenant à 5 espèces différentes. Ces confusions entre classes et l'absence de validation biologique, ne permettent pas d'évaluer les partitions obtenues par les algorithmes. Notons que pour $K = 18$, la valeur propre associée est égale à 0.4742 (cf. figure 6.4(a)). Celle-ci n'est donc pas pertinente en classification spectrale.

6.3.4 Classification semi-supervisée avec contraintes

Dans cette section, nous proposons de retenir $K = 12$ comme nombre de groupes. Nous supposons disposer de quelques étiquettes de classes permettant de générer des contraintes de



(a) Vérité terrain ($K = 18$)

(b) K-moyennes ($K = 18$)

(c) Classification hiérarchique ($K = 18$)

(d) Classification spectrale ($K = 18$)

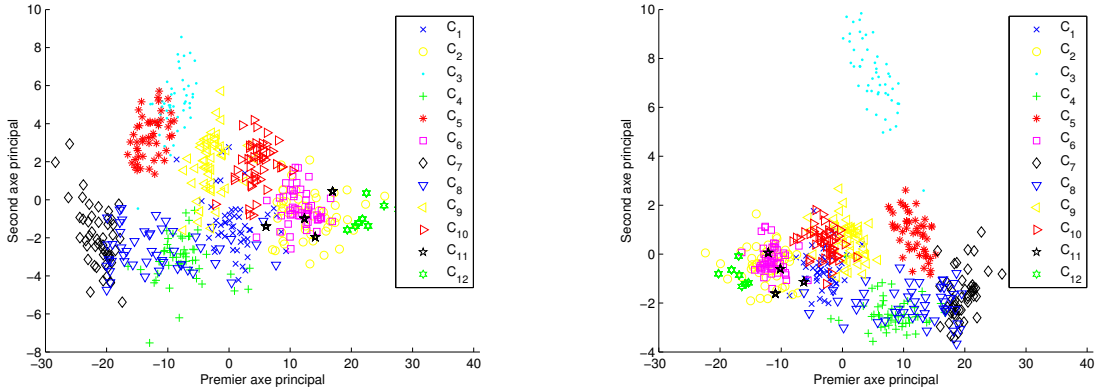
FIGURE 6.9 – Visualisation 2D des résultats obtenus par les différents algorithmes sur la base "attributs" ($K = 18$).

comparaison.

Visualisation des données par projection sous contraintes.

La figure 6.10 montre une visualisation en deux dimensions de l'ACP contrainte (avec une cellule étiquetée par espèce, puis deux cellules étiquetées par espèce), pour la base "attributs" (cf. figures 6.10(a) et 6.10(b)). Dans l'espace de projection construit à partir des deux premières composantes principales, l'ajout de contraintes de comparaison permet donc une meilleure discrimination des espèces phytoplanctoniques que le cas d'ACP standard (cf. figure 6.2(c)).

En effet, certaines classes se démarquent des autres grâce à l'intégration de ces contraintes de comparaison. En particulier, la classe rouge (représentée par des croix) présente un chevau-



(a) ACP 2% - Visualisation 2D

(b) ACP 4% - Visualisation 2D

FIGURE 6.10 – Résultats obtenus par ACP pour 2 et 4% de contraintes (visualisation 2D), sur la base "attributs".

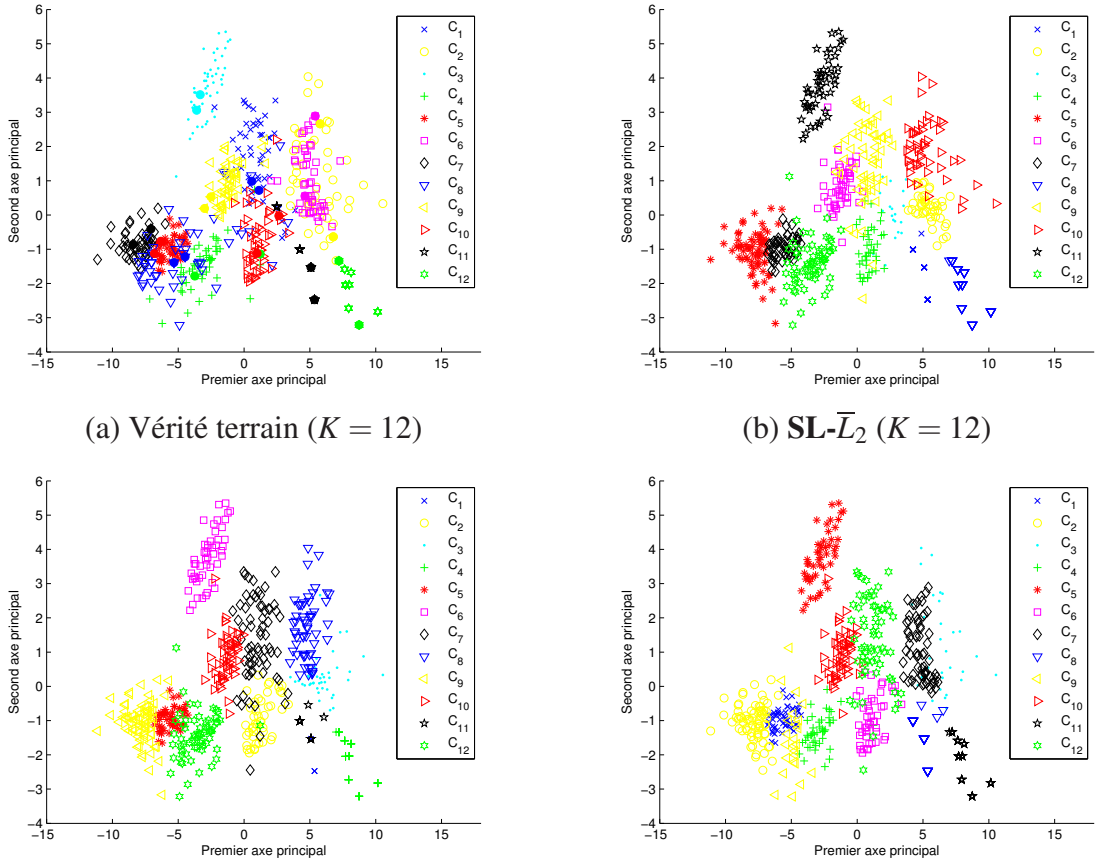
chement moins important avec la classe bleue (représentée par des triangles), que pour l'ACP standard.

Résultats obtenus par les algorithmes semi-supervisés.

La figure 6.11 montre les partitions obtenues par les algorithmes semi-supervisés **SL-$\overline{L}_2$**, **FCSC-θSP** et **SSSC** pour $K = 12$. Nous présentons les résultats dans l'espace de projection obtenu à partir de l'analyse en composantes principales classique. Cet espace est construit à partir des deux premiers axes principaux. Ici, nous supposons disposer de deux cellules étiquetées par espèce, soit 24 cellules étiquetées, ce qui représente 4% de la totalité des cellules.

Les cellules contraintes sont représentées par des ronds pleins sur la figure 6.11(a). Visuellement, il est possible de montrer que, contrairement aux méthodes **FCSC-θSP** et **SL-$\overline{L}_2$**, l'algorithme **SSSC** proposé permet d'obtenir une discrimination plus précise des classes jaune (représentée par des cercles sur la figure 6.11(a)) et rose (représentée par des carrés sur la figure 6.11(a)), grâce à l'intégration de contraintes de comparaison. De plus, la méthode proposée est la seule capable de respecter la totalité des contraintes de comparaison générées entre les classes bleue (représentée par des triangles) et verte (représentée par des '+'). Nous complétons cette analyse en présentant les scores de performance obtenus par les trois algorithmes (par comparaison avec la partition "vérité-terrain" fournie par les biologistes) dans le tableau 6.5.

Les scores obtenus pour 4% d'étiquettes connues, mettent en évidence une amélioration significative des performances de classification pour toutes les méthodes testées. Cependant, l'indice de Rand le plus élevé est obtenu pour la méthode **SSSC** proposée (ainsi que la valeur de



(a) Vérité terrain ($K = 12$)

(b) **SL-$\overline{L}_2$** ($K = 12$)

(c) **FCSC-θSP** ($K = 12$)

(d) **SSSC** ($K = 12$)

FIGURE 6.11 – Visualisation 2D des résultats obtenus par les différents algorithmes semi-supervisés (4% d'étiquettes connues) sur la base "attributs" ($K = 12$).

% d'étiquettes connues	Méthodes	% *ML*	% *CL*	% Total	J_{MNCut}	Indice de Rand	F-mesure
0	SL-$\overline{L}_2$	/	/	/	**0.303**	**0.817**	**0.714**
	FCSC-θSP	/	/	/	**0.303**	**0.817**	**0.714**
	SSSC	/	/	/	**0.303**	**0.817**	**0.714**
4	SL-$\overline{L}_2$	66.7	98.1	82.4	0.343	0.844	0.824
	FCSC-θSP	**100.0**	98.5	99.2	0.377	0.834	0.802
	SSSC	**100.0**	**100.0**	**100.0**	**0.365**	**0.850**	**0.842**

TABLE 6.5 – Scores de performances sur la base de cellules issues d'un échantillon naturel ($K = 12$), pour les attributs caractéristiques (4% d'étiquettes connues).

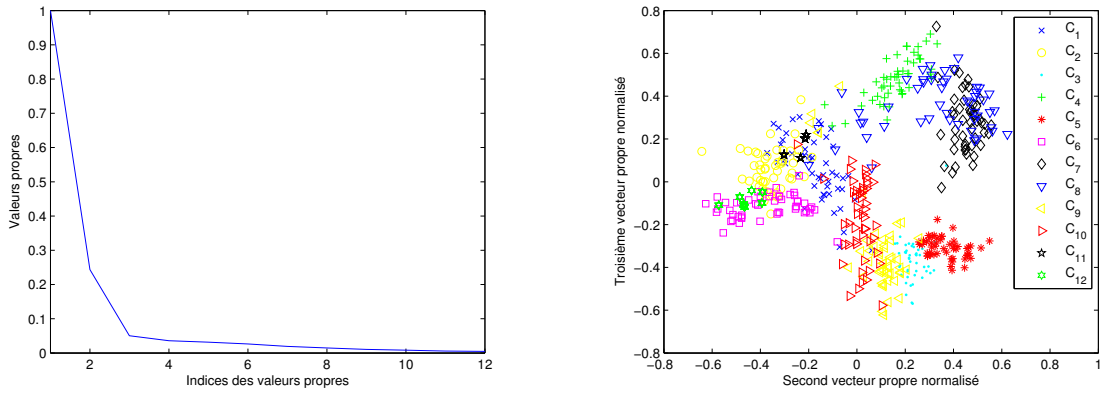
F-mesure). Cette dernière offre également les meilleurs taux de respect des contraintes des deux types (100.0% pour les contraintes "Must-Link" et 100.0% pour les contraintes "Cannot-Link"), avec une valeur de coupe normalisée satisfaisante (0.365 contre 0.303 pour la classification non

contrainte).

6.4 Expérimentations sur la base ”similarités”

6.4.1 Visualisation par projection dans l’espace spectral non contraint

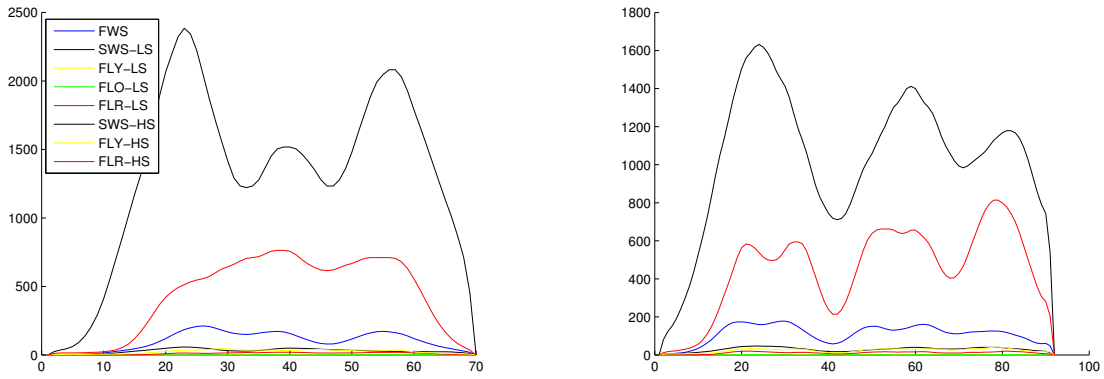
Nous appliquons l’algorithme de classification spectrale proposée par Ng et al. [Ng et al., 2002] sur la base ”similarités”, afin d’obtenir une visualisation plane (en deux dimensions) des projections des cellules dans l’espace spectral (comme montré sur la figure 6.12). Cette représentation 2D est définie à partir des vecteurs propres principaux de la matrice Laplacienne $\overline{L}_2$.



(a) Valeurs propres triées (b) Espace spectral

FIGURE 6.12 – Projections des points sur la sphère unité, grâce aux différents vecteurs propres de la matrice Laplacienne normalisée symétrique (second et troisième vecteurs propres).

La figure 6.12(a) montre les K plus grandes valeurs propres de la matrice Laplacienne $\overline{L}_2$, associées aux K vecteurs propres permettant de construire l’espace spectral. La figure 6.12(b), quant à elle, permet d’obtenir une visualisation en deux dimensions des différentes classes par projection des objets dans le plan construit à partir du second et du troisième vecteurs propres. Il est alors aisé de montrer la difficulté de discrimination des cellules en utilisant les appariements élastiques des signaux. Il est notamment possible de voir des chevauchements importants entre différentes espèces : par exemple, entre la classe verte (*Lauderia annulata* en colonies, représentée par des étoiles, C_{12}) et rose (*Lauderia annulata* en cellules seules, représentée par des carrés, C_6), et entre la classe bleue (*Skeletonema costatum* en colonies, représentée par des triangles, C_8) et noire (*Skeletonema costatum* en cellules seules, représentée par des losanges, C_7).



(a) Cellule seule (b) Colonie

FIGURE 6.13 – Illustration de la variabilité entre les cellules seules et les colonies appartenant à l'espèce *Skeletonema costatum*, grâce aux signaux cytométriques.

Par exemple, nous représentons les signaux profils obtenus par cytométrie, pour une cellule seule et une colonie de l'espèce *Skeletonema costatum* (cf. figure 6.13). Nous pouvons alors noter une forte similitude entre ces profils, qui explique donc le chevauchement important des deux classes correspondantes (triangles bleus et losanges noirs), dans l'espace spectral.

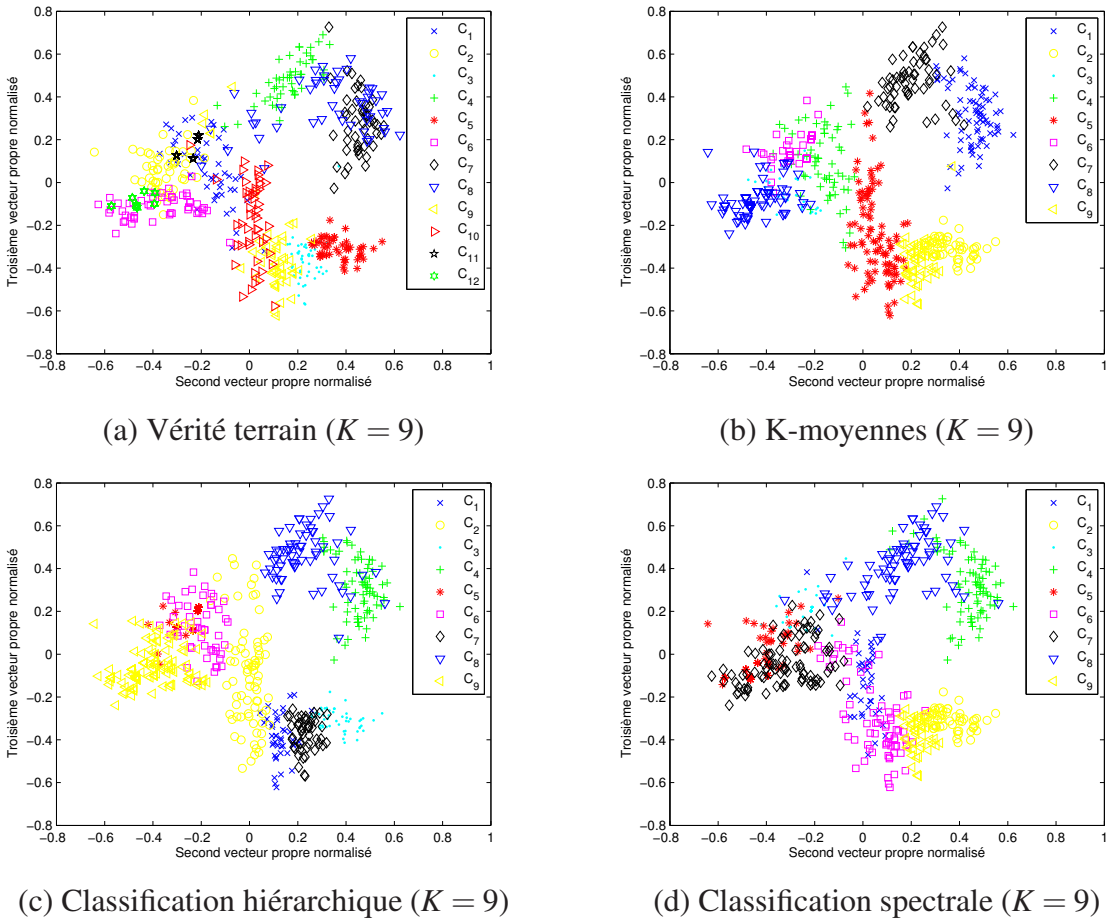
6.4.2 Classification non supervisée

De la même manière que pour la base "attributs", nous appliquons des méthodes de classification non-supervisée, dans le but de montrer la difficulté de discrimination des espèces phytoplanctoniques, sur la base composée des similarités entre les signaux cytométriques de chacune des cellules. Les trois algorithmes utilisés sont donc :

- **la classification ascendante hiérarchique**,
- **l'algorithme des K-moyennes**,
- **la classification spectrale**.

L'initialisation des ces algorithmes est identique à celle présentée dans la section 6.3.3. Chacune des cellules étant décrite par un vecteur similarités (de dimension $N \times 1$, avec N étant égal au nombre total d'objets), les lignes de la matrice composée de ces valeurs, peuvent être présentées en entrée de chacun des classifieurs. Pour nos expérimentations, nous choisissons de visualiser la partition obtenue par ces derniers, dans l'espace spectral formé par le second et le troisième vecteurs propres normalisés. L'étude est réalisée pour les valeurs de K obtenues pour la base "attributs" (K =9, 10, 12 et 18), afin de pouvoir comparer les résultats de partitionnement entre les deux méthodes d'extraction de l'information (c'est-à-dire, la méthode d'extraction d'attributs et celle basée sur l'appariement élastique de séquences temporelles).

Hypothèse $K = 9$



(a) Vérité terrain ($K = 9$)

(b) K-moyennes ($K = 9$)

(c) Classification hiérarchique ($K = 9$)

(d) Classification spectrale ($K = 9$)

FIGURE 6.14 – Visualisation 2D des résultats obtenus par les différents algorithmes sur la base ”similarités” ($K = 9$).

Comme montré sur la figure 6.14 ($K = 9$), les algorithmes de classification hiérarchique, de classification spectrale ainsi que celui des K-moyennes fusionnent les classes noire (représentée par des losanges, C_7) et bleue (représentée par des triangles, C_8). Ces deux classes appartiennent à l'espèce *Skeletonema costatum*, mais se trouvent sous forme de cellules seules ou de colonies. La même observation peut être effectuée pour les classes verte (représentée par des étoiles, C_{12}) et rose (représentée par des carrés, C_6) : il s'agit, dans ce cas, de l'espèce *Lauderia annulata* (cellule seules et colonies). Afin de compléter ces résultats, nous choisissons de fusionner les classes de colonies et de cellules seules dans une même espèce (après analyse des matrices de confusion présentées en Annexe C).

En résumé, nous fusionnons les classes suivantes : *Lauderia annulata* en colonies et en

cellules seules, *Skeletonema costatum* en colonies et en cellules seules et *Thalassiosira rotula* en colonies et en cellules seules. Nous notons ici que les cellules fusionnées (c'est-à-dire, celles présentant des chevauchements importants) appartiennent à la même espèce, contrairement à la base "attributs". De cette manière, nous obtenons neuf classes et pouvons les comparer avec les groupes obtenus par les différents algorithmes.

Méthodes	Indice de Rand	F-mesure	% de reconnaissance
KM	**0.907**	**0.814**	**83.5**
CAH	0.892	0.762	75.0
SC	0.896	0.796	82.3

TABLE 6.6 – Indices de Rand, F-mesures et pourcentages de bonne reconnaissance obtenus pour les différents algorithmes de classification non supervisée, sur la base "similarités" ($K = 9$).

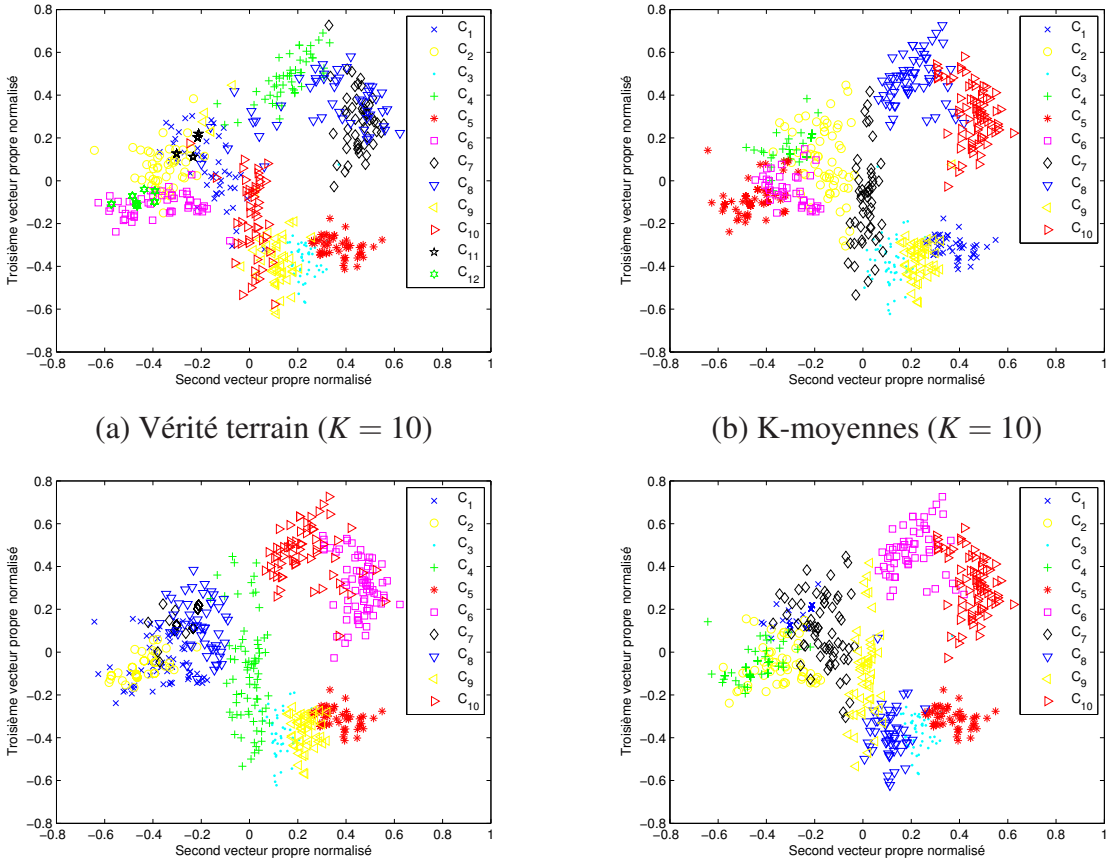
Les scores présentés dans le tableau 6.6 permettent de mettre en évidence l'algorithme des K-moyennes pour le cas $K = 9$. La supériorité des résultats obtenus par cet algorithme peut s'expliquer par le fait, qu'après fusion des classes se chevauchant, une séparation linéaire s'avère suffisante pour discriminer les différents groupes d'espèces phytoplanctoniques. La classification spectrale obtient les seconds meilleurs scores de performance avec une différence de pourcentage de bonne reconnaissance égale à 1.2%, par rapport à celui obtenu par les K-moyennes. A l'exception de l'algorithme de classification hiérarchique, les résultats de partitionnement apparaissent meilleurs que ceux obtenus sur la base "attributs" (cf. tableau 6.2).

Hypothèse $K = 10$

La figure 6.15 présente les résultats obtenus par les différents algorithmes pour un nombre de groupes K fixé à 10. De la même manière que pour le cas $K = 9$, les classes noire (représentée par des losanges, C_7) et bleue (représentée par des triangles, C_8) sont fusionnées par les algorithmes de classification non supervisée. Le tableau 6.7 présente les scores de performance obtenus dans le cas $K = 10$, après fusion des classes présentant le plus important chevauchement.

Méthodes	Indice de Rand	F-mesure	% de reconnaissance
KM	**0.901**	**0.788**	**83.2**
CAH	0.889	0.772	78.3
SC	0.895	0.781	79.8

TABLE 6.7 – Indices de Rand, F-mesures et pourcentages de bonne reconnaissance obtenus pour les différents algorithmes de classification non supervisée, sur la base "similarités" ($K = 10$).



(a) Vérité terrain ($K = 10$)

(b) K-moyennes ($K = 10$)

(c) Classification hiérarchique ($K = 10$)

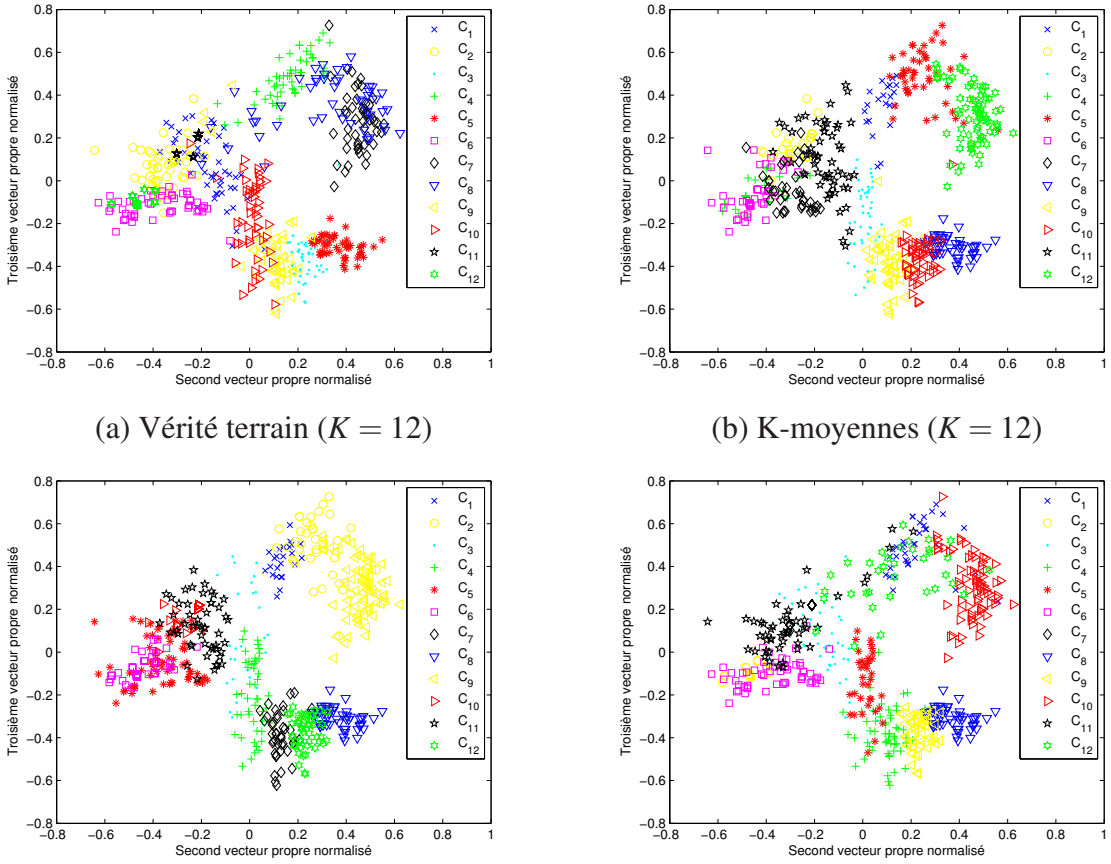
(d) Classification spectrale ($K = 10$)

FIGURE 6.15 – Visualisation 2D des résultats obtenus par les différents algorithmes sur la base ”similarités” ($K = 10$).

Dans le cas $K = 10$, et pour la base ”similarités”, les espèces présentant un chevauchement important sont les suivantes : *Skeletonema costatum* en cellules seules et en colonies, et *Lauderia annulata* en cellules seules et en colonies (cf. matrices de confusion C.15, C.19 et C.23 présentées en Annexe C). Après fusion de ces espèces, nous remarquons que, de la même manière que pour le cas précédent ($K = 9$), les meilleurs scores de performance sont obtenus par l'algorithme des K-moyennes (indice de Rand égal à 0.901 et pourcentage de bonne reconnaissance égal à 83.2%).

Hypothèse $K = 12$

La figure 6.16 montre les partitions obtenues pour les différentes méthodes testées à $K = 12$. De la même manière que pour les deux cas précédents ($K = 9$ et $K = 10$), il est possible de constater une fusion des classes noire (représentée par des losanges, C_7) et bleue (représentée



(a) Vérité terrain ($K = 12$)

(b) K-moyennes ($K = 12$)

(c) Classification hiérarchique ($K = 12$)

(d) Classification spectrale ($K = 12$)

FIGURE 6.16 – Visualisation 2D des résultats obtenus par les différents algorithmes sur la base "similarités" ($K = 12$).

par des triangles, C_8) de la figure 6.16(a). Le nombre de groupes recherchés étant égal à celui fourni par les biologistes, il est possible de comparer directement les résultats obtenus, en termes d'indice de Rand et de F-mesures.

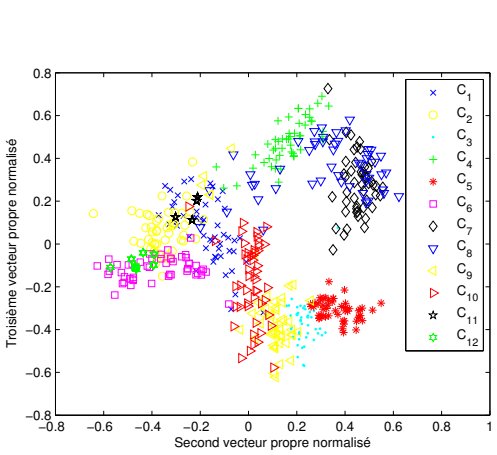
Méthodes	Indice de Rand	F-mesure	% de reconnaissance
KM	0.899	0.804	81.7
CAH	0.905	0.810	83.0
SC	**0.914**	**0.836**	**84.2**

TABLE 6.8 – Indices de Rand, F-mesures et pourcentages de bonne reconnaissance obtenus pour les différents algorithmes de classification non supervisée, sur la base "similarités" ($K = 12$).

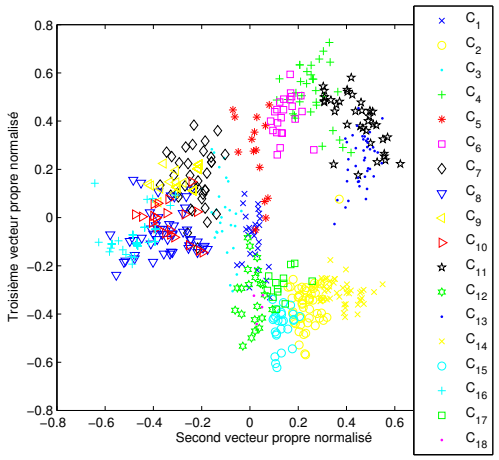
Le tableau 6.8 présente les scores de performances obtenus par les algorithmes testés. Nous pouvons constater que la classification spectrale appliquée à la base "similarités", permet d'obtenir les scores de performance les plus élevés (indice de Rand égal à 0.914 et F-mesure égale

à 0.836, ce qui représente 84.2% de bonne reconnaissance). Les erreurs de classification sont principalement dues au chevauchement important de certaines classes de cellules (notamment, celles représentant les espèces *Skeletonema costatum* en cellules seules et en colonies). Néanmoins, les scores de performance apparaissent meilleurs que ceux obtenus sur la base "attributs" (cf. tableau 6.4) : par exemple, pour la classification spectrale, nous constatons une augmentation de 2.7% d'objets correctement reconnus.

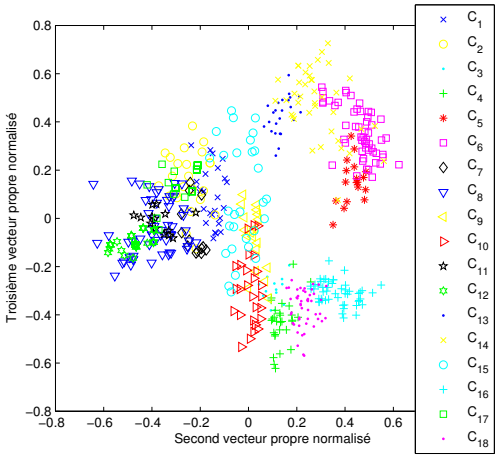
Hypothèse $K = 18$



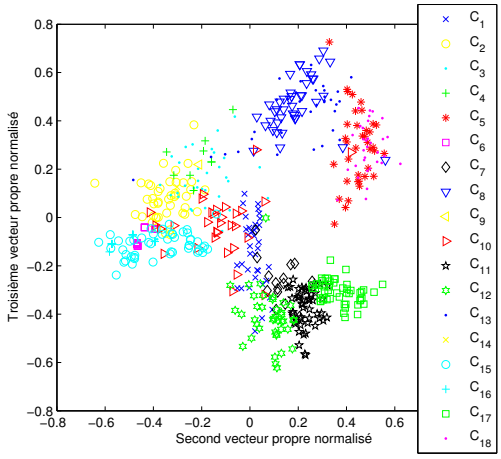
(a) Vérité terrain ($K = 18$)



(b) K-moyennes ($K = 18$)



(a) Classification hiérarchique ($K = 18$)



(b) Classification spectrale ($K = 18$)

FIGURE 6.17 – Visualisation 2D des résultats obtenus par les différents algorithmes sur la base "similarités" ($K = 18$).

Nous choisissons, enfin, de tester les méthodes de classification non supervisée dans le cas $K = 18$ afin de mettre en évidence la présence de sous-groupes appartenant à la même espèce.

Les figures 6.17(b) et 6.17(c) montrent que l'algorithme des K-moyennes ainsi que celui de classification hiérarchique, scindent les classes C_4 et C_6 en deux sous-groupes, contrairement à l'algorithme de classification spectrale (par exemple, pour la classe C_4, carrés roses et '+' verts pour les K-moyennes, points bleus et croix jaunes pour la classification hiérarchique). Comme pour la base "attributs", les confusions importantes entre classes, conjuguées à l'absence de validation biologique, ne permettent pas d'évaluer les partitions obtenues par les différents algorithmes.

6.4.3 Classification semi-supervisée avec contraintes

Dans cette section, nous procédons d'abord à une visualisation plane pour explorer les données et ensuite à une classification automatique semi-supervisée, en fixant le nombre de groupes à 12.

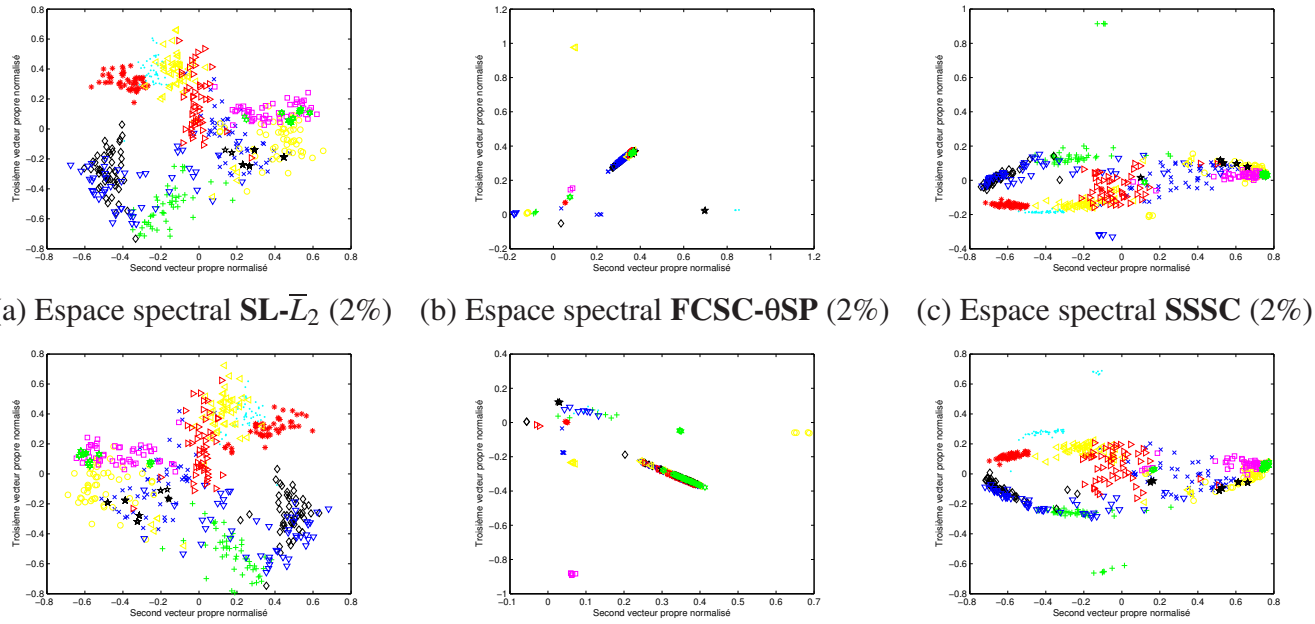
Visualisation des données par projection sous contraintes.

Nous montrons l'impact des contraintes de comparaison par paires sur la représentation des données en groupes, grâce à une visualisation en deux dimensions de l'espace spectral contraint (2% et 4% de contraintes), pour la base "similarités" et pour les algorithmes **SL-$\overline{L}_2$**, **FCSC-θSP** et **SSSC** (cf. figure 6.18). Visuellement, nous pouvons noter que les espaces de projection obtenus par les algorithmes **FCSC-θSP** (cf. figures 6.18(b) et (e)) et **SSSC** (cf. figures 6.18(c) et (f)) permettent de respecter les contraintes de comparaison par paires d'objets. Pour ces deux méthodes, les cellules contraintes sont représentées par les objets les plus isolés dans l'espace spectral. Cependant, dans ce plan, la discrimination des groupes de cellules non contraintes par la méthode **FCSC-θSP** apparaît plus difficile que pour les autres algorithmes (les données sont compactées dans l'intervalle [0.2-0.4 ;0.2-0.4]).

Résultats obtenus par les algorithmes semi-supervisés.

La figure 6.19 montre les partitions obtenues par les algorithmes semi-supervisés **SL-$\overline{L}_2$**, **FCSC-θSP** et **SSSC** pour $K = 12$. Nous présentons les résultats dans un plan 2D de l'espace spectral, obtenu à partir de l'algorithme non contraint proposé par Ng et al. [Ng et al., 2002]. Rappelons que cet espace est restreint, pour une visualisation plane, aux second et troisième vecteurs propres normalisés, de la matrice Laplacienne $\overline{L}_2$.

La figure (a) montre la partition "vérité-terrain" fournie par les biologistes. Sur cette même figure, les cellules contraintes sont représentées par des ronds pleins colorés selon le groupe d'appartenance. Contrairement aux méthodes **SL-$\overline{L}_2$** et **FCSC-θSP**, l'algorithme **SSSC** proposé



(a) Espace spectral **SL-$\overline{L}_2$** (2%) (b) Espace spectral **FCSC-θSP** (2%) (c) Espace spectral **SSSC** (2%)

(d) Espace spectral **SL-$\overline{L}_2$** (4%) (e) Espace spectral **FCSC-θSP** (4%) (f) Espace spectral **SSSC** (4%)

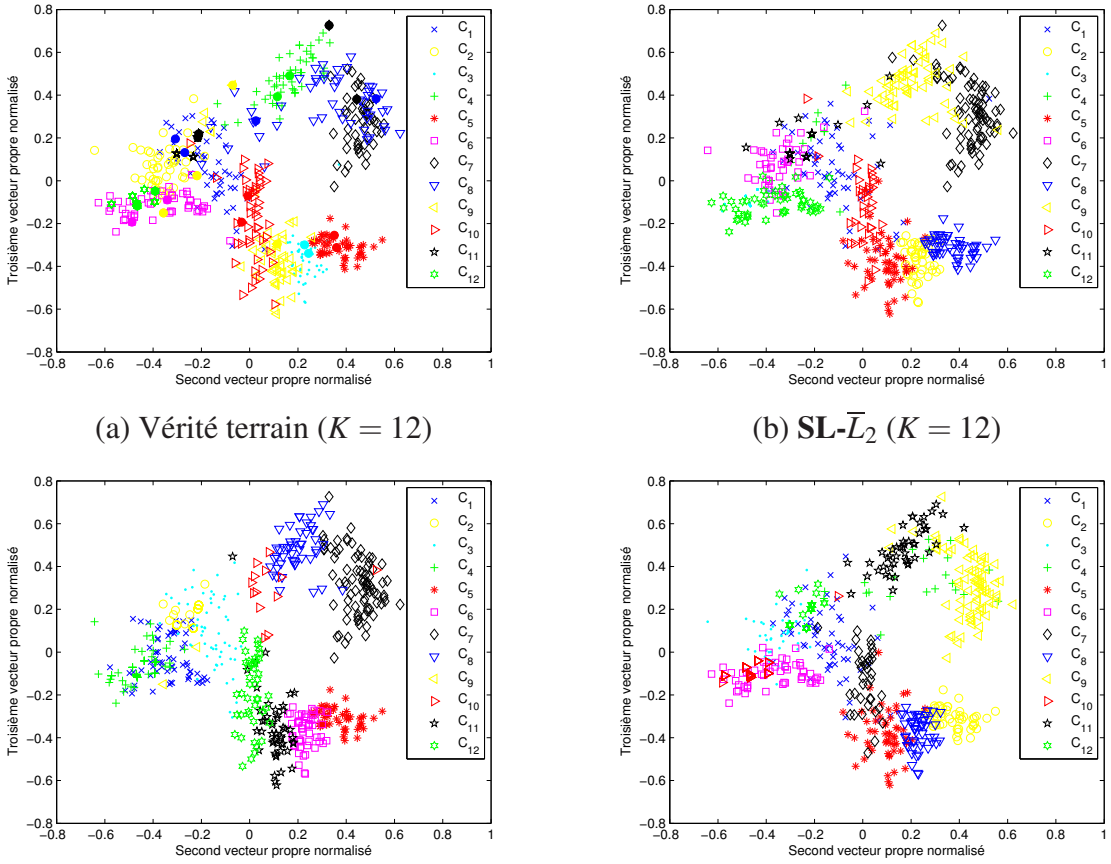
FIGURE 6.18 – Projections des points sur la sphère unité, grâce aux différents vecteurs propres de la matrice Laplacienne normalisée symétrique, sur la base "similarités" (2% et 4% de contraintes).

permet de séparer correctement les classes verte (représentée par des étoiles sur la figure 6.19(a)) et rose (représentée par des carrés sur la figure 6.19(a)), grâce à l'intégration de contraintes de comparaison par paires. Afin de mettre en évidence les performances de chaque méthode, nous présentons quelques scores de performance pour les trois algorithmes (par comparaison avec la partition "vérité-terrain" fournie par les biologistes) dans le tableau 6.9.

% d'étiquettes connues	Méthodes	% *ML*	% *CL*	% Total	J_{MNCut}	Indice de Rand	F-mesure
0	SL-$\overline{L}_2$	/	/	/	**0.882**	**0.914**	**0.836**
	FCSC-θSP	/	/	/	**0.882**	**0.914**	**0.836**
	SSSC	/	/	/	**0.882**	**0.914**	**0.836**
4	SL-$\overline{L}_2$	75.0	**98.1**	86.6	**0.884**	0.936	0.894
	FCSC-θSP	**91.7**	98.5	95.1	0.888	0.924	0.876
	SSSC	100.0	98.5	**99.3**	0.884	**0.946**	**0.912**

TABLE 6.9 – Scores de performances sur la base de cellules issues d'un échantillon naturel ($K = 12$), pour les signaux profils (4% d'étiquettes connues).

Les scores présentés dans ce tableau (pour 4% d'étiquettes connues) permettent de montrer une amélioration des performances de classification grâce à l'introduction de connaissances



(a) Vérité terrain ($K = 12$)

(b) **SL-**$\overline{L}_2$ ($K = 12$)

(c) **FCSC-θSP** ($K = 12$)

(d) **SSSC** ($K = 12$)

FIGURE 6.19 – Visualisation 2D des résultats obtenus par les différents algorithmes semi-supervisés (4% d'étiquettes connues) sur la base "similarités" ($K = 12$).

contextuelles. Cependant, l'indice de Rand le plus élevé est obtenu pour la méthode **SSSC** proposée (ainsi que la valeur de F-mesure). Cette dernière offre également les meilleurs taux de respect des contraintes des deux types (100.0% pour les contraintes "Must-Link" et 98.5% pour les contraintes "Cannot-Link"), avec une valeur de coupe normalisée satisfaisante par rapport à celle obtenue dans le cas non supervisée, c'est-à-dire lorsqu'aucune étiquette n'est connue (0.884 contre 0.882 pour la classification non contrainte). Nous pouvons également remarquer une nette amélioration des résultats par rapport à ceux obtenus pour la base composée des attributs caractéristiques des signaux (cf. tableau 6.5).

6.5 Conclusion

Nous avons présenté une démarche méthodologique pour la recherche de groupes dans un échantillon issu du milieu marin. Deux bases de données ont été étudiées : une base "attributs"

et une base ”similarités”. Nous avons décrit la manière dont les biologistes explorent leurs données. L'estimation automatique du nombre de groupes a été réalisée par deux techniques qui ont donné des valeurs différentes de celle estimée par les biologistes. Nous avons considéré différentes hypothèses sur le nombre de classes et analysé les résultats obtenus par divers algorithmes de groupement. Nous avons effectué des visualisations planes afin de se construire une idée de la structure des données. La concordance des résultats de visualisation et de groupement obtenus devrait conforter l'expert dans son analyse.

Il est clair que l'apport d'informations, même partielles, sur les données peut améliorer les résultats de la classification. Nous avons appliqué la même démarche semi-supervisée sur les deux bases de données avec 4% d'étiquettes connues. Nous avons comparé différents algorithmes de classification spectrale contrainte des chapitres 3 et 4, en utilisant des indices de performance. Ces indices permettent alors de mettre en avant l'intérêt des algorithmes de classification spectrale et notamment celui semi-supervisé proposé.

Conclusion générale et perspectives

1 Conclusion générale

Notre travail de thèse se situe dans le cadre de la classification semi-supervisée. Celle-ci reçoit de plus en plus d'intérêt dans la communauté scientifique vu qu'elle apporte des solutions intéressantes à de nombreux problèmes réels où les connaissances disponibles sont partielles. Dans ce contexte, l'élaboration d'un classifieur s'appuie sur la disponibilité de bases de données numériques abondantes et éventuellement de certaines connaissances contextuelles qualitatives, disponibles *a priori* ou recueillies par retour d'expérience. Nous avons mis l'accent sur la classification spectrale, en particulier, avec intégration de connaissances contextuelles de type contraintes de comparaison car elles sont simples à formaliser par un non spécialiste du domaine.

Les algorithmes de classification spectrale organisent les données en graphes et traitent le problème de partitionnement des données sous l'angle de coupes de graphes. Ils consistent en une étape de génération d'un espace spectral construit à partir des vecteurs propres de la matrice Laplacienne du graphe, suivie de l'étape de partitionnement dans l'espace spectral. Cet espace est censé mieux révéler la présence de groupements naturels linéairement séparables pour que le partitionnement soit réalisé par un simple algorithme type K-moyennes.

L'intégration de connaissances de plus haut niveau dans la conception des algorithmes de classification spectrale augmente, en général, leurs performances. Nous nous sommes intéressés aux connaissances type contraintes de comparaison "Must-link" et "Cannot-Link". L'espace spectral généré est supposé, dans ce cas, révéler la structuration en groupes naturels tout en respectant autant que possible ces contraintes.

Nous avons présenté un état de l'art des approches spectrales semi-supervisées contraintes. Nous y distinguons deux modalités d'intégration des contraintes : soit explicitement dans la matrice de similarités en imposant des poids (des valeurs binaires), soit implicitement comme

pénalité dans le critère de coupe normalisé.

Nous avons proposé un nouvel algorithme qui permet de générer un espace de projection par optimisation d'un critère multi-coupes normalisé avec ajustement des coefficients de pénalité dus aux contraintes. Les performances de l'algorithme sont mises en évidence sur différentes bases de données UCI par comparaison à d'autres algorithmes semi-supervisés de la littérature. Les critères de comparaison utilisés sont l'indice de Rand, la F-mesure et le pourcentage de contraintes respectées.

Les approches présentées ont été appliquées à la surveillance de l'écosystème marin. Dans ce cadre, les cellules phytoplanctoniques sont analysées par cytométrie en flux, générant des signaux profils caractéristiques. Il est d'usage d'extraire, à partir des signaux profils, les attributs caractéristiques des cellules. Pour éviter toute perte d'informations due à l'extraction d'attributs, nous avons préféré traiter les signaux profils bruts pour l'analyse des espèces phytoplanctoniques. Pour ce faire, nous avons dérivé l'approche d'appariement élastique (DTW) pour quantifier la similarité entre signaux profils qui sont, en général, de longueurs et de formes très variées.

Deux bases de données illustratives ont été utilisées : une supervisée issue d'un échantillon de culture de laboratoire et l'autre, non supervisée, issue d'un échantillon prélevé du milieu naturel. Une démarche méthodologique a été mise au point pour la recherche d'espèces, s'appuyant sur des techniques récentes de classification issues de la théorie des graphes, comparées aux approches classiques d'analyse de données. Une étude comparative a été menée employant, selon les cas, des classifieurs supervisés et semi-supervisés pour la base de culture, et non supervisés et semi-supervisés pour la base naturelle.

En résumé, dans ce travail, nous avons montré l'intérêt de la similarité élastique pour l'analyse de séquences de signaux cytométriques et l'efficacité de notre algorithme de classification spectrale contrainte par rapport aux autres algorithmes de la littérature. Ces résultats ont été validés sur des données réelles pour la caractérisation de l'écosystème marin.

2 Perspectives

Aux niveaux fondamental et applicatif, les perspectives de nos travaux se situent sur plusieurs niveaux :

– notre travail a concerné uniquement les algorithmes de classification semi-supervisée

transductive, c'est-à-dire où la prédiction est relative uniquement aux données non étiquetées sans s'intéresser à la généralisation sur d'autres bases de données tests. Il serait intéressant de regarder aussi l'apport de la classification semi-supervisée inductive.

– les contraintes de comparaison contribuent différemment dans les résultats de la classification. En effet, il arrive que certaines soient superflues, redondantes, voire même incohérentes. Il est important de déterminer les contraintes qui apportent le plus de connaissances au processus de classification et de vérifier leurs cohérences, faute de quoi les résultats de la classification peuvent même être détériorés. L'évaluation du caractère informatif et de la cohérence des contraintes, a été définie dans le chapitre 3. A la lumière de ces définitions, il est intéressant de vérifier la qualité des contraintes choisies par l'analyste et de l'informer et le guider par retour d'expérience dans son choix.

– pour éviter un processus fastidieux de génération manuelle de contraintes, il serait, peut être, plus judicieux que ce soit l'algorithme lui-même, qui génère d'une manière intelligente des contraintes qu'il soumettrait à l'approbation de l'analyste. Ceci ouvre la voie à l'apprentissage actif.

– les bases de données disponibles sont de grandes tailles. Il serait intéressant de réduire cette taille par des techniques de sélection de prototypes afin que la matrice de similarités soit de dimension raisonnable.

– il est important d'accorder une attention particulière aux données et groupes aberrants. En effet, ceux-ci contiennent souvent des informations qui peuvent être révélatrices d'incidents.

– le cytomètre en flux fournit des profils caractéristiques des cellules (ensemble de huit séquences de signaux par cellule). L'analyste compare ces profils pour élaborer ses contraintes. Une caméra a été installée sur le cytomètre et permet, de ce fait, d'avoir en plus du profil cytométrique, l'image de la cellule. L'exploitation de l'information signal et image devrait améliorer la comparaison et la reconnaissance des cellules.

– suite aux campagnes d'inter-calibration dans la Mer du Nord, nous disposons de grandes bases de données d'échantillons de mesures : billes de calibration, cellules de cultures, prélèvements du milieu marin. Ces bases serviront à la validation de nos approches de classification spectrale semi-supervisée.

Annexe A

Exemple illustratif à $K = 2$ (chapitre 4)

Description du problème. La base de données synthétiques est composée de 240 objets construits à partir d'un mélange de trois distributions gaussiennes, comme montré sur la figure A.1. Chaque distribution est représentée par une couleur et un symbole différents. La proportion de chacune d'entre elles est de $\frac{1}{3}$. Pour cet exemple, le nombre de groupes K désirés est fixé à 2.

Trois contraintes sont considérées : deux contraintes de type "Must-Link" (lignes pleines) entre des points appartenant à des groupes différents, et une contrainte de type "Cannot-Link" (ligne pointillée) entre deux points appartenant à un même groupe gaussien. Ces contraintes de comparaison par paires sont choisies délibéremment de façon à "casser" la coupe naturelle des données, obtenue par l'algorithme de classification spectrale traditionnel.

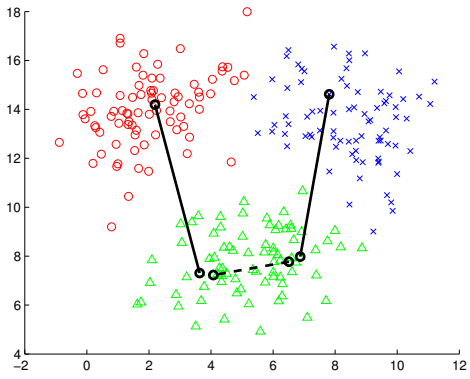
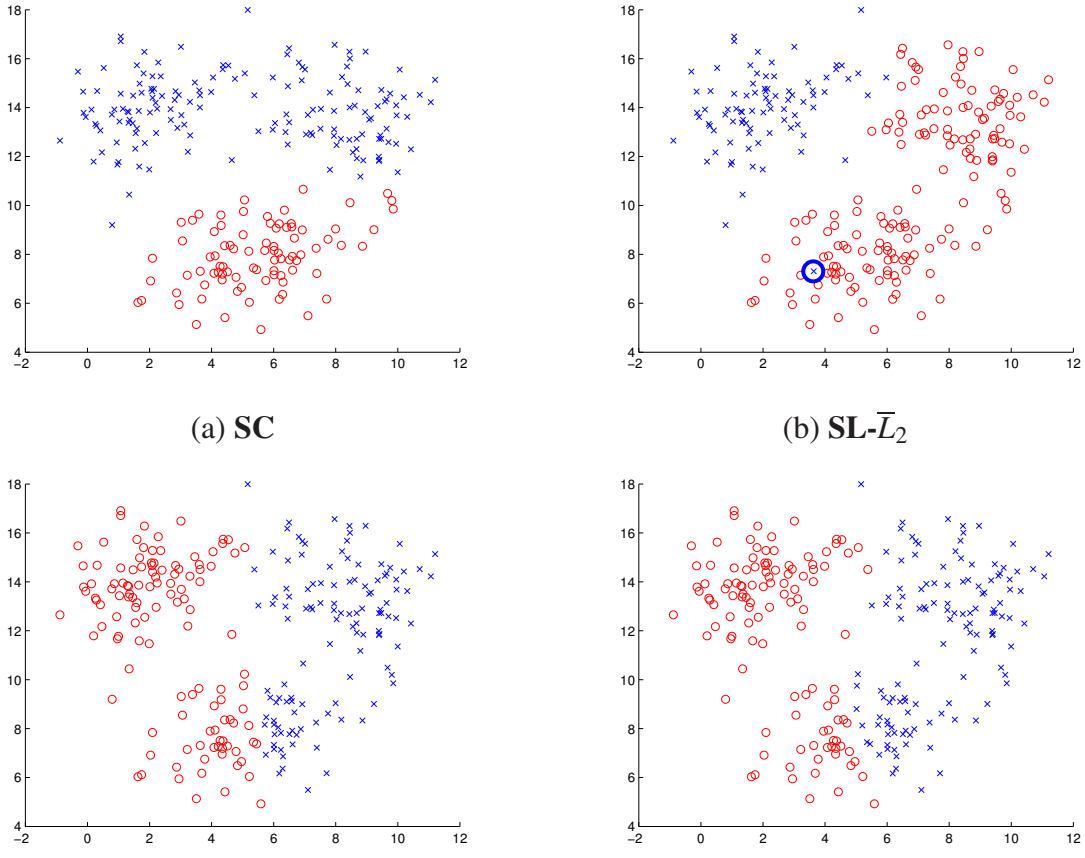


FIGURE A.1 – Données originales (trois groupes gaussiens), avec deux contraintes *ML* (lignes pleines) et une contrainte *CL* (ligne pointillée).

Pour cet exemple, la matrice de similarités est construite à partir d'un noyau gaussien, pre-

nant comme arguments un paramètre de dispersion locale σ fixé à 1, et une distance d définie comme étant la distance euclidienne.

Résultats obtenus et discussions La figure A.2 montre les partitions obtenues pour les quatre méthodes testées. Ici, les algorithmes **SSSC** et **FCSC-θSP** réussissent à faire respecter les trois contraintes définies (cf. figures (c) et (d)) et ainsi, à obtenir la partition souhaitée. En revanche, la méthode **SL-**$\overline{L}_2$, même si elle permet de faire respecter 66.6% de contraintes, ne parvient pas à casser la structure originale du groupe vert (représenté par des triangles), comme illustré sur les figures (a) et (b). De plus, même si la contrainte "Must-Link" reliant le groupe rouge (représenté par des cercles) au groupe vert (représenté par des triangles) est respectée, l'algorithme **SL-**$\overline{L}_2$ tend à isoler le point contraint. Il est alors aisé de montrer le faible pouvoir de généralisation de cette méthode, de par le fait qu'elle ne propage pas la contrainte à ses voisins (croix bleu dans le groupe du bas).



(a) **SC** (b) **SL-**$\overline{L}_2$

(c) **FCSC-θSP** (d) **SSSC**

FIGURE A.2 – Résultats de partitionnement sur trois groupes gaussiens ($K = 2$), avec deux contraintes *ML* (lignes pleines) et une contrainte *CL* (ligne pointillée).

Méthodes	J_{MNCut}	$\%total(ML_{resp}+CL_{resp})$
SC	**0.004**	0.0
SL-$\overline{L}_2$	0.014	66.6
FCSC-θSP	0.046	**100.0**
SSSC	0.041	**100.0**

TABLE A.1 – Valeurs de coupe et pourcentage total de contraintes respectées, pour les différentes méthodes, avec deux contraintes *ML* et une contrainte *CL*.

Le tableau A.1 présente deux indicateurs de performance. Les valeurs de coupe les plus élevées sont donc obtenues pour les méthodes **SSSC** et **FCSC-θSP** (0.041 et 0.046 respectivement). Ceci peut s'expliquer par le fait que ces algorithmes permettent d'obtenir une partition différente de celle naturelle. La structure originale des données n'est alors plus totalement préservée, ce qui implique une augmentation de la valeur de coupe. Cependant, pour **SSSC**, cette dernière est plus faible que celle obtenue pour **FCSC-θSP**.

Annexe B

Comparaison des résultats obtenus sur les bases UCI (chapitre 4)

"Glass1"

2%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	81.3%	0.0%	80.4%	0.9%
	Non Reconnus	0.0%	18.7%	0.0%	18.7%

TABLE B.1 – Comparaison des algorithmes pour la base "Glass1" (2% d'étiquettes connues).

5%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	73.4%	15.9%	87.4%	1.9%
	Non Reconnus	8.7%	2.0%	0.9%	9.8%

TABLE B.2 – Comparaison des algorithmes pour la base "Glass1" (5% d'étiquettes connues).

100%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	100.0%	0.0%	100.0%	0.0%
	Non Reconnus	0.0%	0.0%	0.0%	0.0%

TABLE B.3 – Comparaison des algorithmes pour la base "Glass1" (100% d'étiquettes connues).

"Hepatitis"

2%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	75.0%	12.5%	77.5%	10.0%
	Non Reconnus	0.0%	12.5%	6.2%	6.3%

TABLE B.4 – Comparaison des algorithmes pour la base "Hepatitis" (2% d'étiquettes connues).

5%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	56.2%	23.8%	67.5%	12.5%
	Non Reconnus	17.5%	2.5%	16.2%	3.8%

TABLE B.5 – Comparaison des algorithmes pour la base "Hepatitis" (5% d'étiquettes connues).

100%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	100.0%	0.0%	100.0%	0.0%
	Non Reconnus	0.0%	0.0%	0.0%	0.0%

TABLE B.6 – Comparaison des algorithmes pour la base "Hepatitis" (100% d'étiquettes connues).

"Ionosphere"

2%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	62.1%	2.3%	64.1%	0.3%
	Non Reconnus	3.1%	32.5%	0.0%	35.6%

TABLE B.7 – Comparaison des algorithmes pour la base "Ionosphere" (2% d'étiquettes connues).

5%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	72.2%	10.1%	82.3%	0.6%
	Non Reconnus	5.7%	11.4%	1.3%	15.8%

TABLE B.8 – Comparaison des algorithmes pour la base ”Ionosphere” (5% d’étiquettes connues).

100%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	100.0%	0.0%	100.0%	0.0%
	Non Reconnus	0.0%	0.0%	0.0%	0.0%

TABLE B.9 – Comparaison des algorithmes pour la base ”Ionosphere” (100% d’étiquettes connues).

”Wine”

2%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	92.7%	1.1%	91.0%	2.8%
	Non Reconnus	0.0%	6.2%	0.0%	6.2%

TABLE B.10 – Comparaison des algorithmes pour la base ”Wine” (2% d’étiquettes connues).

5%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	93.8%	2.8%	96.1%	0.5%
	Non Reconnus	0.0%	3.4%	0.0%	3.4%

TABLE B.11 – Comparaison des algorithmes pour la base ”Wine” (5% d’étiquettes connues).

100%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	100.0%	0.0%	96.9%	3.1%
	Non Reconnus	0.0%	0.0%	0.0%	0.0%

TABLE B.12 – Comparaison des algorithmes pour la base "Wine" (100% d'étiquettes connues).

"Glass2"

2%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	37.4%	7.0%	14.0%	30.4%
	Non Reconnus	5.6%	50.0%	26.0%	29.6%

TABLE B.13 – Comparaison des algorithmes pour la base "Glass2" (2% d'étiquettes connues).

5%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	21.0%	27.6%	31.9%	16.7%
	Non Reconnus	21.2%	30.2%	6.9%	44.5%

TABLE B.14 – Comparaison des algorithmes pour la base "Glass2" (5% d'étiquettes connues).

100%		**SL-$\overline{L}_2$**		**FCSC-θSP**	
		Reconnus	Non Reconnus	Reconnus	Non Reconnus
SSSC	Reconnus	100.0%	0.0%	43.0%	57.0%
	Non Reconnus	0.0%	0.0%	0.0%	0.0%

TABLE B.15 – Comparaison des algorithmes pour la base "Glass2" (100% d'étiquettes connues).

Annexe C

Matrices de confusions pour les données naturelles (chapitre 6)

Numéros de classe	Espèces	États
Classe 1	*Asterionella glacialis*	Cellules seules
Classe 2	*Asterionella glacialis*	Colonies
Classe 3	Cryptophycées	Inconnu
Classe 4	*Cylindrothetca closterium*	Inconnu
Classe 5	Dinoflagellés	Inconnu
Classe 6	*Lauderia annulata*	Cellules seules
Classe 7	*Skeletonema costatum*	Cellules seules
Classe 8	*Skeletonema costatum*	Colonies
Classe 9	*Thalassiosira rotula*	Cellules seules
Classe 10	*Thalassiosira rotula*	Colonies
Classe 11	*Guinardia delicatula*	Colonies
Classe 12	*Lauderia annulata*	Colonies

TABLE C.1 – Récapitulatif des espèces présentes dans la base de données naturelles.

Expérimentations sur la base "attributs"

K-moyennes sur la base "attributs" ($K = 9$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Classe 1	3	0	3	0	0	1	0	0	43
Classe 2	1	0	36	5	0	8	0	0	0
Classe 3	0	0	0	0	0	0	49	1	0
Classe 4	5	0	0	0	0	0	0	43	2
Classe 5	0	0	0	0	50	0	0	0	0
Classe 6	1	0	43	0	0	5	0	0	1
Classe 7	0	50	0	0	0	0	0	0	0
Classe 8	1	20	2	0	0	0	0	20	7
Classe 9	0	0	0	0	0	0	0	1	49
Classe 10	46	0	1	0	0	0	0	0	3
Classe 11	13	0	0	0	0	37	0	0	0
Classe 12	0	0	0	50	0	0	0	0	0

TABLE C.2 – Matrice de confusions pour l'algorithme des K-moyennes ($K = 9$) sur la base "attributs".

K-moyennes sur la base "attributs" ($K = 10$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Classe 1	1	1	0	1	0	5	0	42	0	0
Classe 2	35	10	0	0	0	0	0	0	0	5
Classe 3	0	0	0	0	49	0	1	0	0	0
Classe 4	0	0	2	2	0	1	43	2	0	0
Classe 5	0	0	0	0	0	0	0	0	50	0
Classe 6	37	11	0	0	0	0	0	2	0	0
Classe 7	0	0	50	0	0	0	0	0	0	0
Classe 8	0	0	19	1	0	5	21	4	0	0
Classe 9	0	0	0	0	0	47	0	3	0	0
Classe 10	0	1	0	45	0	2	0	2	0	0
Classe 11	0	37	0	0	0	0	0	13	0	0
Classe 12	0	0	0	0	0	0	0	0	0	50

TABLE C.3 – Matrice de confusions pour l'algorithme des K-moyennes ($K = 10$) sur la base "attributs".

K-moyennes sur la base ”attributs” ($K = 12$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Classe 1	43	1	0	1	0	0	4	0	0	0	1	0
Classe 2	0	24	3	1	0	0	0	0	0	0	0	22
Classe 3	0	0	0	0	1	0	0	49	0	0	0	0
Classe 4	2	0	0	0	11	1	0	0	0	34	2	0
Classe 5	0	0	0	0	0	0	0	0	50	0	0	0
Classe 6	2	30	0	0	0	0	0	0	0	0	0	18
Classe 7	0	0	0	0	1	49	0	0	0	0	0	0
Classe 8	4	1	0	0	15	18	4	0	0	7	1	0
Classe 9	3	0	0	0	0	0	47	0	0	0	0	0
Classe 10	2	0	0	1	0	0	1	0	0	0	46	0
Classe 11	13	0	0	37	0	0	0	0	0	0	0	0
Classe 12	0	0	50	0	0	0	0	0	0	0	0	0

TABLE C.4 – Matrice de confusions pour l’algorithme des K-moyennes ($K = 12$) sur la base ”attributs”.

K-moyennes sur la base ”attributs” ($K = 18$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
Classe 1	0	0	38	0	1	0	0	0	4	0	0	0	2	0	0	5	0	0
Classe 2	0	0	0	0	16	0	17	0	3	1	0	0	0	0	13	0	0	0
Classe 3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	21	28
Classe 4	0	0	0	0	0	1	0	0	0	0	32	0	4	13	0	0	0	0
Classe 5	0	23	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Classe 6	0	0	1	0	31	0	15	0	1	0	0	0	0	0	2	0	0	0
Classe 7	0	0	0	0	0	37	0	13	0	0	0	0	0	0	0	0	0	0
Classe 8	0	0	2	0	0	19	0	0	2	0	7	0	4	14	0	2	0	0
Classe 9	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0
Classe 10	40	0	1	0	0	0	0	0	7	0	0	0	0	0	0	2	0	0
Classe 11	0	37	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0
Classe 12	0	0	0	0	0	0	0	0	0	0	0	37	0	0	13	0	0	0

TABLE C.5 – Matrice de confusions pour l’algorithme des K-moyennes ($K = 18$) sur la base ”attributs”.

Classification hiérarchique sur la base ”attributs” ($K = 9$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Classe 1	1	1	3	45	0	0	0	0	0
Classe 2	0	47	2	0	0	0	0	1	0
Classe 3	0	0	0	0	0	1	0	0	49
Classe 4	0	0	0	4	0	45	1	0	0
Classe 5	0	0	0	0	50	0	0	0	0
Classe 6	0	47	1	2	0	0	0	0	0
Classe 7	0	0	0	0	0	1	49	0	0
Classe 8	0	0	2	9	0	24	15	0	0
Classe 9	0	0	0	50	0	0	0	0	0
Classe 10	0	0	44	6	0	0	0	0	0
Classe 11	37	0	13	0	0	0	0	0	0
Classe 12	0	0	0	0	0	0	0	50	0

TABLE C.6 – Matrice de confusions pour l’algorithme de classification hiérarchique ($K = 9$) sur la base ”attributs”.

Classification hiérarchique sur la base ”attributs” ($K = 10$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Classe 1	3	42	1	1	3	0	0	0	0	0
Classe 2	0	0	0	47	2	0	0	0	1	0
Classe 3	0	0	0	0	0	0	1	0	0	49
Classe 4	0	4	0	0	0	0	45	1	0	0
Classe 5	0	0	0	0	0	50	0	0	0	0
Classe 6	0	2	0	47	1	0	0	0	0	0
Classe 7	0	0	0	0	0	0	1	49	0	0
Classe 8	0	9	0	0	2	0	24	15	0	0
Classe 9	47	3	0	0	0	0	0	0	0	0
Classe 10	5	1	0	0	44	0	0	0	0	0
Classe 11	0	0	37	0	13	0	0	0	0	0
Classe 12	0	0	0	0	0	0	0	0	50	0

TABLE C.7 – Matrice de confusions pour l’algorithme de classification hiérarchique ($K = 10$) sur la base ”attributs”.

Classification hiérarchique sur la base ”attributs” ($K = 12$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Classe 1	0	1	0	3	3	42	1	0	0	0	0	0
Classe 2	16	31	0	2	0	0	0	0	0	0	1	0
Classe 3	0	0	0	0	0	0	0	0	1	0	0	49
Classe 4	0	0	0	0	0	4	0	0	45	1	0	0
Classe 5	0	0	0	0	0	0	0	50	0	0	0	0
Classe 6	19	28	0	1	0	2	0	0	0	0	0	0
Classe 7	0	0	0	0	0	0	0	0	1	49	0	0
Classe 8	0	0	0	2	0	9	0	0	24	15	0	0
Classe 9	0	0	0	0	47	3	0	0	0	0	0	0
Classe 10	0	0	39	5	5	1	0	0	0	0	0	0
Classe 11	0	0	0	13	0	0	37	0	0	0	0	0
Classe 12	0	0	0	0	0	0	0	0	0	0	50	0

TABLE C.8 – Matrice de confusions pour l’algorithme de classification hiérarchique ($K = 12$) sur la base ”attributs”.

Classification hiérarchique sur la base ”attributs” ($K = 18$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
Classe 1	0	0	0	0	0	0	1	0	0	42	0	0	0	0	3	3	1	0
Classe 2	1	0	0	0	0	0	18	13	0	0	0	0	16	0	2	0	0	0
Classe 3	0	0	23	26	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Classe 4	0	0	0	0	0	1	0	0	4	0	34	11	0	0	0	0	0	0
Classe 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
Classe 6	0	0	0	0	0	0	27	1	0	2	0	0	19	0	1	0	0	0
Classe 7	0	0	0	0	12	37	0	0	0	0	0	1	0	0	0	0	0	0
Classe 8	0	0	0	0	0	15	0	0	7	2	5	19	0	0	2	0	0	0
Classe 9	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	47	0	0
Classe 10	0	0	0	0	0	0	0	0	0	1	0	0	0	39	5	5	0	0
Classe 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	37	0
Classe 12	31	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE C.9 – Matrice de confusions pour l’algorithme de classification hiérarchique ($K = 18$) sur la base ”attributs”.

Classification spectrale sur la base ”attributs” ($K = 9$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Classe 1	5	44	0	0	0	0	1	0	0
Classe 2	45	0	0	0	2	0	3	0	0
Classe 3	0	0	1	49	0	0	0	0	0
Classe 4	0	4	44	0	0	0	0	2	0
Classe 5	0	0	0	0	0	0	0	0	50
Classe 6	49	1	0	0	0	0	0	0	0
Classe 7	0	0	0	0	0	0	0	50	0
Classe 8	2	7	20	0	0	1	0	20	0
Classe 9	0	50	0	0	0	0	0	0	0
Classe 10	1	5	0	0	0	44	0	0	0
Classe 11	13	0	0	0	0	0	37	0	0
Classe 12	0	0	0	0	50	0	0	0	0

TABLE C.10 – Matrice de confusions pour l’algorithme de classification spectrale ($K = 9$) sur la base ”attributs”.

Classification spectrale sur la base ”attributs” ($K = 10$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Classe 1	42	1	0	1	0	6	0	0	0	0
Classe 2	2	3	0	43	2	0	0	0	0	0
Classe 3	0	0	0	0	0	0	0	1	0	49
Classe 4	4	0	2	0	0	0	0	44	0	0
Classe 5	0	0	0	0	0	0	50	0	0	0
Classe 6	2	0	0	48	0	0	0	0	0	0
Classe 7	0	0	50	0	0	0	0	0	0	0
Classe 8	4	0	20	0	0	5	0	20	1	0
Classe 9	3	0	0	0	0	47	0	0	0	0
Classe 10	2	1	0	0	0	3	0	0	44	0
Classe 11	13	37	0	0	0	0	0	0	0	0
Classe 12	0	0	0	0	50	0	0	0	0	0

TABLE C.11 – Matrice de confusions pour l’algorithme de classification spectrale ($K = 10$) sur la base ”attributs”.

Classification spectrale sur la base ”attributs” ($K = 12$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Classe 1	3	0	1	0	3	0	1	0	0	0	42	0
Classe 2	2	0	2	0	0	0	42	0	3	1	0	0
Classe 3	0	1	0	0	0	0	0	49	0	0	0	0
Classe 4	0	44	0	0	0	0	0	0	0	0	4	2
Classe 5	0	0	0	50	0	0	0	0	0	0	0	0
Classe 6	1	0	0	0	0	0	47	0	1	0	1	0
Classe 7	0	0	0	0	0	0	0	0	0	0	0	50
Classe 8	1	20	0	0	1	1	0	0	0	0	7	20
Classe 9	0	0	0	0	45	0	0	0	0	0	5	0
Classe 10	1	0	1	0	3	44	0	0	0	0	1	0
Classe 11	13	0	37	0	0	0	0	0	0	0	0	0
Classe 12	0	0	0	0	0	0	0	0	31	19	0	0

TABLE C.12 – Matrice de confusions pour l’algorithme de classification spectrale ($K = 12$) sur la base ”attributs”.

Classification spectrale sur la base ”attributs” ($K = 18$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
Classe 1	0	0	0	0	0	1	0	0	43	0	3	1	0	0	2	0	0	0
Classe 2	0	0	11	22	0	15	0	0	0	0	0	1	0	0	0	0	1	0
Classe 3	0	0	0	0	0	0	0	0	0	26	0	0	23	1	0	0	0	0
Classe 4	0	0	0	0	0	0	33	0	4	0	0	0	0	12	0	1	0	0
Classe 5	0	23	0	0	0	0	0	27	0	0	0	0	0	0	0	0	0	0
Classe 6	0	0	25	4	0	19	0	0	1	0	1	0	0	0	0	0	0	0
Classe 7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	21	0	28
Classe 8	1	0	0	0	0	0	8	0	7	0	1	0	0	15	0	16	0	2
Classe 9	0	0	0	0	0	0	0	0	7	0	0	0	0	0	43	0	0	0
Classe 10	44	0	0	0	0	0	0	0	2	0	1	1	0	0	2	0	0	0
Classe 11	0	0	0	0	12	0	0	0	0	0	13	25	0	0	0	0	0	0
Classe 12	0	0	0	7	0	6	0	0	0	0	0	0	0	0	0	0	37	0

TABLE C.13 – Matrice de confusions pour l’algorithme de classification spectrale ($K = 18$) sur la base ”attributs”.

Expérimentations sur la base "similarités"

K-moyennes sur la base "similarités" ($K = 9$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Classe 1	0	0	0	44	2	4	0	0	0
Classe 2	0	0	20	1	0	16	0	13	0
Classe 3	0	0	0	0	0	0	0	0	50
Classe 4	2	0	0	3	1	0	44	0	0
Classe 5	0	50	0	0	0	0	0	0	0
Classe 6	0	0	11	3	0	0	0	36	0
Classe 7	49	0	0	0	0	0	1	0	0
Classe 8	27	0	0	4	5	0	14	0	0
Classe 9	0	0	0	4	40	1	0	0	5
Classe 10	0	0	0	1	48	1	0	0	0
Classe 11	0	0	0	0	0	50	0	0	0
Classe 12	0	0	7	0	0	0	0	43	0

TABLE C.14 – Matrice de confusions pour l'algorithme de K-moyennes ($K = 9$) sur la base "similarités".

K-moyennes sur la base "similarités" ($K = 10$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Classe 1	0	43	1	4	0	1	1	0	0	0
Classe 2	0	1	0	17	9	23	0	0	0	0
Classe 3	0	0	0	0	0	0	0	0	50	0
Classe 4	0	3	0	0	0	0	2	43	0	2
Classe 5	50	0	0	0	0	0	0	0	0	0
Classe 6	0	3	0	0	30	17	0	0	0	0
Classe 7	0	0	0	0	0	0	0	1	0	49
Classe 8	0	4	1	0	0	0	4	15	0	26
Classe 9	0	4	43	1	0	0	2	0	0	0
Classe 10	0	1	6	1	0	0	42	0	0	0
Classe 11	0	0	0	50	0	0	0	0	0	0
Classe 12	0	0	0	0	50	0	0	0	0	0

TABLE C.15 – Matrice de confusions pour l'algorithme de K-moyennes ($K = 10$) sur la base "similarités".

K-moyennes sur la base ”similarités” ($K = 12$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Classe 1	0	3	1	0	0	0	3	0	1	0	42	0
Classe 2	0	10	0	13	0	11	15	0	0	0	1	0
Classe 3	0	0	0	0	0	0	0	0	0	50	0	0
Classe 4	14	0	0	0	31	0	0	0	0	0	3	2
Classe 5	0	0	0	0	0	0	0	50	0	0	0	0
Classe 6	0	0	0	5	0	20	22	0	0	0	3	0
Classe 7	0	0	0	0	1	0	0	0	0	0	0	49
Classe 8	7	0	0	0	15	0	0	0	0	0	4	24
Classe 9	0	1	2	0	0	0	0	0	43	0	4	0
Classe 10	1	1	42	0	0	0	0	0	5	0	1	0
Classe 11	0	50	0	0	0	0	0	0	0	0	0	0
Classe 12	0	0	0	20	0	30	0	0	0	0	0	0

TABLE C.16 – Matrice de confusions pour l’algorithme de K-moyennes ($K = 12$) sur la base ”similarités”.

K-moyennes sur la base ”similarités” ($K = 18$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
Classe 1	0	0	26	0	0	0	21	0	1	0	0	2	0	0	0	0	0	0
Classe 2	0	0	0	0	0	0	3	16	9	17	0	0	0	0	0	5	0	0
Classe 3	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Classe 4	0	0	0	21	5	20	2	0	0	0	2	0	0	0	0	0	0	0
Classe 5	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0
Classe 6	0	0	1	0	0	0	2	33	0	4	0	0	0	0	0	10	0	0
Classe 7	0	0	0	1	0	0	0	0	0	0	13	0	36	0	0	0	0	0
Classe 8	0	0	1	11	6	3	2	0	0	0	21	0	6	0	0	0	0	0
Classe 9	0	0	0	0	2	0	4	0	0	0	0	2	0	0	25	0	13	4
Classe 10	27	0	0	0	2	0	2	0	0	0	0	18	0	0	1	0	0	0
Classe 11	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0
Classe 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0

TABLE C.17 – Matrice de confusions pour l’algorithme des K-moyennes ($K = 18$) sur la base ”similarités”.

Classification hiérarchique sur la base "similarités" ($K = 9$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Classe 1	1	10	0	0	1	37	0	0	1
Classe 2	0	0	0	0	7	3	0	0	40
Classe 3	0	0	0	0	0	0	49	1	0
Classe 4	0	5	0	2	0	0	0	43	0
Classe 5	0	0	50	0	0	0	0	0	0
Classe 6	0	1	0	0	0	2	0	0	47
Classe 7	0	0	0	49	0	0	0	1	0
Classe 8	0	7	0	22	0	2	0	19	0
Classe 9	39	7	0	0	0	4	0	0	0
Classe 10	2	46	0	0	0	2	0	0	0
Classe 11	0	0	0	0	50	0	0	0	0
Classe 12	0	0	0	0	0	0	0	0	50

TABLE C.18 – Matrice de confusions pour l'algorithme de classification hiérarchique ($K = 9$) sur la base "similarités".

Classification hiérarchique sur la base "similarités" ($K = 10$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Classe 1	1	0	1	10	0	0	1	37	0	0
Classe 2	27	13	0	0	0	0	7	3	0	0
Classe 3	0	0	0	0	0	0	0	0	49	1
Classe 4	0	0	0	5	0	2	0	0	0	43
Classe 5	0	0	0	0	50	0	0	0	0	0
Classe 6	8	39	0	1	0	0	0	2	0	0
Classe 7	0	0	0	0	0	49	0	0	0	1
Classe 8	0	0	0	7	0	22	0	2	0	19
Classe 9	0	0	39	7	0	0	0	4	0	0
Classe 10	0	0	2	46	0	0	0	2	0	0
Classe 11	0	0	0	0	0	0	50	0	0	0
Classe 12	0	50	0	0	0	0	0	0	0	0

TABLE C.19 – Matrice de confusions pour l'algorithme de classification hiérarchique ($K = 10$) sur la base "similarités".

Classification hiérarchique sur la base ”similarités” ($K = 12$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Classe 1	0	0	10	0	1	0	1	0	0	1	37	0
Classe 2	0	0	0	0	27	13	0	0	0	7	3	0
Classe 3	0	1	0	0	0	0	0	0	0	0	0	49
Classe 4	22	21	5	0	0	0	0	0	2	0	0	0
Classe 5	0	0	0	0	0	0	0	50	0	0	0	0
Classe 6	0	0	1	0	39	8	0	0	0	0	2	0
Classe 7	0	1	0	0	0	0	0	0	49	0	0	0
Classe 8	2	17	7	0	0	0	0	0	22	0	2	0
Classe 9	0	0	2	5	0	0	39	0	0	0	4	0
Classe 10	0	0	4	42	0	0	2	0	0	0	2	0
Classe 11	0	0	0	0	0	0	0	0	0	50	0	0
Classe 12	0	0	0	0	0	50	0	0	0	0	0	0

TABLE C.20 – Matrice de confusions pour l’algorithme de classification hiérarchique ($K = 12$) sur la base ”similarités”.

Classification hiérarchique sur la base ”similarités” ($K = 18$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
Classe 1	25	12	1	0	0	0	1	0	0	0	0	0	0	0	10	0	1	0
Classe 2	0	3	0	0	0	0	3	24	0	0	13	0	0	0	0	0	7	0
Classe 3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	49
Classe 4	0	0	0	0	0	2	0	0	0	0	0	0	22	21	5	0	0	0
Classe 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0
Classe 6	0	2	0	0	0	0	9	30	0	0	1	7	0	0	1	0	0	0
Classe 7	0	0	0	0	17	32	0	0	0	0	0	0	0	1	0	0	0	0
Classe 8	1	1	0	0	1	21	0	0	0	0	0	0	2	17	7	0	0	0
Classe 9	0	4	11	28	0	0	0	0	0	5	0	0	0	0	2	0	0	0
Classe 10	0	2	1	1	0	0	0	0	20	22	0	0	0	0	4	0	0	0
Classe 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
Classe 12	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0

TABLE C.21 – Matrice de confusions pour l’algorithme de classification hiérarchique ($K = 18$) sur la base ”similarités”.

Classification spectrale sur la base "similarités" ($K = 9$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
Classe 1	0	0	7	0	1	16	21	5	0
Classe 2	2	0	3	0	27	0	18	0	0
Classe 3	0	0	0	1	0	0	0	0	49
Classe 4	0	0	0	7	1	0	0	42	0
Classe 5	0	50	0	0	0	0	0	0	0
Classe 6	0	0	0	0	1	1	48	0	0
Classe 7	0	0	0	50	0	0	0	0	0
Classe 8	0	0	0	24	0	1	1	24	0
Classe 9	0	0	5	0	0	45	0	0	0
Classe 10	38	0	1	0	0	9	1	1	0
Classe 11	0	0	50	0	0	0	0	0	0
Classe 12	0	0	0	0	50	0	0	0	0

TABLE C.22 – Matrice de confusions pour l'algorithme de classification spectrale ($K = 9$) sur la base "similarités".

Classification spectrale sur la base "similarités" ($K = 10$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Classe 1	0	0	0	0	3	0	1	0	0	46
Classe 2	15	2	0	0	2	0	26	0	0	5
Classe 3	0	0	0	0	0	1	0	0	49	0
Classe 4	0	0	44	0	0	5	1	0	0	0
Classe 5	0	0	0	50	0	0	0	0	0	0
Classe 6	47	0	0	0	0	0	1	1	0	1
Classe 7	0	0	0	0	0	50	0	0	0	0
Classe 8	0	0	22	0	0	24	0	0	0	4
Classe 9	0	0	0	0	5	0	0	44	0	1
Classe 10	1	36	1	0	1	0	0	11	0	0
Classe 11	0	0	0	0	50	0	0	0	0	0
Classe 12	0	0	0	0	0	0	50	0	0	0

TABLE C.23 – Matrice de confusions pour l'algorithme de classification spectrale ($K = 10$) sur la base "similarités".

Classification spectrale sur la base ”similarités” ($K = 12$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Classe 1	0	0	43	0	0	1	1	0	0	0	2	3
Classe 2	0	0	3	0	0	6	0	0	0	0	41	0
Classe 3	0	0	0	0	0	0	0	0	49	1	0	0
Classe 4	32	0	0	0	0	0	0	0	0	2	5	11
Classe 5	0	0	0	0	0	0	0	50	0	0	0	0
Classe 6	0	1	1	1	0	47	0	0	0	0	0	0
Classe 7	0	0	0	0	0	0	0	0	0	50	0	0
Classe 8	4	0	3	0	0	0	0	0	0	22	1	20
Classe 9	0	0	6	44	0	0	0	0	0	0	0	0
Classe 10	0	0	1	11	36	1	0	0	0	0	0	1
Classe 11	0	0	13	0	0	0	25	0	0	0	12	0
Classe 12	0	50	0	0	0	0	0	0	0	0	0	0

TABLE C.24 – Matrice de confusions pour l’algorithme de classification spectrale ($K = 12$) sur la base ”similarités”.

Classification spectrale sur la base ”similarités” ($K = 18$) :

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
Classe 1	0	1	25	2	0	0	0	0	0	22	0	0	0	0	0	0	0	0
Classe 2	0	43	1	0	0	0	0	0	0	5	0	0	1	0	0	0	0	0
Classe 3	0	0	0	0	1	0	0	0	0	0	49	0	0	0	0	0	0	0
Classe 4	0	0	1	0	0	0	0	38	0	0	0	0	11	0	0	0	0	0
Classe 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0
Classe 6	0	0	1	0	0	0	0	0	0	0	0	1	0	0	47	1	0	0
Classe 7	0	0	0	0	34	0	0	0	0	0	0	0	1	0	0	0	0	15
Classe 8	0	0	3	0	2	0	0	4	0	4	0	0	20	0	0	0	0	17
Classe 9	0	0	0	5	0	0	15	0	0	0	0	30	0	0	0	0	0	0
Classe 10	37	0	1	1	0	0	1	0	0	0	0	9	0	0	1	0	0	0
Classe 11	0	0	13	12	0	0	0	0	13	0	0	0	0	12	0	0	0	0
Classe 12	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	20	0	0

TABLE C.25 – Matrice de confusions pour l’algorithme de classification spectrale ($K = 18$) sur la base ”similarités”.

Liste des tableaux

Table des figures

Bibliographie

[Bach and Jordan, 2004] Bach, F. and Jordan, M. (2004). Learning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 305–321.

[Basu et al., 2004] Basu, S., Bilenko, M., and Mooney, R. (2004). A probabilistic framework for semi-supervised clustering. In *KDD, International Conference on Knowledge Discovery and Data Mining*, pages 59–68.

[Belkin and Niyogi, 2003] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. In *Journal : Neural Computation*, pages 1373–1396.

[Bradley et al., 2000] Bradley, P., Bennett, K., and Demiriz, A. (2000). Constrained k-means clustering. In *Technical Report*.

[Brand and Huang, 2003] Brand, M. and Huang, K. (2003). A unifying theorem for spectral embedding and clustering. In *ICAIS, Proceedings of the 9th International Conference on Artificial Intelligence and Statistics*.

[Buskey and Hyatta, 2006] Buskey, E. and Hyatta, C. (2006). Use of the flowcam for semi-automated recognition and enumeration of red tide cells (karenia brevis) in natural plankton samples. In *Harmful Algae, vol. 5*, pages 685–692.

[Caillault et al., 2009] Caillault, E., Hébert, P., and Wacquet, G. (2009). Dissimilarity-based classification of multidimensional signals by conjoint elastic matching : Application to phytoplanktonic species recognition. In *EANN, 11th International Conference of Engineering Applications of Neural Networks*, pages 153–164.

[Cevikalp and Verbeek, 2008] Cevikalp, H. and Verbeek, J. (2008). Semi-supervised dimensionality reduction using pairwise equivalence constraints. In *International Conference on Computer Vision Theory and Applications*, pages 489–496.

[Chapelle et al., 2006] Chapelle, O., Scholkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. MIT Press, Cambridge.

[Chen et al., 2009] Chen, C., Zhang, L., Bu, J., Wang, C., and Chen, W. (2009). Constrained

laplacian eigenmap for dimensionality reduction. In *Journal : Neurocomputing*, pages 951–958.

[Chen et al., 2005] Chen, Y., Zhang, Y., and Ji, X. (2005). Size regularized cut for data clustering. In *Advances in Neural Information Processing Systems*.

[Chung, 1997] Chung, F. (1997). Spectral graph theory. In *Conference Board of the Mathematical Sciences*.

[Cloern, 1996] Cloern, J. (1996). Phytoplankton bloom dynamics in coastal ecosystems : a review with some general lessons from sustained investigation of san francisco bay, california. In *Reviews of Geophysics, vol. 34*.

[Cohn et al., 2003] Cohn, D., Caruana, R., and Mccallum, A. (2003). Semi-supervised clustering with user feedback. In *Technical Report*.

[Collins and Krzanowski, 2002] Collins, G. and Krzanowski, W. (2002). Non-parametric discriminant analysis of phytoplankton species using data from analytical flow cytometry. In *Wiley InterScience*, pages 26–33.

[Davidson et al., 2006] Davidson, I., Wagstaff, K., and Basu, S. (2006). Measuring constraint-set utility for partitional clustering algorithms. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 115–126.

[Deerwester et al., 1990] Deerwester, S., Dumais, S., Landauer, T., G.Furnas, and Harshman, R. (1990). Indexing by latent semantic analysis. In *Journal of the American Society of Information Science*, pages 391–407.

[Demiriz et al., 1999] Demiriz, A., Bennett, K., and Embrechts, M. (1999). Semi-supervised clustering using genetic algorithms. In *Artificial Neural Networks in Engineering*, pages 809–814.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. In *Journal of the Royal Statistical Society : Series B*, pages 1–38.

[Dhillon, 2004] Dhillon, I. (2004). Kernel k-means, spectral clustering and normalized cuts. In *Proceedings of the International Conference on Knowledge discovery and Data mining*, pages 551–556.

[Ding et al., 2001] Ding, C., He, X., Zha, H., Gu, M., and Simon, H. (2001). A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of the International Conference on Data Mining*, pages 107–114.

[Dir00, 2000] Dir00 (2000). Directive 2000/60/ec of the european parliament and of the council of 23 october 2000 establishing a framework for community action in the field of water policy. In *Official Journal of the European Communities EN, vol. 2000/60/EC*.

[Dubelaar and Jonker, 2000] Dubelaar, G. and Jonker, R. (2000). Flow cytometry as a tool for the study of phytoplankton. In *Scientia Marina*, pages 135–156.

[Fortunato, 2010] Fortunato, S. (2010). Finding and evaluating community structure in networks. In *Physics Reports 486*, pages 75–174.

[Ge et al., 2007] Ge, R., Ester, M., Jin, W., and Davidson, I. (2007). Constraint driven clustering. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 320–329.

[Gregori, 2001] Gregori, G. (2001). Ultraplancton dans la baie de marseille : Séries temporelles, viabilité bactérienne et mesure de la respiration par cytométrie en flux. In *Thèses, Université de la Méditerranée de Marseille*.

[Guiselin, 2010] Guiselin, N. (2010). Etude de la dynamique des communautés phytoplanctoniques par microscopie et cytométrie en flux, en eaux côtières de la manche orientale. In *Thèse, Université du Littoral Côte d'Opale*.

[Hagen and Kahng, 1992] Hagen, L. and Kahng, A. (1992). New spectral methods for ratio cut partitioning and clustering. In *IEEE Computer Aided Design*, pages 1074–1085.

[Hamad and Biela, 2008] Hamad, D. and Biela, P. (2008). Introduction to spectral clustering. In *Information and Communication Technologies : From Theory to Applications*, pages 1–6.

[Han and Kamber, 2006] Han, J. and Kamber, M. (2006). *Data Mining : Concepts and Techniques*. Morgan Kaufmann Publishers.

[He and Niyogi, 2002] He, X. and Niyogi, P. (2002). Locality preserving projections. In *Computer and Information Science*, pages 153–160.

[Jain and Farrokhnia, 1991] Jain, A. and Farrokhnia, F. (1991). Unsupervised texture segmentation using gabor filters. In *Pattern Recognition*, pages 1167–1186.

[Jain et al., 1999] Jain, A., Murty, M., and Flynn, P. (1999). Data clustering : a review. In *ACM Computing Surveys*.

[Ji and Xu, 2006] Ji, X. and Xu, W. (2006). Document clustering with prior knowledge. In *SIGIR, International Conference on Research and Development in Information Retrieval*, pages 405–412.

[Kamvar et al., 2003] Kamvar, S., Klein, D., and Manning, C. (2003). Spectral learning. In *IJCAI, International Joint Conference on Artificial Intelligence*, pages 561–566.

[Karush, 1939] Karush, W. (1939). Minima of functions of several variables with inequalities as side conditions. In *Thesis, Department of Mathematics, University of Chicago*.

[Kuhn and Tucker, 1982] Kuhn, H. and Tucker, A. (1982). Non linear programming. In *ACM SIGMAP Bulletin*, pages 6–18.

[Kunegis et al., 2008] Kunegis, J., Lommatzsch, A., and Dai-Labor, C. (2008). Alternative similarity functions for graph kernels. In *International Conference on Pattern Recognition*, pages 1–4.

[Lang, 2006] Lang, K. (2006). Fixing two weaknesses of the spectral method. In *Advances in Neural Information Processing Systems*, pages 715–722.

[Lepage, 2004] Lepage, R. (2004). Reconnaissance d'algues toxiques par vision artificielle et réseau de neurones. In *Thesis, Université du Québec à Rimouski*.

[Li et al., 2009] Li, Z., Liu, J., and Tang, X. (2009). Constrained clustering via spectral regularization. In *International Conference on Computer Vision and Pattern Recognition*, pages 421–428.

[Losee, 1998] Losee, R. M. (1998). *Text retrieval and filtering analytic models of performance*. Kluwer Academic Publishers.

[Lund et al., 1958] Lund, J., Kipling, G., and Cren, E. (1958). The inverted microscope method of estimating algal numbers and the statistical basis of estimation by counting. In *Hydrobiologia*, pages 143–170.

[Luxburg, 2007] Luxburg, U. (2007). A tutorial on spectral clustering. In *Statistics and Computing*, pages 395–416.

[Malkassian et al., 2011] Malkassian, A., Nerini, D., van Dijk, M., Thyssen, M., Mante, C., and Gregori, G. (2011). Functional analysis and classification of phytoplankton based on data from an automated flow cytometer. In *Cytometry part A*, pages 263–275.

[Meila and Shi, 2000] Meila, M. and Shi, J. (2000). Learning segmentation by random walks. In *NIPS12, Neural Information Processing Systems*, pages 873–879.

[Mohar, 1997] Mohar, B. (1997). Some applications of laplace eigenvalues of graphs. In *Graph symetry : algebraic methods and applications*, pages 225–275.

[Murtagh, 1985] Murtagh, F. (1985). A survey of algorithms for contiguity-constrained clustering and related problems. In *The computer Journal*, pages 82–88.

[Newman and Girvan, 2004] Newman, M. and Girvan, M. (2004). Finding and evaluating community structure in networks. In *Physical Review E*.

[Ng et al., 2002] Ng, A., Jordan, M., and Weiss, Y. (2002). On spectral clustering : Analysis and an algorithm. In *NIPS14, Neural Information Processing Systems*, pages 849–856.

[Oliver and Webster, 1989] Oliver, M. and Webster, R. (1989). A geostatistical basis for spatial weighting in multivariate classification. In *Mathematical Geology*, pages 15–35.

[Perona and Freeman, 1998] Perona, P. and Freeman, W. (1998). A factorization approach to grouping. In *European Conference on Computer Vision*, pages 655–670.

[Rand, 2007] Rand, W. (2007). Objective criteria for the evaluation of clustering methods. In *Journal of the American Statistical Association*, pages 395–416.

[Roberts et al., 1996] Roberts, S., Gisler, G., and Theiler, J. (1996). Spatio-spectral image analysis using classical and neural algorithms. In *Intelligent Engineering Systems Through Artificial Neural Networks*, pages 425–430.

[Sakoe and Chiba, 1978] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 43–49.

[Salton, 1989] Salton, G. (1989). *Automatic Text Processing : the transfomation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing.

[Salton and Buckley, 1988] Salton, G. and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523.

[Sanguinetti et al., 2005] Sanguinetti, G., Laidler, J., and Lawrence, N. (2005). Automatic determination of the number of clusters using spectral algorithms. In *IEEE Machine Learning for Signal Processing conference*.

[Shapiro, 2003] Shapiro, H. (2003). *Practical Flow Cytometry*. 4th edition Wiley-Liss Inc.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. In *PAMI, Transactions on Pattern Analysis and Machine Intelligence*, pages 888–905.

[Shortreed, 2006] Shortreed, S. (2006). Learning in spectral clustering. In *Thesis, University of Washington*.

[Shortreed and Meila, 2005] Shortreed, S. and Meila, M. (2005). Unsupervised spectral learning. In *Proceedings of the Twenty-First Conference Annual on Uncertainty in Artificial Intelligence*, pages 534–541.

[Sieracki, 2005] Sieracki, M. (2005). Automated recognition of phytoplankton morphotypes from flowcam images. In *ZooImage, GLOBEC / SPACC workshop on image analysis to count and identify zooplankton*.

[Smayda, 1990] Smayda, T. (1990). Novel and nuisance phytoplankton blooms in the sea : Evidence for a global epidemic. In *Proceedings of the 4th International Conference on Toxic Marine Phytoplankton*, pages 29–40.

[Takabayashi et al., 2006] Takabayashi, M., Lew, K., Johnson, A., Marchi, A., and Dugdale, A. (2006). The effect of nutrient availability and temperature on chain length of the diatom, skeletonema costatum. In *Journal of Plankton Research*, pages 831–840.

[Tang and Zhong, 2007] Tang, W. and Zhong, S. (2007). *Pairwise Constraints-Guided Dimensionality Reduction*. Computational Methods of Feature Selection, Eds. Huan Liu and Hiroshi Motoda.

[Thyssen and Denis, 2008] Thyssen, M. and Denis, M. (2008). Analyse in situ haute fréquence du phytoplancton : vers la mise au point d'un bio indicateur des variations environnementales. In *Laboratoire de Microbiologie, Géochimie et Ecologie Marines, Université de la Méditerranée de Marseille*.

[Wacquet, 2010] Wacquet, G. (2010). Automatic classification of phytoplanktonic cells. In *Workshop on data analysis in Flow Cytometry, Delft, The Netherlands*.

[Wacquet et al., 2011a] Wacquet, G., Hamad, D., and Artigas, L. (2011a). Classification spectrale semi-supervisée. application à la surveillance de l'écosystème marin. In *Workshop du Groupement d'Intérêt Scientifique, Surveillance, Sûreté et Sécurité des Grands Systèmes, GIS 3SGS, Valenciennes, France*.

[Wacquet et al., 2011b] Wacquet, G., Hébert, P., Caillault, E., and Hamad, D. (2011b). Classification semi-supervisée pour l'identification de cellules phytoplanctoniques. In *STIC et Environnement, Colloque Sciences et Techniques de l'Information et de la Communication Pour l'Environnement*.

[Wacquet et al., 2011c] Wacquet, G., Hébert, P., Caillault, E., and Hamad, D. (2011c). Semi-supervised k-way spectral clustering using pairwise constraints. In *International Conference on Neural Computation Theory and Applications*.

[Wacquet et al., 2008] Wacquet, G., Hébert, P., Caillault, E., Hamad, D., and Artigas, L. (2008). Classification de signaux issus de mesures cytomètriques. In *Rapport de stage de deuxième année de Master INS3I*, page 47.

[Wagstaff, 2002] Wagstaff, K. (2002). Intelligent clustering with instance-level constraints. In *Thesis, Faculty of the Graduate School of Cornell University*.

[Wagstaff and Cardie, 2000] Wagstaff, K. and Cardie, C. (2000). Clustering with instance-level constraints. In *ICML, International Conference on Machine Learning*, pages 1103–1110.

[Wagstaff et al., 2001] Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings of 18th International Conference on Machine Learning*, pages 577–584.

[Wan and Eick, 1998] Wan, T. and Eick, C. (1998). Similarity measures for multi-valued attributes for database clustering. In *Conference on Smart Engineering System Design : Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining and Rough Sets*, pages 1–4.

[Wang and Davidson, 2010] Wang, X. and Davidson, I. (2010). Flexible constrained spectral clustering. In *KDD, International Conference on Knowledge Discovery and Data Mining*, pages 563–572.

[Weiss, 1999] Weiss, Y. (1999). Segmentation using eigenvectors : an unifying view. In *IEEE, International Conference on Computer Vision*, pages 975–982.

[White and Smyth, 2005] White, S. and Smyth, P. (2005). A spectral clustering approach to finding communities in graphs. In *SIAM, International Conference on Data Mining*.

[Xiang and Gong, 2008] Xiang, T. and Gong, S. (2008). Spectral clustering with eigenvector selection. In *Pattern Recognition*, pages 1012–1029.

[Xu et al., 2005] Xu, Q., desJardins, M., and Wagstaff, K. (2005). Constrained spectral clustering under a local proximity structure assumption. In *Proceedings of the International Conference of the Florida Artificial Intelligence Research Society*, pages 866–867.

[Yu et al., 2010] Yu, G., Peng, H., Wei, J., and Ma, Q. (2010). Robust locality preserving projections with pairwise constraints. In *Joint Symposium on Information Systems*, pages 1631–1636.

[Zapien, 2009] Zapien, K. (2009). Algorithme de chemin de régularisation pour l'apprentissage statistique. In *Thèse, Institut National des Sciences Appliquées de Rouen*.

[Zelnik-Manor and Perona, 2004] Zelnik-Manor, L. and Perona, P. (2004). Self tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608.

[Zhang et al., 2007] Zhang, D., Zhou, Z., and Chen, S. (2007). Semi-supervised dimensionality reduction. In *SIAM, 7th International Conference on Data Mining*, pages 629–634.

Liste des publications liées à la thèse

Conférences internationales avec comités de lecture et publication des actes

- Wacquet, G., Hébert, P.-A., Caillault-Poisson, É., and Hamad, D., **Semi-Supervised K-Way Spectral Clustering using Pairwise Constraints**. *International Conference on Neural Computation Theory and Applications, NCTA*, Paris, October 24-26, 2011.
- Caillault, É., Hébert, P.-A., and Wacquet, G., **Dissimilarity-based classification of multidimensional signals by conjoint elastic matching : Application to phytoplanktonic species recognition**. *Engineering Applications of Neural Networks*, London, United Kingdom, pages 153-164, August 27-29, 2009 (*www.springerlink.com/content/h*721763602*k*9733*l*/).

Conférence nationale avec comité de lecture et publication des actes

- Wacquet, G., Hébert, P.-A., Caillault-Poisson, É., et Hamad, D., **Classification Semi-Supervisée pour l'Identification de Cellules Phytoplanctoniques**. *STIC et Environnement, Colloque Sciences et Techniques de l'Information et de la Communication Pour l'Environnement*, Saint-Étienne, pages 15-28, 11-13 mai, 2011. Prix Jeune Chercheur (*www.mines – paristech.fr/Fr/Services/PressesENSMP/Resumes/STIC*$_1$*res.pdf*).

Workshops avec poster et présentation orale

- Wacquet, G., Hamad, D., et Artigas, L.F., **Classification spectrale semi-supervisée. Application à la surveillance de l'écosystème marin**. *4ème Workshop du Groupement d'Intérêt Scientifique, Surveillance, Sûreté et Sécurité des Grands Systèmes, GIS 3SGS*, Valenciennes, 12-13 octobre, 2011.
- Caillault-Poisson, É., and Wacquet, G., **Data analysis**. *DYMAPHY Project*, CEFAS, Lowerstoft, United Kingdom, March 23, 2011.
- Wacquet, G., **Automatic Classification of Phytoplanktonic Cells**. *Workshop on data analysis in Flow Cytometry* , Delft, The Netherlands, March 23-24, 2010.

Rapport de stage de Master INS3I

- Wacquet, G., **Classification de signaux issus de mesures cytomètriques**. *Rapport de stage de deuxième année de Master INS3I*, Université du Littoral Côte d'Opale, Calais, 47 pages, 17 mars-13 septembre, 2008.

Résumé

Dans les systèmes d'aide à la décision, sont généralement à disposition des données numériques abondantes et éventuellement certaines connaissances contextuelles qualitatives, disponibles *a priori* ou fournies *a posteriori* par retour d'expérience. Les performances des approches de classification, en particulier spectrale, dépendent de l'intégration de ces connaissances dans leur conception. Les algorithmes de classification spectrale permettent de traiter la classification sous l'angle de coupes de graphe. Ils classent les données dans l'espace des vecteurs propres de la matrice Laplacienne du graphe. Cet espace est censé mieux révéler la présence de groupements naturels linéairement séparables.

Dans ce travail, nous nous intéressons aux algorithmes intégrant des connaissances type contraintes de comparaison. L'espace spectral doit, dans ce cas, révéler la structuration en classes tout en respectant, autant que possible, les contraintes de comparaison. Nous présentons un état de l'art des approches spectrales semi-supervisées contraintes. Nous proposons un nouvel algorithme qui permet de générer un sous-espace de projection par optimisation d'un critère de multi-coupes normalisé avec ajustement des coefficients de pénalité dus aux contraintes. Les performances de l'algorithme sont mises en évidence sur différentes bases de données par comparaison à d'autres algorithmes de la littérature.

Dans le cadre de la surveillance de l'écosystème marin, nous avons développé un système de classification automatique de cellules phytoplanctoniques, analysées par cytométrie en flux. Pour cela, nous avons proposé de mesurer les similarités entre cellules par comparaison élastique entre leurs signaux profils caractéristiques.

Mots-clés: Classification spectrale, contraintes de comparaison, réduction de la dimension, phytoplancton, écosystème marin.

Abstract

In the decision support systems, often, there are huge digital data and possibly some contextual knowledge available a priori or provided a posteriori by feedback. The performances of classification approaches, particularly spectral ones, depend on the integration of the domain knowledge in their design. Spectral classification algorithms address the problem of classification in terms of graph cuts. They classify the data in the eigenspace of the graph Laplacian matrix. The generated eigenspace may better reveal the presence of linearly separable data clusters.

In this work, we are particularly interested in algorithms integrating pairwise constraints : constrained spectral clustering. The eigenspace may reveal the data structure while respecting the constraints. We present a state of the art approaches to constrained spectral clustering. We propose a new algorithm, which generates a subspace projection, by optimizing a criterion integrating both normalized multicut and penalties due to the constraints. The performances of the algorithms are demonstrated on different databases in comparison to other algorithms in the literature.

As part of monitoring of the marine ecosystem, we developed a phytoplankton classification system, based on flow cytometric analysis. For this purpose, we proposed to characterize the phytoplanktonic cells by similarity measures using elastic comparison between their cytogram signals.

Keywords: Spectral clustering, pairwise constraints, dimensionality reduction, Phytoplankton, marine ecosystem.