

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMARI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

de MASTER en Automatique, Option : Commande des systèmes

Thème

**Classification automatique de données basée sur
la théorie des graphes
Application à la segmentation d'images couleurs**

Proposé et encadré par :

Mr. Hammouche Kamal

Présenté par :

Mr. Younsi Merzouk

Mr. Salmi Abderezak

Soutenu le : 12/09/2011

Promotion 2011

Ce travail a été préparé au : Laboratoire Robotique et Vision de la FGEI de l'UMMTO

Remerciements



*Avant tout
développement de ce
travail, il apparaît opportun
de commencer par des remerciements.*

*Nous tenons à remercier Allah le Tout-Puissant
de nous avoir donné le courage, la volonté et la patience
de mener à terme ce présent travail. Nous tenons tout
particulièrement à remercier Mr Hammouche Kamal, notre promoteur
qui nous a formés et accompagnés tout au long de cette expérience professionnelle
avec beaucoup de patience et de pédagogie, pour sa disponibilité permanente, ses
discussions fructueuses, mais aussi pour l'aide qu'il nous a offert pendant
la réalisation de ce travail. Tous nos remerciements vont aussi aux
membres de jury ayant accepté d'analyser et de corriger notre
modeste travail. Nos reconnaissances vont également à
l'ensemble des enseignants du département
d'automatique qui nous ont
énormément transmis tout
au long de notre
cursus.*



Dédicaces

Je dédier ce modeste travail à :

- ✓ *La mémoire de mon grand père, que son âme repose en paix*
- ✓ *Mes parents pour leurs aides précieuses, leurs soutient et encouragements durant chaque étape de ma vie*
- ✓ *Mon frère et sœurs*
- ✓ *Tous les membres de ma famille*
- ✓ *Tous mes amis et mes collègues*

Merzouk, Y

Dédicaces

Je dédier cet humble travail :

- ✓ *À La mémoire de ma grande mère.*
- ✓ *À mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance.*
- ✓ *À mes sœurs ainsi qu'à tous les membres de ma famille.*
- ✓ *À Mes amis et mes collègues.*

Abderezak, S

Table des matières

INTRODUCTION GÉNÉRALE	1
-----------------------------	---

Chapitre I : Classification automatique

I.1 Introduction	3
I.2 Définition de la classification	4
I.3 Les méthodes de classification	5
I.3.1 Classification supervisée (Semi-automatique)	6
I.3.1.1 Règle de décision bayésienne	6
I.3.1.2 Classification par les K -plus proches voisins (K -PPV).....	7
I.3.1.3 Classification par arbres de décision (Decision tree).....	9
I.3.1.4 Classification neuronale	11
I.3.2 Classification non supervisée (Clustering)	12
I.3.2.1 Méthodes de classification hiérarchiques	12
I.3.2.1.1 Classification ascendante hiérarchique CAH	13
I.3.2.1.2 Classification descendante hiérarchique CDH	14
I.3.2.2 Méthodes de classification non hiérarchiques	15
I.3.2.2.1 La méthode EM (Estimation Maximisation)	15
I.3.2.2.2 Méthodes des k -Means (k -moyennes)	16
I.3.2.2.3 Méthode ISODATA	20
I.3.2.2.4 La méthode des nuées dynamiques	21
I.3.2.2.5 Classification floue (Fuzzy clustering)	23
I.4 Conclusion	26

Chapitre II : Théorie des graphes

II.1 Introduction.....	27
II.2 Concepts fondamentaux de la théorie des graphes	28
II.2.1 Définition	28
II.2.1.1 Graphe orienté	28
II.2.1.2 Graphe non orienté	30

II.2.1.3	Graphe simple et multiple	31
II.2.1.4	Prédécesseurs, successeurs et voisins d'un graphe	32
II.2.1.5	Degré d'un sommet	33
II.2.1.6	Cheminement dans un graphe	34
1.	Chaîne	34
2.	Chemin	35
3.	Cycle	35
4.	Circuit	35
II.2.2	Structure d'un graphe	36
II.2.3	Graphes particuliers	37
II.2.3.1	Graphe complet	37
II.2.3.2	Graphe complémentaire	37
II.2.3.3	Graphe planaire	38
II.2.3.4	Graphe biparti	39
II.2.3.5	Graphe eulérien	40
II.2.3.6	Graphe hamiltonien	41
II.2.3.7	Arbres	41
II.2.3.8	Arbre recouvrant minimal	42
II.2.3.9	Arborescence	42
II.3	Représentation matricielle d'un graphe	43
II.3.1	Matrice d'adjacence	43
II.3.2	Matrice associée	44
II.3.3	Matrice d'incidence aux arcs	45
II.4	Connexité dans un graphe	45
II.4.1	Connexité	45
II.4.1.1	Notion de connexité	45
II.4.1.2	Composantes connexes	46
II.4.1.3	Graphe connexe	47
II.4.2	Forte connexité	47
II.4.2.1	Notion de forte connexité	47
II.4.2.2	Composantes fortement connexes	48
II.4.2.3	Graphe réduit	48
II.4.2.4	Graphe fortement connexe	49

II.5 Domaines d'applications	49
II.6 Conclusion	50

Chapitre III : Classification spectrale

III.1 Introduction	51
III.2 Notions liées à la classification spectrale	51
III.2.1 Notations sur les graphes	52
III.2.2 Notion de fonction de similarité	53
III.3 Différents types de graphes de similarité (graphes de pondération)	54
III.3.1 Graphe de voisinage - ε (The ε - neighborhood graph)	54
III.3.2 Graphe des K plus proches voisins (k -nearest neighbor graph)	55
III.3.3 Graphe totalement connecté (Fully connected graph)	56
III.4 Matrices laplaciennes d'un graphe et leurs propriétés de base	56
III.4.1 Matrice laplacienne <i>non normalisée</i> d'un graphe	57
III.4.2 Matrice laplacienne <i>normalisée</i> d'un graphe	59
III.5 Algorithmes de classification spectrale	60
1. Algorithme classification spectrale <i>non normalisée</i>	60
2. Algorithme classification spectrale <i>normalisée</i> selon Shi et Malik	61
3. Algorithme classification spectrale <i>normalisée</i> selon Ng, Jordan, et Weiss	62
III.6 Notions de coupes (Cuts) de graphe	63
III.7 Choix des paramètres pour la classification spectrale	71
III.7.1 Paramètre de la fonction de similarité σ	71
III.7.1.1 Approche seuillage	72
III.7.1.2 Approche globale	72
III.7.1.3 Approche locale	72
III.7.2 Choix du type de graphe de similarité	73
III.7.3 Paramètres du graphe de similarité	75
III.7.4 Choix du nombre de classes k	75
III.7.5 Choix de la matrice laplacienne	76
III.7.6 Exemple de classification spectrale	76
III.8 Résumé de la méthode de classification spectrale	78
III.9 Avantages et inconvénients de la classification spectrale	79
III.10 Classification spectrale accélérée (Fast Spectral Clustering)	79

III.11 Conclusion	80
--------------------------------	----

Chapitre IV : Tests et résultats

IV.1 Introduction	81
IV.2 Données artificielles à 2D	82
IV.2.1 Classification spectrale non normalisée	84
IV.2.2 Classification spectrale normalisée selon Shi et Malik	86
IV.2.3 Classification spectrale normalisée selon Ng, Jordan et Weiss	88
IV.3 Données artificielles à 3D	90
IV.4 Classification spectrale accélérée KASP sur des données artificielles	95
IV.4.1 Cas 2D	95
IV.4.2 Cas 3D	97
IV.5 Segmentation d'images couleurs par classification	98
IV.5.1 Rappels sur la segmentation d'images	98
IV.5.2 Représentation d'une image couleur	99
IV.5.3 Segmentation d'images couleurs utilisant la méthode spectrale KASP	100
IV.5.3.1 Segmentation de l'image "synthétique"	101
IV.5.3.2 Segmentation des images réelles	104
IV.5.3.2.1 Segmentation de l'image de "fleurs"	104
IV.5.3.2.2 Segmentation de l'image "house"	107
IV.6 Conclusion	110
CONCLUSION GÉNÉRALE	111
RÉFÉRENCES BIBLIOGRAPHIQUES	
ANNEXES	

Liste des figures

Chapitre I : Classification automatique

I.1	<i>Le principe de la classification</i>	5
I.2	<i>Taxinomie des méthodes de classification</i>	5
I.3	<i>Classification par les KPPV</i>	8
I.4	<i>Classification par arbre de décision</i>	10
I.5	<i>Exemple d'une hiérarchie (dendrogramme)</i>	13
I.6	<i>Critères d'agrégation</i>	13
I.7	<i>Classification ascendante hiérarchique CAH</i>	14
I.8	<i>Inertie intraclasse et inertie interclasse</i>	17
I.9	<i>Déroulement de l'algorithme de k-Means</i>	19
I.10	<i>Phase initiale et phase d'expansion des nuées dynamiques</i>	22
I.11	<i>Phase de contraction et d'expansion</i>	22
I.12	<i>Classification stricte</i>	23
I.13	<i>Classification floue</i>	23

Chapitre II : Théorie des graphes

II.1	<i>Représentation des ponts de la ville de Königsberg</i>	27
II.2	<i>Rives (régions) de la ville de Königsberg</i>	27
II.3	<i>Exemple d'un graphe</i>	28
II.4	<i>Graphe orienté</i>	29
II.5	<i>Les extrémités d'un arc</i>	30
II.6	<i>Graphe non orienté</i>	30
II.7	<i>Arêtes multiples</i>	31
II.8	<i>Arcs multiples</i>	31
II.9	<i>Arcs non multiples</i>	31
II.10	<i>Graphe d'Euler</i>	31
II.11	<i>Les prédécesseurs d'un sommet</i>	32
II.12	<i>Les successeurs d'un sommet</i>	33

II.13	<i>Les voisins d'un sommet</i>	33
II.16	<i>Sous graphe</i>	36
II.17	<i>Graphe partiel</i>	36
II.18	<i>Sous graphe partiel</i>	36
II.19	<i>Graphe complet et graphe non complet</i>	37
II.20	<i>Graphe complémentaire</i>	38
II.21	<i>Graphe planaire</i>	38
II.22	<i>Faces et frontières dans un graphe</i>	39
II.23	<i>Graphe biparti</i>	39
II.24	<i>Graphe simple, biparti-complet</i>	40
II.25	<i>Graphe eulérien</i>	40
II.26	<i>Graphe hamiltonien</i>	41
II.27	<i>Exemple d'arbre et de forêt</i>	41
II.28	<i>Arbre recouvrant minimal</i>	42
II.29	<i>Arborescence</i>	43
II.32	<i>Graphe connexe</i>	47
II.35	<i>Graphe Réduit</i>	49

Chapitre III : Classification spectrale

III.1	<i>Graphe de voisinage ε</i>	54
III.2	<i>Graphe des KPPV standard et mutuel</i>	55
III.3	<i>Graphe totalement connecté</i>	56
III.4	<i>Exemple de graphe pondéré des données</i>	63
III.5	<i>Différents graphes de similarité</i>	74
III.6	<i>Exemple de classification spectrale</i>	77
III.7	<i>Etapas de classification spectrale</i>	78

Chapitre IV : Tests et résultats

IV.1	<i>Données originales brutes (2D)</i>	82
IV.2	<i>Données de référence</i>	83
IV.3	<i>Résultat de classification par la méthode de k-Means</i>	83

IV.4,	<i>Classification spectrale non normalisée</i>	84
IV.5	<i>Classification spectrale normalisée selon Shi et Malik</i>	86
IV.6	<i>Classification spectrale normalisée selon Ng, Jordan et Weiss</i>	88
IV.7	<i>Données originaux brutes (500 points)</i>	90
IV.8	<i>Données originaux brutes (2000 points)</i>	90
IV.9	<i>Données de référence (500 points)</i>	90
IV.10	<i>Données de référence (2000 points)</i>	90
IV.11	<i>Résultat de la méthode de k-Means (500 points)</i>	91
IV.12	<i>Résultat de la méthode de k-Means (2000 points)</i>	91
IV.13	<i>Classification spectrale non normalisée (500 points)</i>	91
IV.14	<i>Classification spectrale non normalisée (2000 points)</i>	92
IV.15	<i>Résultats de classification spectrale accélérée KASP (2D)</i>	95
IV.16	<i>Résultats de classification spectrale accélérée KASP (3D)</i>	97
IV.17	<i>Représentation de l'image de synthèse dans l'espace couleur R, V, B ...</i>	100
IV.18	<i>L'image synthétique</i>	101
IV.19	<i>Résultat obtenu de segmentation par la méthode de k-Means</i>	101
IV.20	<i>Résultats obtenus de segmentation par classification spectrale</i>	102
IV.21	<i>L'image de fleurs</i>	104
IV.22	<i>Résultat obtenu de segmentation par la méthode de k-Means</i>	104
IV.23	<i>Segmentation obtenue par l'algorithme non normalisé</i>	105
IV.24	<i>Segmentation obtenue par l'algorithme normalisé selon SM</i>	105
IV.25	<i>Segmentation obtenue par l'algorithme normalisé selon NJW</i>	105
IV.26	<i>L'image "house"</i>	107
IV.27	<i>Résultat obtenu de segmentation par la méthode de k-Means</i>	107
IV.28	<i>Segmentation obtenue par l'algorithme non normalisé</i>	108
IV.29	<i>Segmentation obtenue par l'algorithme normalisé selon SM</i>	108
IV.30	<i>Segmentation obtenue par l'algorithme normalisé selon NJW</i>	108

Liste des tableaux

Chapitre IV : Tests et résultats

IV.1	<i>Résultat de classification par la méthode de k-Means à 2D</i>	83
IV.2	<i>Résultats de classification spectrale non normalisée à 2D</i>	85
IV.3	<i>Classification spectrale normalisée à 2D selon Shi et Malik</i>	87
IV.4	<i>Classification spectrale normalisée à 2D selon Ng, Jordan et Weiss</i>	89
IV.5	<i>Résultats de classification par la méthode de k-Means à 3D</i>	91
IV.6	<i>Classification spectrale non normalisée à 3D (500 points)</i>	92
IV.7	<i>Classification spectrale non normalisée à 3D (2000 points)</i>	93
IV.8	<i>Classification spectrale non normalisée en 2D (1150 points)</i>	96
IV.9	<i>Classification spectrale accélérée KASP non normalisée en 3D</i>	97
IV.10	<i>Résultat de l'algorithme non normalisé (image synthétique)</i>	102
IV.11	<i>Résultat de l'algorithme normalisé selon SM (image synthétique)</i>	102
IV.12	<i>Résultat de l'algorithme normalisé selon NJW (image synthétique)</i> ...	103
IV.13	<i>Résultat de l'algorithme non normalisé (image fleurs)</i>	106
IV.14	<i>Résultat de l'algorithme normalisé selon SM (image fleurs)</i>	106
IV.15	<i>Résultat de l'algorithme normalisé selon NJW (image fleurs)</i>	106
IV.16	<i>Résultat de l'algorithme non normalisé (image house)</i>	109
IV.17	<i>Résultat de l'algorithme normalisé selon SM (image house)</i>	109
IV.18	<i>Résultat de l'algorithme normalisé selon NJW (image house)</i>	109

Abréviations et notations

Abréviations

KPPV	<i>K plus proches voisins</i>
CAH	<i>Classification ascendante hiérarchique</i>
CDH	<i>Classification descendante hiérarchique</i>
EM	<i>Estimation-Maximisation (Estimation-Maximisation)</i>
FCM	<i>C-moyennes floues (Fuzzy C-Means)</i>
RVB	<i>Vecteur couleur des composantes trichromatiques R, V, B</i>
KASP	<i>Classification spectrale approximée (accélérée)</i>
SM	<i>Shi et Malik</i>
NJW	<i>Ng, Jordan et Weiss</i>

Notations

n	<i>Nombre des observations (individus)</i>
N	<i>Nombre d'attributs</i>
k	<i>Nombre de classes</i>
K	<i>Nombres des K plus proches voisins</i>
X	<i>Ensemble des observations</i>
C	<i>Ensemble des classes</i>
I_{intra}	<i>Inertie intraclasse</i>
I_{inter}	<i>Inertie interclasse</i>
T_c	<i>Taux de classification</i>
t_c	<i>Temps de calcul</i>
T_{cMax}	<i>Taux de classification maximal</i>
Q	<i>Valeur du critère de Borsotti</i>
ε	<i>Rayon de voisinage pour le graphe de voisinage ε</i>
σ	<i>Paramètre de la fonction de similarité gaussienne</i>
\mathbb{P}	<i>Nombre de points représentatifs de la méthode KASP</i>

Introduction générale

Introduction générale

Aujourd'hui, nous vivons dans un monde où l'information est disponible en grande quantité. Mais toute cette information complexe serait sans intérêt si notre capacité à y accéder efficacement n'augmentait pas elle aussi. Ces informations sont généralement multidimensionnelles, très nombreuses et souvent bruitées. Pour cela, nous avons besoin d'outils permettant de chercher, classer, conserver, analyser toutes ces informations. Il est ainsi nécessaire de proposer des systèmes capables d'accéder rapidement à l'information désirée tout en réduisant l'implication humaine. Un des domaines qui tente d'apporter des améliorations et de réduire la tâche de l'humain est sans doute la classification de données.

La classification de données (data clustering) désigne l'ensemble des méthodes destinées à l'exploration et l'analyse des données (images, sons, textes, pages web,...etc.), de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information pertinente à partir de la quelle l'interprétation pourra se réaliser.

Le problème de la classification peut s'énoncer comme suit : étant donné n individus (ou objets) définis par N variables (ou attributs). On regroupe ces individus à l'aide d'une mesure de similarité, en classes de telle sorte que les objets de la même classe soient le plus semblable possible et des objets de classes différentes le moins semblable possible.

Les méthodes de classification peuvent être divisées en deux grandes familles : si les classes sont prédéterminées et les prototypes des classes (exemples) sont connus à *priori*, le système apprend à classer selon un modèle de classement. Un expert doit, au préalable, correctement étiqueter ces exemples ; on parle alors de classification supervisée ou d'analyse discriminante. Quand on ne dispose que d'exemples, mais non d'étiquettes, et la nature des classes n'a pas été prédéterminée, on parle de classification non supervisée dénommée aussi apprentissage automatique. Dans ce cas, aucun expert n'est requis. L'algorithme de classification doit découvrir par lui-même la structure plus ou moins cachée des données en partitionnant les données en classes.

Plusieurs méthodes de classification automatique basées sur des principes différents ont été proposées. Parmi elles, on peut citer l'algorithme de: k -Means, Fuzzy C-Means (FCM), Estimation-Maximisation (EM),...etc.

Notre objectif principal dans ce mémoire est l'étude d'une méthode de classification non supervisée appelée méthode spectrale (spectral clustering). Celle-ci est issue de la théorie des graphes qui constitue un outil puissant de modélisation et de résolution de problèmes concrets (en : recherche opérationnelle, génie électrique, programmation, gestion des réseaux, biologie, ...etc.). Le but de la classification spectrale consiste à partitionner un graphe de similarité en composantes connexes dont chacune d'elle correspond à une classe.

Un des domaines d'application de la classification non supervisée est la segmentation d'images qui peut être définie comme une classification des pixels ayant des caractéristiques similaires. La segmentation d'images constitue une étape très importante dans un système de vision artificielle. Ainsi, un autre intérêt de notre travail est l'application de la méthode spectrale à la segmentation d'images couleurs.

Notre mémoire est organisé comme suit :

Le premier chapitre dresse un état de l'art sur les méthodes de classification de données. Le début du chapitre commence par une définition. Ensuite, des différentes méthodes de classification seront présentées selon leur approche supervisée ou non supervisée.

Le deuxième chapitre est consacré à la théorie des graphes sur laquelle est basée la classification spectrale. Le début du chapitre rappelle quelques concepts fondamentaux sur la théorie des graphes. La représentation matricielle d'un graphe et les différentes matrices reliées au graphe seront également présentées. La fin du chapitre décrit la notion de connexité et de forte connexité dans un graphe.

Le troisième chapitre est dédié à l'étude de la classification spectrale (spectral clustering). Le début du chapitre présente les différents types de graphes de similarité utilisés en classification spectrale. Les différents algorithmes de cette méthode, leurs fondements théoriques, leurs caractéristiques principales ainsi que leurs principales limitations seront principalement présentés.

Le chapitre quatre expose les principaux résultats obtenus en utilisant la méthode de k -Means, puis par la méthode spectrale accompagnés de leurs interprétations. La première partie présente les résultats obtenus avec des données artificielles (en 2D et 3D). La deuxième partie présente les résultats de segmentation d'images couleurs. Des comparaisons seront faites, entre la méthode de k -Means et la méthode spectrale d'une part, et entre les différents algorithmes de classification spectrale d'une autre part. La conclusion générale synthétise notre travail.

CHAPITRE I



Classification Automatique

I.1 Introduction

La classification automatique, ou plus simplement classification est la plus répandue des techniques descriptives d'analyse de données. Elle est utilisée quand on dispose d'un grand volume de données (pages web, textes, images, sons, espèces animales et végétales...etc.) qui constituent initialement un mélange complexe, on utilise la classification pour les organiser en classes homogènes vu de les modéliser.

En tant que branche de la statistique, la classification à été énormément étudiée depuis de nombreuses années. Avec l'avènement de l'informatique dans les années 1960, les méthodes de classification ont connues de nouveaux développements méthodologiques qui ont donné lieu à des algorithmes de classification automatique, mais il faut savoir que ce sujet connaît une effervescence théorique jamais ralentie.

De nos jours, la classification est une démarche qui est appliquée dans un nombre important de domaines tels que la surveillance et le diagnostic, le traitement et l'analyse du signal et de l'image, la biologie et la géologie,...etc. En économie par exemple, la classification peut aider les analystes à découvrir des groupes distincts dans leurs base clientèle et à caractériser ces groupes de clients, en se basant sur des habitudes de consommations. En biologie, on peut l'utiliser pour dériver des taxinomies de plantes et d'animaux, pour catégoriser des gènes avec une ou plusieurs fonctionnalités similaires, pour mieux comprendre les structures propres aux populations. Il est possible aussi de classer des documents sur le web, pour obtenir de l'information utile. En marketing, pour le lancement d'un nouveau produit, le responsable du service de marketing d'une entreprise pourra chercher à constituer des groupes de villes semblables vis à vis d'un certain nombre de critères, et choisir dans chaque groupe une ville type utilisée comme marché-test.

En traitement d'image, la classification joue un rôle très important pour résoudre plusieurs problèmes en reconnaissance de formes (reconnaissance de la parole, des visages, des empreintes,...), en segmentation d'images, elle est utilisée afin de regrouper en classes les pixels qui ont des propriétés similaires,...etc. Elle a pour but de produire à partir des mesures radiométriques, une image thématique qui est une interprétation et une catégorisation des objets associés aux mesures. Il s'agit d'obtenir des classes dont les éléments (pixels) ont en commun divers propriétés (intensité, couleur, texture, forme,...etc.) telles que deux pixels de la même classe se ressemblent le plus possible et deux pixels de deux classes distinctes se ressemblent le moins possible.

Dans ce chapitre, nous décrivons les principales méthodes de classification, qu'on peut regrouper en deux catégories: les méthodes de classification supervisée, qui s'opèrent à partir de la connaissance de chacune des classes, et la classification non supervisée (clustering) qui vise à séparer automatiquement un ensemble d'objets en clusters naturels. Ainsi, en classification non supervisée aucune information sur la classe d'un objet n'est fournie à la méthode (on dit que les objets ne sont pas étiquetés).

La forme des classes, le degré de chevauchement entre les classes et la nature multidimensionnelle des observations constituent les difficultés les plus importantes dans la résolution d'un problème de classification.

I.2 Définition de la classification

La classification (clustering en anglais) est l'opération statistique qui consiste à regrouper un ensemble de données (objets, individus ou variables) en un nombre limité de classes (clusters) selon un ou plusieurs critères de classification (inertie, distance, densité de probabilité...etc.). L'algorithme qui réalise automatiquement une classification est appelé classifieur. La figure (I.1) illustre le principe de la classification.

Les données à classer se présentent, soit sous forme d'une matrice (ou tableau) rectangulaire dont les lignes sont les n individus et dont les colonnes sont les N variables (attributs), soit sous forme d'une matrice carrée de similarité, représentant des distances entre individus ou entre variables [1].

Graphiquement, les données sont représentées par des vecteurs (points) à N attributs projetés dans un espace euclidien E^N à N dimensions (dans notre exemple, $N = 2$). En sortie de la classification, nous obtenons k classes de points selon les critères donnés et le classifieur choisi (dans notre exemple, $k = 3$). Ainsi, les individus (objets) ayant des attributs semblables appartiennent à la même classe et forment un nuage de points très dense dans l'espace de représentation (espace des attributs) à N dimensions. Mathématiquement, la classification est définie de la manière suivante :

- Soit $X = \{x_1, x_2, \dots, x_n\}$, l'ensemble des n observations (individus) à classer. Chaque individu x_i de X est caractérisé par N paramètres (attributs), tel que $x_i \in \mathbb{R}^N$.
- Soit $C = \{C_1, C_2, \dots, C_k\}$ l'ensemble des k classes non vides de X d'intersections vides deux à deux et dont la réunion forme X ayant les propriétés suivantes [2] :

1. $C_i \neq \emptyset, \forall i=1,2,\dots,k$
2. $C_i \cap C_j = \emptyset, \forall i,j = 1,2, \dots, k \text{ et } i \neq j$
3. $\bigcup_{i=1}^k C_i = X$

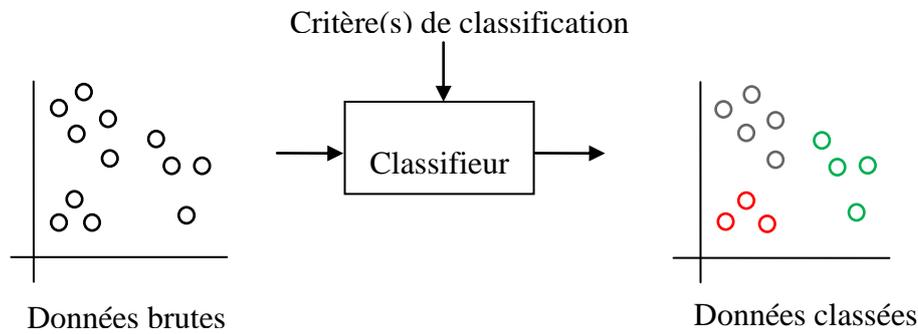


Figure I.1 : Le principe de la classification

I.3 Les méthodes de classification

Les méthodes de classification permettent de réunir les éléments d'un ensemble hétérogène, en un nombre limité de classes les plus homogènes possibles suivant un critère connu (que ce soit la similarité entre les objets, les mesures probabilistes ou encore la densité des classes). Plusieurs méthodes de classification ont été proposées dans la littérature. Ces méthodes peuvent être subdivisées en deux catégories selon les résultats qu'on veut obtenir et la nature des données (figure (I.2))

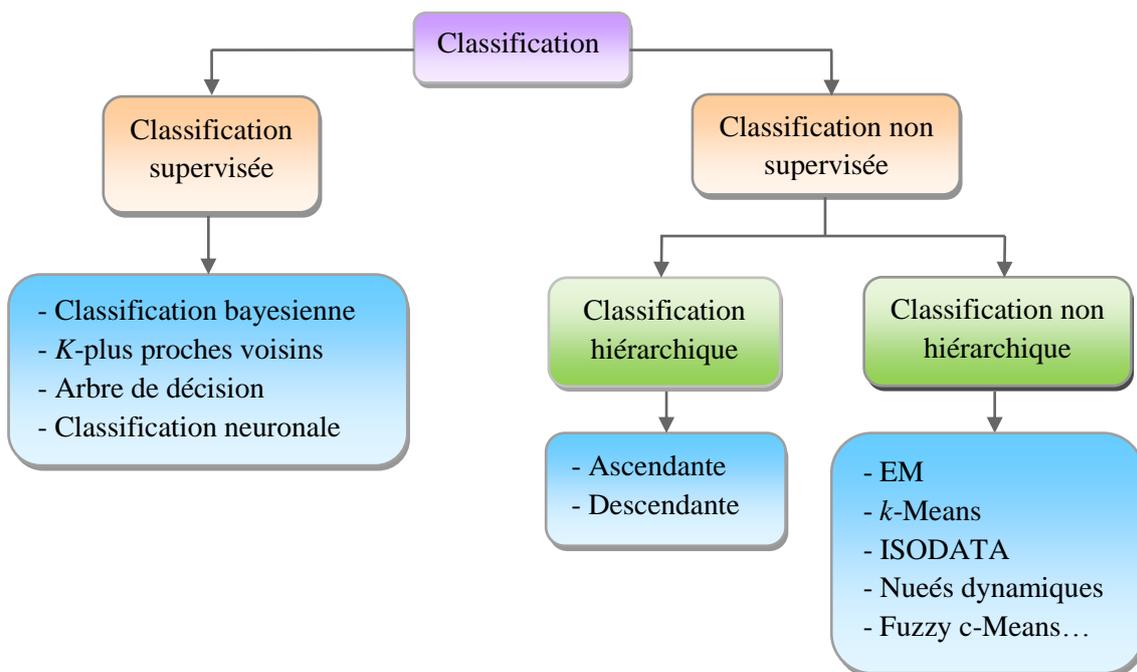


Figure I.2 : Taxinomie des méthodes de classification

I.3.1 Classification supervisée (Semi-automatique)

La classification supervisée consiste à déterminer les règles qui permettent de séparer un ensemble d'individus en classes connues à *priori*. Son but principal est de trouver à partir d'exemples déjà étiquetés un modèle afin de prédire les classes des nouvelles données.

I.3.1.1 Règle de décision bayésienne

Le problème de la classification en général, consiste à associer une classe à une nouvelle donnée, en fonction des observations disponibles dont l'appartenance à une classe est connue à *priori*. Il est question donc, de déterminer une règle de décision qui permet d'affecter à une classe adéquate une nouvelle donnée. Intuitivement, il semble que la meilleure règle de décision est celle qui choisit la classe ayant la probabilité à *posteriori* (c'est-à-dire sachant les observations) maximale.

Effectivement, plusieurs travaux ont été consacrés à la démonstration de cette règle, appelée *règle de décision bayésienne*.

La décision bayésienne permet de reconnaître si un vecteur de forme X appartient à une classe parmi les k classes de l'ensemble $C = \{C_r\}$ ($r=1, 2, 3, \dots, k$), ce vecteur X est supposé aléatoire dont on connaît la densité de probabilité conditionnelle $p(X/C_r)$. La règle de décision de Bayes permet de déterminer la meilleure classe $C_r(X)$ ou la classe la plus probable en appliquant la formule suivante qui est très pratique du fait que l'on connaît souvent la probabilité à *posteriori* $p(X/C_r)$;

$$\text{decider } C_j \quad \text{si } p(X/C_j)P(C_j) \geq p(X/C_r)P(C_r) \quad \forall r \neq j, 1 \leq r \leq k$$

Notons que la probabilité qu'une observation X appartienne à la classe C_r est donnée par la formule de Bayes suivante :

$$P(C_r/X) = \frac{p(X/C_r)P(C_r)}{p(X)} \quad (I.1)$$

Il est souhaitable que cette règle soit optimale, dans le sens où elle minimise le coût moyen des erreurs commises lors de la classification. Suite à un calcul statistique, il a été démontré que la minimisation de ce coût revient à la maximisation de la probabilité à *posteriori* $p(X/C_r)$.

I.3.1.2 Classification par les K -plus proches voisins (K -PPV)

Connue aussi en anglais sous le nom *K-nearest neighbor K-NN* (Weiss and Kulikowski, 1990), ou encore *Memory Based Reasoning* (Stanfill and Waltz, 1986). C'est une méthode très connue dans le domaine de la classification supervisée. Elle diffère des méthodes traditionnelles d'apprentissages car aucun modèle n'est induit à partir des exemples. Les observations restent telles quelles, elles sont simplement stockées en mémoire.

Pour prédire la classe d'un nouvel individu, la méthode utilise deux paramètres : le nombre K et la fonction de similarité (la distance par exemple) pour comparer le nouvel individu au cas déjà classés. Ainsi, pour déterminer à quelle classe appartient le nouvel individu (objet), l'algorithme des K -PPV cherche les K plus proches voisins de ce nouvel objet en calculant souvent la distance (*annexe D*) de cet individu aux différents individus classés. La distance la plus utilisée est la distance euclidienne d donnée par la relation suivante:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^N (a_r(x_i) - a_r(x_j))^2} \quad (I.2)$$

Avec :

$a_r(x_i)$: attribut du nouvel individu.

$a_r(x_j)$: attribut de $j^{\text{ième}}$ individu classé avec $j=1, \dots, n$ et n : nombre d'individus classés.

r : indice de l'attribut.

N : nombre d'attributs.

$d(x_i, x_j)$: distance entre le nouvel individu i et le $j^{\text{ième}}$ individu classé.

Pour $K = 1$ la classification est dite 1-PPV et la classe du nouvel objet est donnée par la classe du plus proche voisin. Par contre, dans le cas de plusieurs voisins la décision d'affecter un nouvel objet peut suivre plusieurs stratégies :

- Affecter à la classe majoritaire de ses K voisins.
- Pondérer l'affectation de la classe avec la distance : plus un voisin est éloigné, plus son influence est faible.

Dans l'exemple suivant (figure (I.3.a)), la classification est simple quelque soit le nombre de voisins choisis : le nouvel individu est affecté à la classe des rouge. Par contre, sur la figure (I.3.b) la décision d'affectation dépendra du nombre de voisins choisis et de l'heuristique de classification.

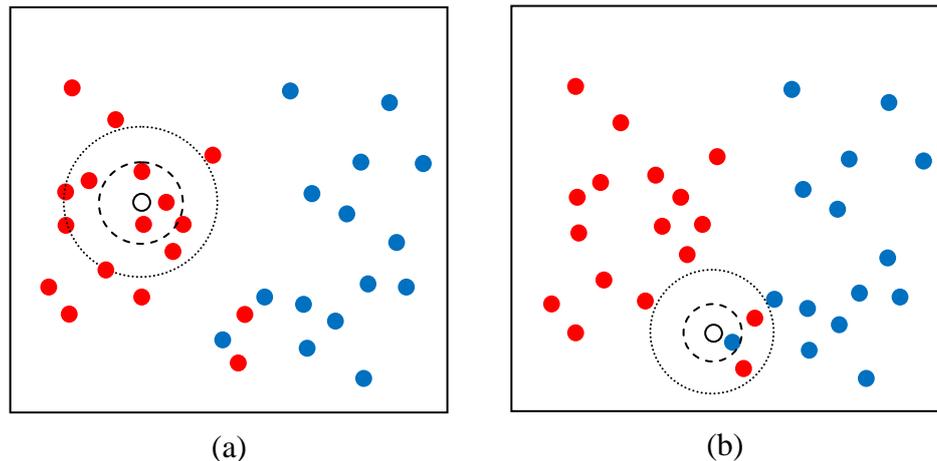


Figure I.3: Classification avec les K -PPV. Dans la figure (a) aucune ambiguïté possible. Dans la figure (b) la décision d'affectation dépendra du nombre de voisins choisis (K) et de l'heuristique de la classification [3].

Ainsi pour $K = 1$, le nouvel individu est affecté à la classe des bleu. Pour $K = 3$, si les trois voisins ont le même poids, alors le nouvel objet est affecté à la classe des rouge. Si par contre, le poids est pondéré par l'inverse de la distance, alors la classe du nouvel objet peut être celle des bleu. On remarque donc que le choix de K est très important dans le résultat final.

Algorithme des K -plus proches voisins

- 1- **Paramètre:** le nombre K de voisins (compris entre 1 et n le nombre maximum de voisins).
 - 2- Soit $C = (C_1, \dots, C_k)$ les k classes prédéfinies.
 - 3- Transformer le nouvel individu v en vecteur $v = (x_1, x_2, \dots, x_N)$.
 - 4- Calculer les distances $d(v, v_j)$, $j=1 \dots n$ (où les v_j sont les individus).
 - 5- Retenir les K plus proches voisins de cet individu v pour lesquels ces distances sont les plus petites.
 - 6- Compter le nombre de fois K_1, \dots, K_m que ces K voisins apparaissent dans chacune des classes.
 - 7- Attribuer v à la classe la plus représentée dans son entourage.
 - 8- **Sortie :** l'individu v affecté à la classe $C(v)$.
-

Avantages de classification par les K plus proches voisins:

La méthode des K -PPV est très connue dans le domaine de classification supervisée. Ses performances la situe parmi les meilleures méthodes de catégorisation. Ses avantages sont :

- *Absence d'apprentissage* : car c'est l'échantillon qui constitue le modèle. L'introduction de nouvelles données permet d'améliorer la qualité de la méthode sans nécessité la reconstruction d'un modèle.
- *Données hétérogènes* : la méthode peut s'appliquer dès qu'il est possible de définir une distance sur les champs. Or, il est possible de définir des distances sur des champs complexes, tels que des informations géographiques, des textes, des images ou du son. C'est parfois un critère de choix de la méthode K -PPV car les autres méthodes traitent difficilement des données complexes. On peut noter, également, que la méthode est robuste face au bruit.
- *Grand nombre d'attributs* : la méthode permet de traiter des problèmes avec un grand nombre d'attributs. Cependant, plus le nombre d'attributs est important, plus le nombre d'exemples doit être grand.

Inconvénients :

- *Le temps de classification* : si la méthode ne nécessite pas d'apprentissage, tous les calculs doivent être effectués lors de la classification. Ceci est la contrepartie à payer par rapport aux méthodes qui nécessitent un apprentissage (éventuellement long) mais qui sont rapides en classification (le modèle est créé, il suffit de l'appliquer à l'exemple à classifier).
- *Définir une distance et le nombre de voisins* : les performances de la méthode dépendent du choix de la distance (*annexe D*) et du nombre de voisins. En règle générale, les distances simples fonctionnent bien. Si les distances simples ne fonctionnent pour aucune valeur de K , il faut envisager le changement de distance, ou le changement de la méthode.

I.3.1.3 Classification par arbres de décision (Decision tree)

La méthode des arbres de décision est une méthode de classification supervisée représentée sous la forme d'une *arborescence* dont le principe repose sur un partitionnement récursif des données. Elle est composée d'une racine (nœud initial) qui est le point de départ de l'arbre de décision.

Les nœuds internes représentent des questions sur les attributs, chaque lien (arc) correspond à une valeur d'attribut ou bien la réponse à la question (pour un arbre binaire, chaque question ne peut avoir que deux réponses), le résultat indique le chemin à suivre dans l'arbre. Les feuilles (nœuds terminaux) quand à elles, sont étiquetées par une classe par défaut. En bref, pour déterminer à quelle classe appartient un nouvel individu, l'arbre de décision test ses attributs les uns à la suite des autres. L'arbre décidera ensuite de la classe à attribuer à cet individu (d'où l'appellation d'arbre de décision).

Un exemple de classification par arbre de décision est donné par la figure suivante :

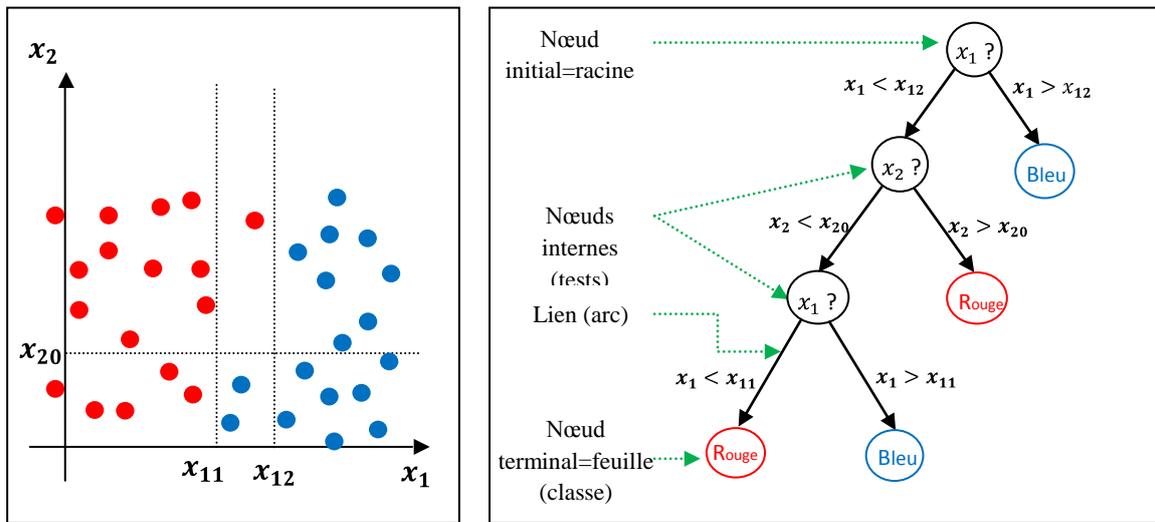


Figure I.4: Classification par arbre de décision. Les valeurs x_{11} , x_{12} et x_{20} sont fixées par l'expert [3]

Sur une population initiale d'exemples, une question (ou sélecteur) est posée à chaque exemple afin de couper cette population en deux parties. La question porte sur un attribut.

Si les attributs sont binaires (chaque nœud interne aura deux nœuds fils), alors la question est « l'attribut λ est il vrai ou faux ? »

Si les attributs sont continus (figure I.4), alors la question est « la valeur de l'attribut λ est elle supérieure à une valeur λ_0 ? »...et ainsi de suite.

La règle de classification est ensuite la suivante : à tout nouvel individu (objet), une succession de questions (tests) lui est soumis, définies à partir de ses réponses successives (cheminement dans l'arbre).

Algorithme de classification par arbres de décision

Algorithme : *construire_arbre*

- 1- **Données :** classes, seuils,...
 - 2- **Début**
 - 3- **Déterminer** le meilleur attribut pour faire le test
 - 4- **Créer** un nœud **T** étiqueté par cet attribut
 - 5- **Créer** autant de fils que l'attribut **T** a de valeurs et étiqueter les arcs par le prédicat ainsi formé
 - 6- **Pour** chaque fils **F faire :**
 - 7- **Associer** à ce fils les objets du nœud père vérifiant le prédicat
 - 8- **Si** nœud vérifie le critère de terminaison **alors :**
 - 9- **Transformer** ce nœud en feuille et lui attribuer une classe
 - 10- **Sinon** *construire_arbre*
-

Avantages des arbres de décision :

- Faciles à comprendre et à utiliser.
- Classification aisée et rapide sur des bases de taille moyenne.
- Interprétation et traduction directe de l'arbre vers une base de règles.
- Construction efficace (technique).

Inconvénients :

- Sensibilité (manque de robustesse) aux bruits et aux points aberrants.
- Difficulté d'optimisation : « *quel est le meilleur arbre ?* »

I.3.1.4 Classification neuronale

Les travaux sur les réseaux de neurones ont fait l'objet de recherches de communautés entières. La classification est un problème sur lequel les réseaux de neurones donnent de bons résultats et s'appliquent avec succès. Il existe plusieurs types de réseaux de neurone dont l'objectif est de classer des objets de deux manières différentes [4]:

Les réseaux à apprentissage supervisé, on donne au réseau des entrées, et au même temps les sorties que l'on désirerait obtenir pour cette entrée. Les réseaux les plus connus sont : Perceptron, Adaline, ... etc [5].

Les réseaux à apprentissage non supervisé, on présente une entrée au réseau et on le laisse évoluer librement jusqu'à ce qu'il se stabilise. Leur fonctionnement s'inspire largement d'une approche de type k -Means (voir la suite). Les plus connus sont : le réseau de Kohonen (Kohonen, T. 1990, [6]) [4] [5], le réseau de Hopfield, le réseau de ART (ang : Adaptive Resonance Theory).

L'avantage majeur que peuvent représenter les réseaux de neurones est la facilité de mise en œuvre [7].

I.3.2 Classification non supervisée (Clustering)

Contrairement à la classification supervisée, la classification non supervisée ne dispose pas d'un ensemble d'apprentissage. Elle consiste à identifier des groupes d'objets ou d'individus similaires (clusters) à partir d'un ensemble de données sans en connaître au préalable la structure. De ce fait, cette dernière cherche à diviser et à classer un ensemble de données de sorte que les classes soient les plus homogènes possibles. Généralement, on distingue deux types d'approches, hiérarchique et non hiérarchique.

I.3.2.1 Méthodes de classification hiérarchiques

Définition d'une hiérarchie [8] :

H_E est un sous ensemble de l'ensemble $\wp(E)$ des parties de E ayant les propriétés suivantes :

- L'ensemble E est un élément de H_E
- $\{i\} \in H_E$ pour tout i élément de E
- $h \cap h' \in \{\emptyset, h, h'\}$, pour h et h' deux parties quelconques de H_E , autrement dit h et h' sont soit disjointes, soit incluses l'une dans l'autre.

La figure (I.5) montre un exemple d'une hiérarchie, si $\wp(E) = \{a, b, c, d, e, f, g\}$ un ensemble d'éléments.

$H_E = \{a, b, c, d, e, f, g, ab, abc, ef, efg, defg, abcdefg\}$ est une hiérarchie de $\wp(E)$.

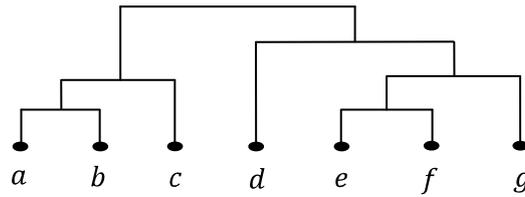


Figure I.5: Exemple d'une hiérarchie (dendrogramme) H_E

Le but des méthodes hiérarchiques est de construire une suite de partitions, elles cherchent à former à partir d'une population donnée une hiérarchie que l'on visualise graphiquement sous forme d'un arbre hiérarchique connu aussi sous le nom de dendrogramme. Ces techniques fournissent un ensemble de classes incluant les unes dans les autres, tel que les clusters des plus bas niveaux sont plus similaires, tandis que les clusters des plus hauts niveaux sont moins similaires. Les deux partitions extrêmes étant la partition discrète, ou chaque individu forme à lui seul une classe, et la partition grossière, ou il n'y a qu'une seule classe formée par tous les individus. Il existe deux types de familles de méthodes hiérarchiques : une descendante dite divisive, et une ascendante dite agglomérative.

I.3.2.1.1 Classification ascendante hiérarchique CAH

C'est la méthode la plus couramment utilisée en classification hiérarchique, elle consiste, à partir des objets (chacun est dans sa propre classe) à agglomérer les classes les plus proches, afin de n'en obtenir plus qu'une seule contenant tous les objets.

Les principales étapes de l'algorithme de classification ascendante hiérarchique (CAH) sont :

- 1- Au départ, chaque objet (singleton) est dans sa propre classe.
- 2- On fusionne les deux classes les plus proches (selon le critère choisi, figure (I.6)).
- 3- On répète l'étape 2 jusqu'à n'avoir plus qu'une seule classe.

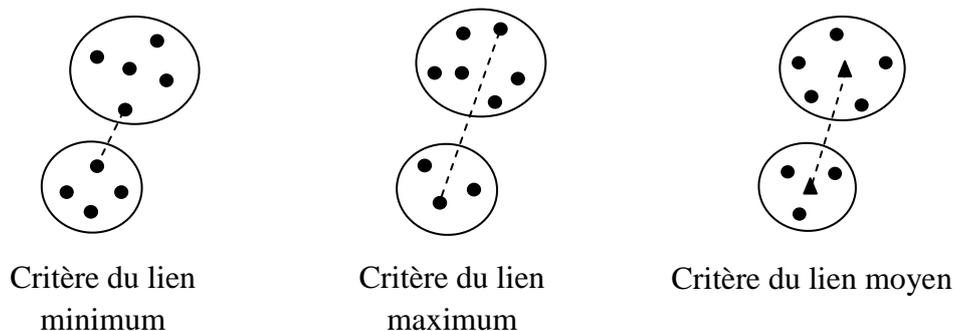


Figure I.6: Critères d'agrégation

La figure ci-dessous présente un exemple d'agglomération de classes (principe de CAH).

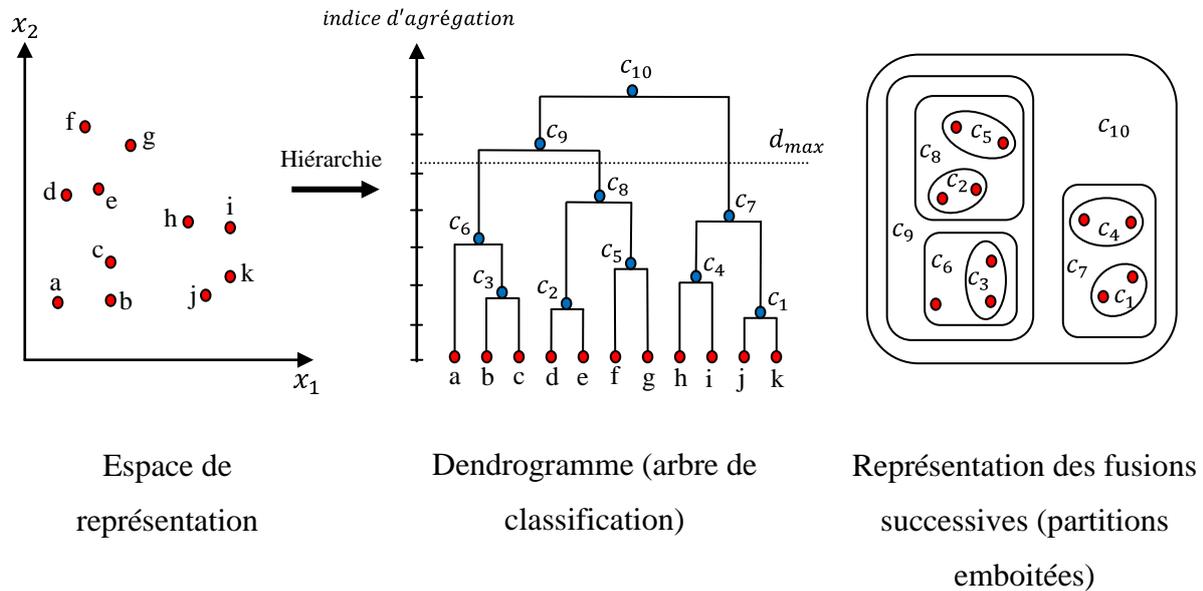


Figure I.7 Exemple de classification ascendante hiérarchique CAH

Pour obtenir un nombre donné de classes, il suffit de tronquer l'arbre de la hiérarchie au niveau convenable. Pour cela, on trace une horizontale d_{max} sur le graphique du dendrogramme (comme ici sur la figure, la droite en pointillé qui indique trois classes, C_6 , C_7 , C_8). On peut en particulier faire en sorte d'obtenir un nombre fixé de classes.

Malgré sa simplicité de programmation, la CAH présente un inconvénient, c'est qu'elle est couteuse en temps de calcul en cas de données très abondantes.

I.3.2.1.2 Classification descendante hiérarchique CDH

Cette technique est moins utilisée par rapport à la CAH. Elle consiste à fractionner l'ensemble des données en sous classes et chaque sous classe est à son tour fractionnée en d'autre classes jusqu'à avoir de petites classes (singletons). Ainsi, on démarre d'une classe mère et on aura comme résultat un ensemble de sous classes. Plus on descend plus on maximise l'homogénéité interne des sous ensembles et la dispersion entre elles.

Nous ne détaillerons pas davantage cette méthode qui est très couteuse pour être utilisée sur les volumes de données manipulés aujourd'hui. En effet, la division d'une partie à n éléments nécessite l'évaluation des $2^{(n-1)} - 1$ divisions possibles.

I.3.2.2 Méthodes de classification non hiérarchiques

Contrairement aux méthodes hiérarchiques, dans ces méthodes, les clusters (classes) ne forment pas d'hierarchie (ils sont sur le même niveau). On parle alors de *partitionnement* de la collection. L'objectif dans ce type de méthodes est alors de déterminer les différentes classes sans pour autant former d'hierarchie. On distingue plusieurs méthodes, à savoir :

I.3.2.2.1 Méthode Estimation Maximisation (Expectation Maximisation EM)

La méthode EM (Dempster et al., 1977 [17]) est une méthode d'optimisation itérative basée sur deux étapes, Estimation-Maximisation. Dans l'étape d'estimation, on calcul les probabilités à *posteriori* d'appartenance d'un objet (observation) à une classe, autrement dit on estime les probabilités de la distribution $f_r(X_i|\theta_r)$ qui peut être soit gaussienne, exponentielle, de Cauchy, de Poisson..., et dans la seconde étape (Maximisation) on estime les paramètres inconnus, autrement dit on maximise la log-vraisemblance donnée sous la forme :

$$L(\theta, C|X) = \sum_{i=1}^n \log \left(\sum_{r=1}^k p_r f_r(X_i|\theta_r) \right) \quad (I.3)$$

Où :

- $X = (x_1, x_2, \dots, x_n)$ un échantillon d'individus suivant une loi $f(X|\theta)$ paramétrée par θ
- $C = \{C_1, C_2, \dots, C_k\}$ est l'ensemble des différentes classes
- p_r est la probabilité pour une observation (individu) d'appartenir à la classe C_r ($r=1, \dots, k$)
- θ est le vecteur de paramètres à estimé

Algorithme de classification par la méthode EM [48]

1) Entrées : nombre de classes k , l'ensemble d'individus $X = (x_1, \dots, x_n)$, critère d'arrêt ε

2) Initialisation au hasard de $\theta^{(0)}$

3) $c = 0$ (avec c le nombre d'itérations)

- Tant que $(Q(\theta, \theta^{(c)}) - Q(\theta, \theta^{(c+1)})) > \varepsilon$, faire:

Etape d'estimation (E) : pour le calcul de $Q(\theta, \theta^{(c)}) = E[L(\theta, C|X)|\theta^{(c)}]$

Etape de maximisation (M) : $\hat{\theta}^{(c+1)} = \arg \max_{\theta} Q(\theta, \theta^{(c)})$

$$c = c + 1$$

- Fin
-

Ces deux étapes sont répétées jusqu'à la convergence de l'algorithme. Celui-ci fournit une solution qui est un optimum local. Il arrive parfois que la convergence amène une solution qui est sur un point selle. Pour pallier cet inconvénient, des versions stochastiques de EM ont vues le jour, comme SEM (Celeux et Diebolt, 1985 [18]), SAEM (Simulated Annealing EM) (Celeux, Diebolt, Biscarat, 1992 [20]) et MCEM (Monte Carlo EM) (Wei et Tanner, 1990[21]).

Un algorithme dérivé de la méthode EM très utilisé, qui est l'algorithme CEM (Celeux et Govaert, 1991 [19]). Entre les deux étapes Estimation et Maximisation, une autre étape de classification à été rajoutée, qui va allouer chaque objet à la classe la plus proche, passant ainsi d'une partition floue (dans EM) à une partition dure (dans CEM).

Avantages et inconvénients de l'algorithme EM :

- En général très simple à mettre en place
- Sensibilité à l'initialisation
- Optimisation locale de la vraisemblance
- Bon comportement pratique
- Peut être lent dans certains cas : classes très mélangées par exemple

I.3.2.2.2 Méthode de k -Means (k -moyennes)

Cette méthode est encore appelée *algorithme des centres mobiles* (Benzécri, 1973). Ce type d'algorithme a été étudié par plusieurs auteurs à savoir, Bonner (1964) ; J-Mc Queen (1967) ; Celeux et al., (1989) [13].

La méthode des k -Means (mise au point par J-Mc Queen en 1967) est la technique de classification non supervisée la plus utilisée de par sa simplicité, la qualité des résultats qu'elle offre. Elle est inspirée de la stratégie de regroupement de la vision humaine. Elle consiste à partitionner l'ensemble des n individus (objets) en k classes (k est déterminé à *priori* par l'utilisateur), de tel sorte à minimiser une fonction objectif.

Cette méthode repose sur un algorithme simple et permet de traiter rapidement des ensembles d'effectifs assez élevés [9], il consiste à minimiser le critère suivant:

$$I_{intra}(C, g) = \frac{1}{n} \sum_{r=1}^k \sum_{i=1}^n d^2(x_i, g_r) \quad (I.4)$$

Ou :

- $X = \{x_i / i = 1, 2, \dots, n\}$ est l'ensemble des n objets à classer.
- $C = \{C_r / r = 1, 2, \dots, k\}$ est l'ensemble des classes tel que : $C_1 \cup C_2 \cup \dots \cup C_k = X$.
- $G = \{g_r / r = 1, 2, \dots, k\}$ est l'ensemble des vecteurs centres des classes C_r .
- d mesure la similarité entre le centre d'une classe et le vecteur de données.

Rappelons que le critère $I_{intra}(C, g)$, qui est simplement la somme des inerties des k classes, est appelé *inertie intraclasse*. La méthode des k -Means consiste à chercher la partition telle que I_{intra} soit minimale pour avoir en moyenne des classes bien homogènes, ce qui revient à chercher le maximum de l'*inertie interclasse* $I_{inter}(C)$:

$$I_{inter}(C) = \frac{1}{n} \sum_{r=1}^k n_r d^2(g_r, \bar{g}) \tag{I.5}$$

Ou :

- \bar{g} est le centre de gravité de l'ensemble des individus.
- n_r est le nombre d'individus dans la classe r .

Ce résultat est dû à la relation liant l'inertie totale qui ne dépend pas de la partition et les *inerties interclasse et intraclasse* (voir, la figure (I.8)) :

$$I_T = I_{intra}(C, g) + I_{inter}(C) \tag{I.6}$$

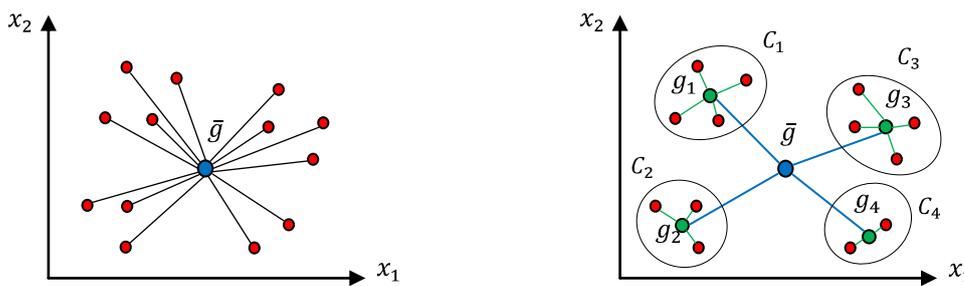


Figure I.8: Inertie totale des points = Inertie intraclasse + Inertie interclasse
(faible inertie intraclasse, forte inertie interclasse)

L'algorithme de k -Means se déroule de la façon suivante (figure (I.9))[8] :

Itération 0 : on tir k individus au hasard, ces individus représentent k centres provisoires de classes:

$$C^0 = \{C_1^0, C_2^0 \dots C_k^0\}$$

Itération 1 : on calcule les distances qui séparent chaque individu des centres des classes.

Chaque individu est affecté à la classe du centre le plus proche. On obtient ainsi une première partition P^0 de l'ensemble en k classes:

$$P^0 = \{P_1^0, P_2^0, \dots, P_k^0\}$$

Itération 2: on calcule donc pour chaque classe de P obtenue précédemment les k nouveaux centres de classes:

$$C^1 = \{C_1^1, C_2^1, \dots, C_k^1\}$$

Puis on réaffecte les individus aux centres les plus proches. Ce qui permet de définir une nouvelle partition P^1 de l'ensemble:

$$P^1 = \{P_1^1, P_2^1, \dots, P_k^1\}$$

Itération m : à cette étape, on détermine les k centres $C_1^{m-1}, C_2^{m-1}, \dots, C_k^{m-1}$ des classes respectives de la partition de l'ensemble obtenu à l'étape $(m - 2)$ soit:

$$P^{(m-1)} = \{P_1^{m-1}, P_2^{m-1}, \dots, P_k^{m-1}\}$$

Le processus est réitéré jusqu'à ce que deux itérations successives conduisent à la même partition, ou lorsque le nombre d'itérations choisi par l'utilisateur au départ est atteint. Il est donc important de s'assurer que le nombre d'itérations choisi au début soit nécessaire pour que les classes constituées se stabilisent. L'utilisateur peut également spécifier un critère, une mesure de la *variance intraclasse*, qui stoppe le processus itératif lorsqu'il est atteint.

Avantages et inconvénients de la méthode de k -Means [8]:

L'algorithme de k -Means est simple et compréhensible. Il est ainsi applicable à des données de grande taille, sans problème de mémoire et de temps de calcul.

Le nombre de classes k doit être fixé au début. La méthode ne permet pas de détecter des données bruitées ou la présence d'*outliers* et fournisse des classes convexes. Le problème principal de cette méthode est qu'elle ne permet pas de détecter des classes non globulaires. Ainsi, le résultat final de cette méthode dépend fortement du tirage initial des points représentant les centres des classes, d'où l'obtention d'une solution locale.

Il est donc conseillé d'effectuer plusieurs tirages et de comparer les différentes partitions correspondantes obtenues.

Pour atténuer la dépendance du tirage initial des centres des classes, une solution globale peut être trouvée par l'utilisateur en utilisant les algorithmes génétiques.

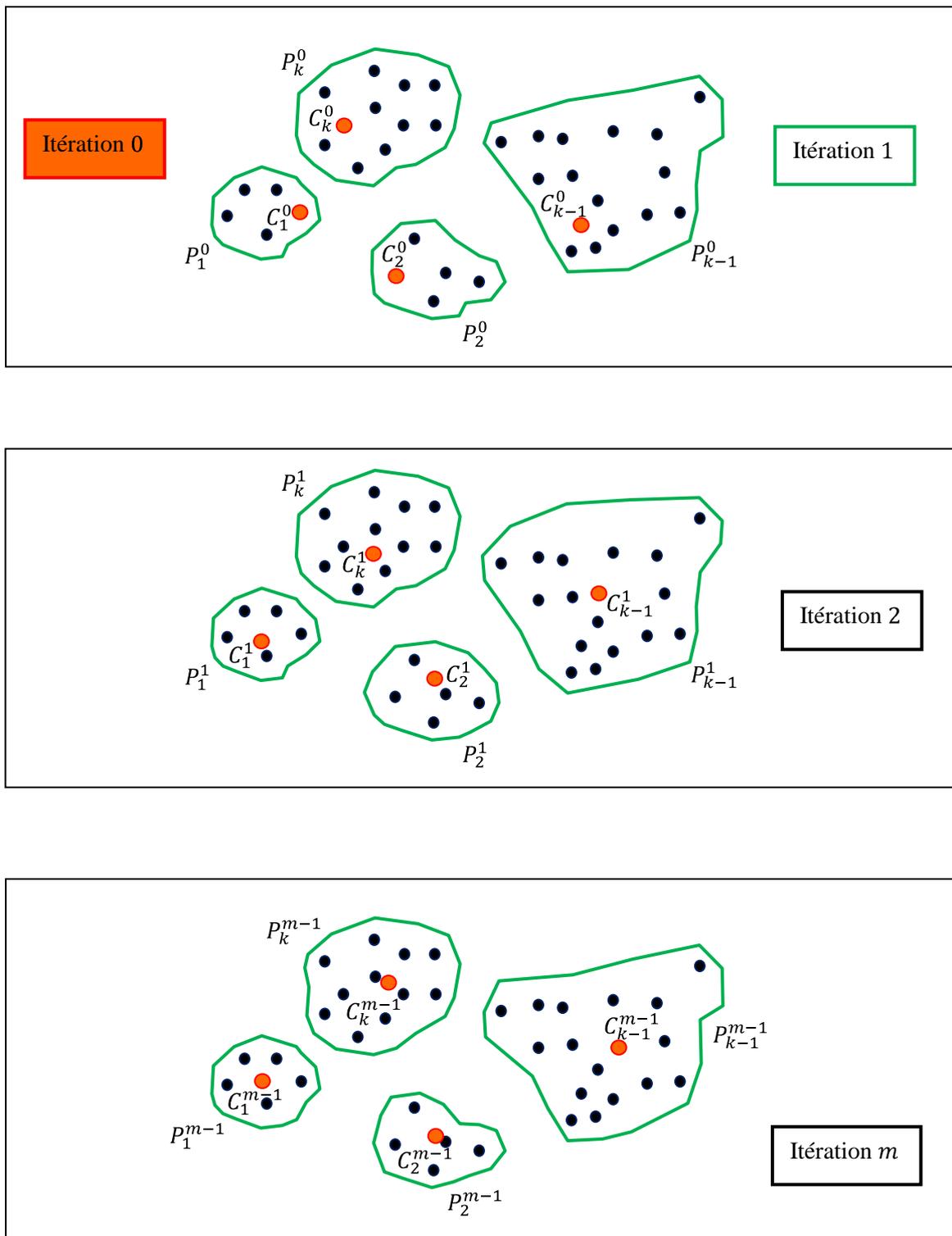


Figure I.9: Déroulement de l'algorithme de k -Means

I.3.2.2.3 Méthode ISODATA

L'algorithme ISODATA (Iterative Self-Organizing Data Analysis Techniques A) procède de façon équivalente que la méthode de k -Means.

Cette méthode permet de partitionner l'ensemble des données sans connaître à *priori* le nombre exact de classes. Elle introduit de nouveaux paramètres afin de modifier le nombre de classes au cours des itérations [10]:

- Lorsque la somme des inerties d'une classe devient plus grande qu'un seuil fixé, la classe est divisée en deux (éclatement d'une classe).
- Lorsque la distance entre les centres de gravité de deux classes devient inférieure à un autre seuil, les classes sont fusionnées.

Elle peut aussi nécessiter d'autres paramètres tels que :

- Le nombre minimal d'individus par classe (suppression d'une classe si son effectif est inférieur à un seuil fixé n_{min}).
- Le nombre maximum de paires de classes pouvant être fusionnées.
- Le nombre maximum d'itérations T_{max} .

Algorithme de classification par ISODATA

- 1) Affecter les n objets aux k classes au hasard.
 - 2) Calculer les centres des classes.
 - 3) Affecter chaque objets à la classe dont le centre est le plus proche.
 - 4) Eliminer les classes comportant moins de n_{min} objets, les individus alors orphelins sont affectés aux classes dont les centres sont plus proches.
 - 5) Calculer les nouveaux centres des classes.
 - 6) Si deux classes sont suffisamment proches, les rassembler.
 - 7) Si la dispersion des objets autour d'une classes est trop importante, éclater la classe en deux.
 - 8) Si le critère d'arrêt est vérifié (il n'y a pas eu des modifications ou un certain nombre T_{max} d'itérations a été atteint) on termine, sinon allez à 3.
-

I.3.2.2.4 La méthode des nuées dynamiques (Dynamical clustering)

La méthode des nuées dynamiques a été proposée par Diday (1972). Elle peut être considérée comme une généralisation de la méthode des centres mobiles.

Les différents sous-ensembles (ou classes), ne sont plus représentés par leurs centres de gravité uniquement, mais par leurs “noyaux”. Un noyau peut être un ensemble de q individus (les plus centraux). Elle consiste à trouver la partition optimale d’un ensemble d’individus en k classes. Le principe de la méthode est le suivant : on tire au hasard k noyaux parmi une famille de noyaux, puis chaque point de l’ensemble d’apprentissage est affecté au noyau dont il est plus proche. On obtient ainsi une partition en k classes dont on calcule les noyaux.

On recommence le processus avec les nouveaux noyaux et ainsi de suite jusqu’à ce que la qualité de la partition ne s’améliore plus [11].

Il est aisé de comprendre visuellement cette méthode. C’est la raison pour laquelle un simple exemple illustré doit permettre d’appréhender l’algorithme de classification de manière satisfaisante [12].

Supposons qu’on veuille constituer k classes à partir de l’ensemble initial des individus. La méthode consiste à constituer k groupes, au hasard, contenant un nombre q d’éléments fixé par l’utilisateur. Ces k groupes sont appelés noyaux car à partir d’eux vont se créer les k parties formant la partition de l’ensemble tout entier. Dans l’exemple de la figure (I.10), on choisit 3 noyaux de 2 éléments.

Il reste des éléments de l’ensemble qui ne sont pas dans les noyaux, il faut les associer à un des noyaux existants. Pour cela on affecte les éléments restants au noyau le plus proche (là encore, la distance entre un individu et un groupe est à choisir). C’est la phase d’expansion : les noyaux auxquels on a affecté les éléments restants forment maintenant une partition de l’ensemble total.

Dans l’exemple de la figure (I.10), l’élément α sera associé au noyau le plus proche, N_3 pour former la partie P_3 .

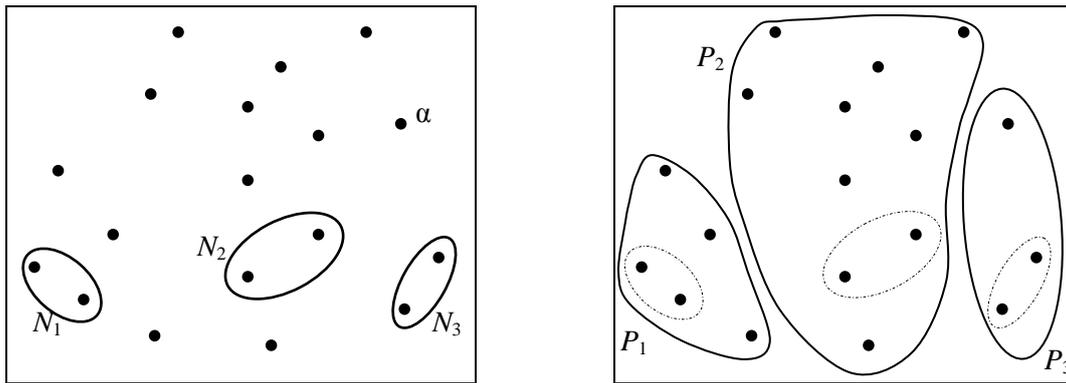


Figure I.10: Phase initiale et phase d'expansion

Maintenant qu'on dispose d'une partition, on veut s'assurer qu'elle ne dépend pas trop du choix initial des noyaux, c'est pourquoi on reforme des nouveaux noyaux : pour chaque partie (il y en a k en tout) on calcule les nouveaux noyaux. C'est la phase de contraction (figure (I.11)).

On répète les phases d'expansion et de contraction jusqu'à ce que la partition obtenue ne varie plus. On peut montrer que l'algorithme se termine et aboutit bien en un temps fini à une partition invariante. En effet, la similarité entre les individus de chaque partie augmente à chaque étape (contraction puis expansion) et s'arrêtera forcément de croître. Dans l'exemple, la seconde phase d'expansion aboutit à une partition (P'_1, P'_2, P'_3) qui n'évoluera plus si l'on refait une contraction et une expansion.

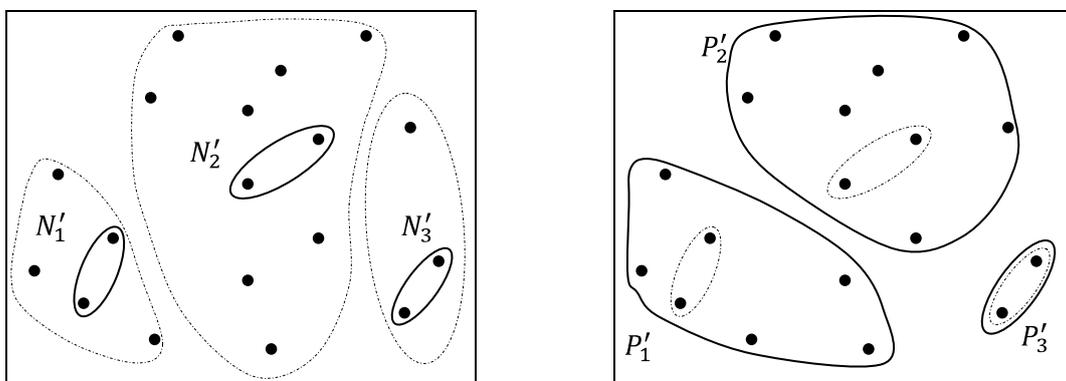


Figure I.11: Phase de contraction et d'expansion

Algorithme de classification par les nuées dynamiques

On répète les opérations suivantes jusqu'à l'obtention d'une classification invariante :

1. Choix des noyaux au hasard
 2. Attribution des individus restants à leur noyau le plus proche
 3. Choix des nouveaux noyaux par critère de distance avec les parties générées
 4. Attribution des individus restants à leur nouveau noyau le plus proche
-

Cette méthode a l'avantage de traiter rapidement de grands ensembles d'individus. Elle fournit une solution dépendant de la configuration initiale et nécessite le choix du nombre de classes. En général le nombre de classes est fixé par l'utilisateur et l'initialisation est faite par un tirage au hasard [28].

I.3.2.2.5 Classification floue (Fuzzy clustering)

Très tôt après l'introduction par L. Zadeh du concept d'ensemble flou, il s'est aperçu que la notion de classe utilisée en classification de données trouvait là son cadre d'expression tout naturel. En effet, on peut définir une classe comme un groupe d'individus présentant des similitudes communes [14]. Ces similitudes peuvent être plus ou moins fortes entre les individus d'une même classe, et d'autre part, un même individu peut présenter des similitudes avec des individus d'autres classes, si bien que son appartenance n'est pas localisée à une classe déterminée mais se trouve distribuée sur plusieurs classes, sans qu'il soit toujours possible de trancher d'une façon nette à quelle (unique) classe appartient l'individu en question (figure I.13).

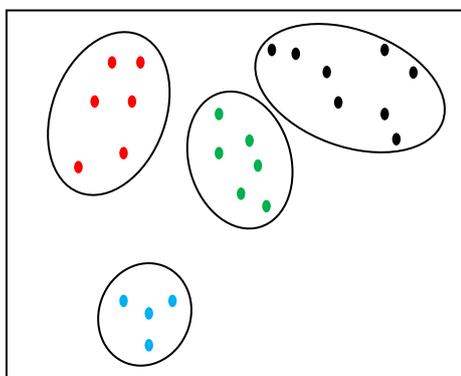


Figure I.12: Classification stricte

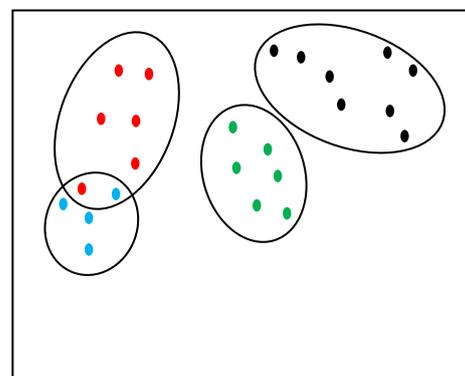


Figure I.13: Classification floue

La classification floue s'avère plus efficace que les méthodes traditionnelles considérées comme un cas particulier des méthodes floues. Elle permet une analyse plus exacte et surtout plus équitable. L'avantage principal par rapport à la classification classique est qu'un individu peut appartenir à plusieurs classes à la fois par un degré d'appartenance compris entre 0 et 1. Contrairement à la classification classique où l'appartenance d'un individu à une classe prend ses valeurs dans la paire [0, 1], il appartient ou non à l'ensemble et les classes sont considérées comme nettes. Il est aussi possible de traiter les points aberrants grâce à la classification floue. Un autre avantage de l'algorithme de classification floue est qu'il peut traiter les points ambigus, ce que ne peut pas faire une méthode de classification classique.

Un bon nombre de méthodes de classification floue sont en fait des extensions des méthodes classiques basées sur l'idée de partition floue. Les algorithmes les plus utilisés sont : l'algorithme des C-moyennes floues (fuzzy C-Means), où on suppose le nombre de classes connu, il a été étudié essentiellement par Bezdek [15]. L'algorithme ISODATA flou proposé par Dunn [16], qui est une version généralisée des C-moyenne floues, où le nombre de classes est obtenu automatiquement.

Algorithme de C-moyennes floues (Fuzzy C-Means ou FCM) :

L'algorithme C-moyennes floues FCM (développé par Dunn en 1974 et amélioré par Bezdek en 1981) est un algorithme de classification non supervisée, dérivé de l'algorithme de k -Means. Cet algorithme est basé sur la minimisation de la somme aux carrés des distances euclidiennes entre les objets et les centres des classes, qui minimise la fonction objectif suivante :

$$J_1(U, G, X) = \sum_{i=1}^k \sum_{j=1}^n (u_{ij})^2 d_M^2(x_j, g_i) \quad (1.7)$$

Avec $i = 1, \dots, k$ et $j = 1, \dots, n$

$d_M^2(x_j, g_i)$: représente la distance euclidienne entre le vecteur x_j de la classe et le centre de la classe g_i .

$U = [u_{ij}]$ représente une matrice de partition dure ($u_{ij} \in \{0,1\}$ et $\sum_{i=1}^k u_{ij} = 1$).

$G = \{g_i\}$ le vecteur des centres des classes.

$X = \{x_j\}, j = 1, \dots, n$ est l'ensemble des objets à classer.

Chaque objet x_j appartient à une seule classe et chaque classe k contient au moins un objet. Dunn (1974) a étendu en premier l'algorithme C-moyennes dur afin d'autoriser une partition floue avec la fonction suivante :

$$J_2(U, G, X) = \sum_{i=1}^k \sum_{j=1}^n (\mu_{ij})^2 d_M^2(x_j, g_i) \quad (I.8)$$

Sujet aux contraintes suivantes :

$$\mu_{ij} \in [0,1] \quad \forall i, j$$

$$\sum_{i=1}^k \mu_{ij} = 1 \quad \forall j$$

$$1 < \sum_{j=1}^n \mu_{ij} < n \quad \forall i$$

Cette fois-ci, $U = [\mu_{ij}]$ est appelée matrice de C-partition floue correspond à une partition floue en k classes. Donc, chaque objet x_j peut appartenir à plus d'une classe avec un degré d'appartenance prenant une valeur entre 0 et 1. Le but de cet algorithme est non seulement de calculer les centres des classes mais aussi l'ensemble des degrés d'appartenance des vecteurs aux classes.

Bezdek (1981) généralise $J_2(U, G, X)$ en un nombre infini de fonctions $J_m(U, G, X)$, où $1 \leq m < \infty$. Il existe toute une famille de fonctions associées à cet algorithme qui se distingue par des valeurs différentes prises par le paramètre réglable m appelé indice de flou « fuzzy index » et qui contrôle le degré de flou (degré de fuzzification) de la partition obtenue.

La nouvelle fonction, sujet aux mêmes contraintes que J_2 , est maintenant :

$$J_m(U, G, X) = \sum_{i=1}^k \sum_{j=1}^n (\mu_{ij})^m d_M^2(x_j, g_i) \quad (I.9)$$

Rappelons que plus le paramètre m à une valeur faible et plus l'algorithme se rapproche de l'algorithme de k -Means.

Pour la minimisation de la fonction $J_m(U, G, X)$, on utilise l'algorithme suivant :

Algorithme de classification par C-moyennes floues FCM

1. Initialisation des valeurs des g_i centres au hasard.
2. Calcul des fonctions d'appartenance :

$$\mu_{ij} = \left[\frac{d_M^2(x_j, g_i)}{\sum_{l=1}^k d_M^2(x_j, g_l)} \right]^{\frac{-1}{(m-1)}} \quad (\text{I. 10})$$

3. Calcul des centres des classes :

$$g_i = \frac{\sum_{j=1}^n \mu_{ij}^m x_j}{\sum_{j=1}^n \mu_{ij}^m} \quad (\text{I. 11})$$

4. Si $\|U^{c+1} - U^c\| < \varepsilon$ arrêter, sinon retour à l'étape 2 (c étant le nombre d'itérations et ε le critère d'arrêt compris entre 0 et 1).
-

On retrouve dans cet algorithme les mêmes inconvénients que pour k -Means au sujet des proportions et les formes des classes. Par contre, la complexité de cet algorithme est aussi linéaire, donc il est adapté à des données de grande taille.

I.4 Conclusion

Nous avons évoqué dans ce chapitre les différentes méthodes de classification automatique en insistant particulièrement sur l'approche non supervisée et plus particulièrement sur la méthode de k -Means et ses variantes. La multitude de méthodes laisse l'utilisateur perplexe devant le choix de l'une d'elles pour une application donnée parce qu'il n'y a pas de règle générale permettant de sélectionner une méthode particulière de classification pour un problème donné, donc ce choix s'appuie sur la connaissance de leurs capacités et surtout leurs limitations.

Le prochain chapitre sera consacré à la présentation de quelques notions sur la théorie des graphes qui seront utiles pour l'étude de la classification spectrale.

CHAPITRE II

Théorie des graphes

II.1 Introduction

La théorie des graphes est une branche des mathématiques, elle constitue un outil puissant de modélisation et de résolution de problèmes concrets. Actuellement, la théorie des graphes est devenue un outil important dans la recherche opérationnelle, le génie électrique (modélisation des circuits...), la programmation, la gestion des réseaux, la biologie, la classification de données,...

À l'origine, la théorie des graphes était présentée comme une curiosité mathématique. Le mathématicien suisse **Leonhard Euler** lors d'une de ses promenades nocturnes a voulu tracer un itinéraire circulaire dans la ville de Königsberg située sur le fleuve Pregel. Aujourd'hui, la ville est appelée Kaliningrad, et elle est devenue un centre industriel et commercial important de la Russie. Le fleuve Pregel traverse la ville, la divisant en quatre régions A, B, C, D comme le montre l'image de la figure suivante :

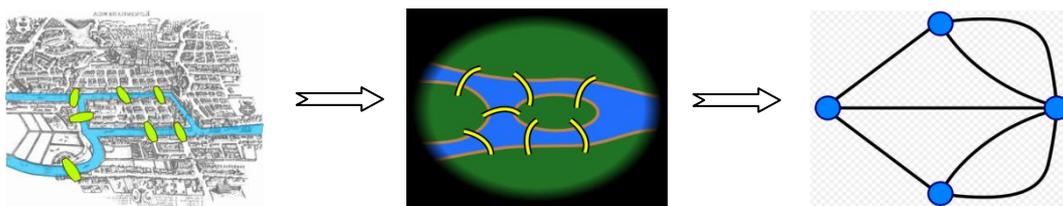


Figure II.1: Représentation des ponts de la ville de Königsberg [22]

On partant d'un point donné, Euler voulut visiter les sept ponts de cette ville (disposés selon le schéma ci-dessous) une seule fois seulement, puis retourner à son point de départ (sans traverser la rivière à la nage). Euler a montré que cela était impossible (chemin eulérien) [23].

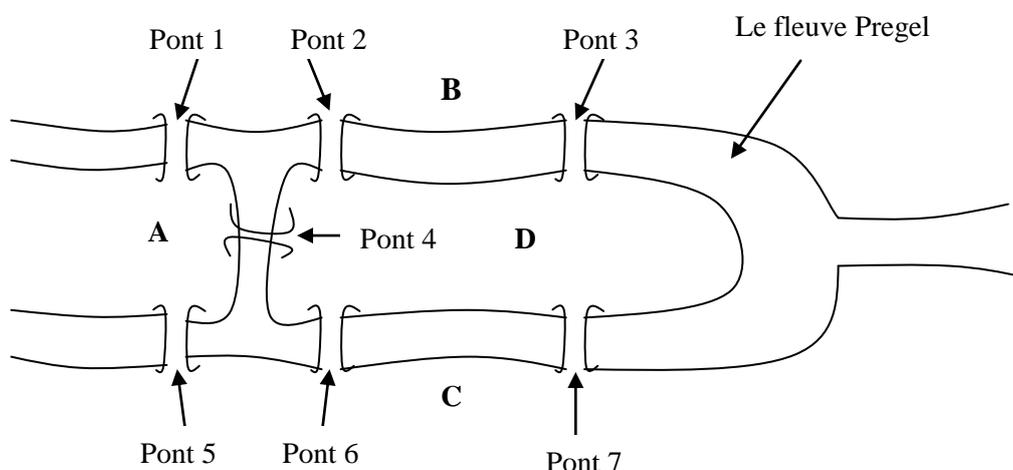


Figure II.2: Les points A, B, C et D sont des rives (régions) [23]

Un problème beaucoup plus simple aurait été de trouver un chemin de promenade tel que chacun des quatre secteurs de la ville ne soit visité qu'une seule fois, un tel parcours est appelé circuit hamiltonien (d'après le nom du mathématicien irlandais W.R.Hamilton).

II.2 Concepts fondamentaux de la théorie des graphes

II.2.1 Définition

Le mot « graphe » a été introduit par l'anglais J.J.Sylvester en 1822. En 1936, il est apparu le premier livre sur la théorie des graphes, écrit par D.Kning,

Un graphe est un dessin géométrique définit par la donnée d'un ensemble de points (appelées sommets ou nœuds), reliés entre eux par un ensemble de lignes ou de flèches (appelées arêtes ou arcs) [23].

La figure (II.3) suivante montre un exemple de graphe représentant un plan de circulation à sens unique d'une ville ou chaque localité est représentée par un point appelé « sommet » et chaque route par « un arc » orienté indiquant le sens de circulation.

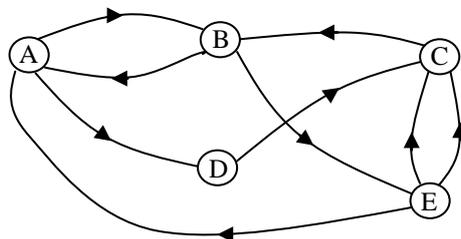


Figure II.3: Exemple d'un graphe

II.2.1.1 Graphe orienté

Un graphe orienté est un système formé d'un ensemble fini de sommets que l'on notera $\{x_1, x_2, \dots, x_n\}$ et d'un ensemble fini d'arcs reliant dans un ordre bien défini ces sommets, ou certains nombres d'entre eux noté $\{u_1, u_2, \dots, u_m\}$.

Exemple :

Le graphe G ci-contre est orienté :

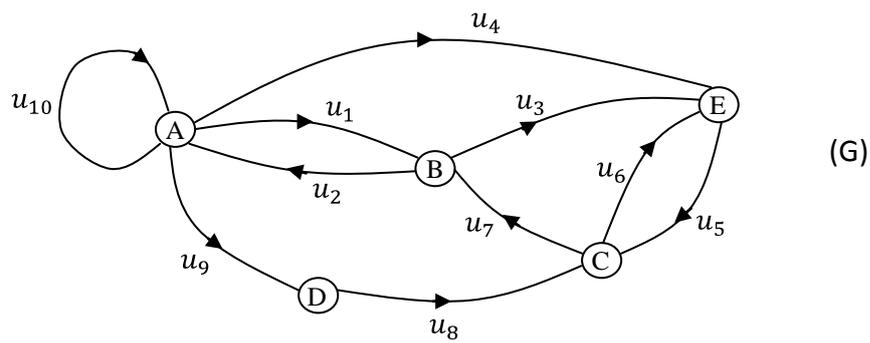
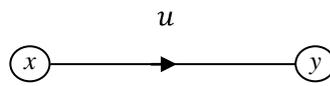


Figure II.4: Graphe orienté

Mathématiquement, un graphe orienté est représenté par le couple $G = (X, U)$, où :

- X est l'ensemble des sommets.
- U est l'ensemble des arcs.

On note un arc relie un sommet x au sommet y dans un graphe G par : $u = (x, y)$.



Si le graphe G contient n sommets, on dit alors que G est d'ordre n.

Chaque arc du graphe G relie respectivement deux sommets, le sommet de départ représente « l'extrémité initiale » de l'arc et le sommet d'arrivée représente « l'extrémité terminale ».

Autrement dit : un graphe orienté est défini par : $G (X, U, I, T)$ où :

- I est l'application extrémité initiale d'un arc définie par :

$$I: U \longrightarrow X$$

$$(x, y) \longmapsto I(x, y) = x$$

- T est l'application extrémité terminale d'un arc définie par :

$$T: U \longrightarrow X$$

$$(x, y) \longmapsto T(x, y) = y$$

Exemple :

Soit $u_1 = (A, B)$ un arc de l'ensemble des arcs U du graphe G ci-dessus :

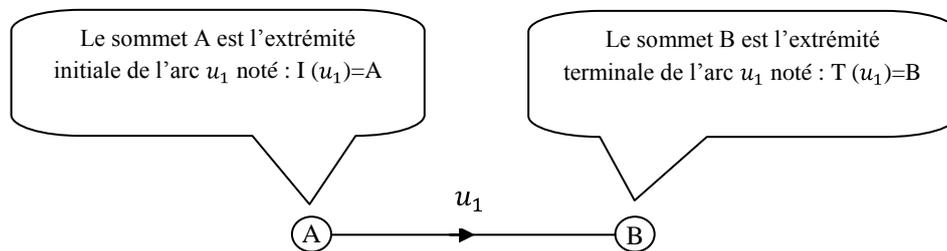
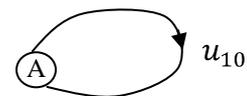


Figure II.5: Les extrémités d'un arc

□ **Remarque :** On appelle l'arc dont l'extrémité initiale est confondue avec l'extrémité terminale une boucle, notée : $u = (x, x)$

Exemple : Soit $u_{10} = (A, A)$ un arc de l'ensemble des arcs U du graphe G de la figure (II.4). Les deux extrémités l'initiale et la terminale de l'arc u_{10} sont confondues, c'est-à-dire :

$I(u_{10}) = A$, et $T(u_{10}) = A$. L'arc u_{10} est donc une boucle.



II.2.1.2 Graphe non orienté

Si on définit une relation sur un ensemble ou la notion d'ordre des sommets n'est pas importante, on représente la relation entre deux sommets par un arc non orienté appelé arête. On obtient alors un graphe non orienté, noté $G = (X, E)$. Tout comme les graphes orientés, on représente les sommets par des points (cercles) et les arêtes par des courbes simples reliant les deux points qui sont ses extrémités. Deux sommets reliés par une arête sont dits adjacents.

Exemple :

Considérons le plan de la ville de Königsberg modélisé par Euler sous forme d'un graphe non orienté $G = (X, E)$ où :

- $X = \{A, B, C, D\}$: représente les différentes rives.
- $E = \{AB, AC, AD, DB(2), DC(2)\}$: représente l'ensemble des ponts.

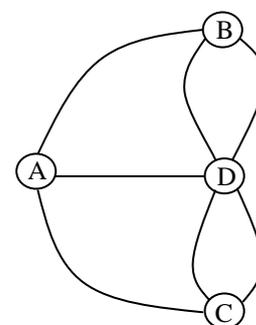
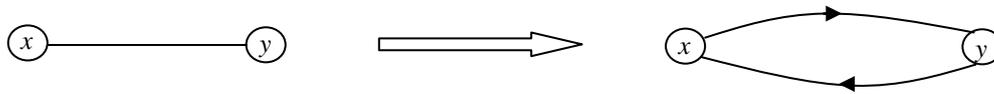


Figure II.6: Graphe non orienté

On dira qu'il existe une arête entre deux sommets s'il existe un pont permettant de relier deux rives.

- Une arête dont les extrémités sont confondues est une boucle.
- Une arête peut être transformée en deux arcs de sens différents.



II.2.1.3 Graphe simple et graphe multiple

Un graphe simple est un graphe sans boucles ni arcs (arêtes) multiples. Dans le cas contraire, c'est-à-dire, si des boucles ou des arcs (arêtes) multiples sont autorisés, on dira que le graphe est multiple.

Exemple :



Figure II.7: Arêtes multiples

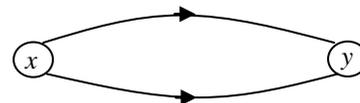


Figure II.8: Arcs multiples

Remarque : Deux arcs de sens opposés ne sont pas des arcs multiples.

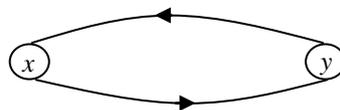


Figure II.9 Arcs non multiples

Exemple :

Le graphe d'Euler de la figure (II.10) est un graphe multiple, car des arêtes multiples relient les sommets D et B et les sommets D et C.

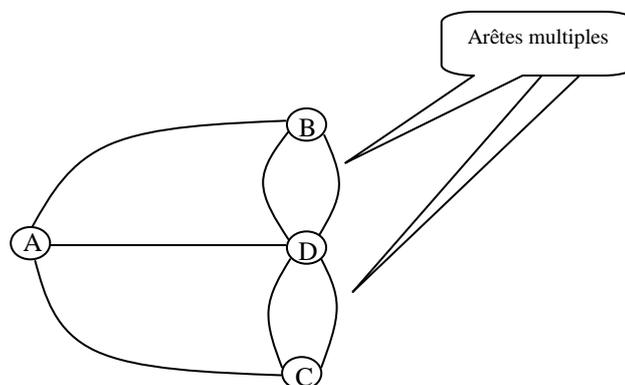


Figure II.10: Graphe d'Euler

La multiplicité d'un graphe

On définit la multiplicité d'un graphe orienté multiple par le nombre maximum d'arcs ayant la même extrémité terminale.

Soit P ce nombre, on dit alors que G est un P - graphe.

$$P = \text{Max} \{u \in U / I(u) = x \text{ et } T(u) = y \}$$

II.2.1.4 Prédécesseurs, successeurs et voisins

Soit le graphe de la figure (II.4) pour définir ces notions

1. Prédécesseurs :

L'ensemble des prédécesseurs d'un sommet x se définit par :

$$\Gamma^-(x) = \{ y \in X / \exists u \in U \text{ ou } I(u) = y \text{ et } T(u) = x \}.$$

Exemple : Partant des localités A et C, on peut atteindre la localité B par deux routes directes AB et CB. Les sommets A et C forment ainsi l'ensemble des prédécesseurs de B, noté $\Gamma^-(B) = \{A, C\}$.

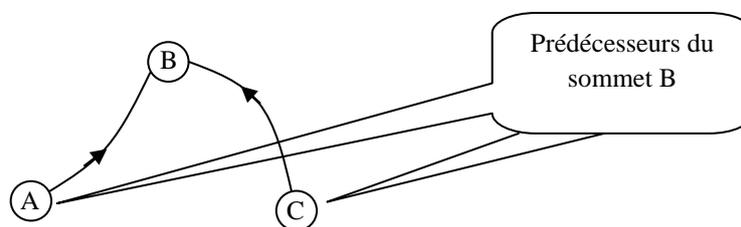


Figure II.11: Les prédécesseurs d'un sommet

2. Successeurs :

L'ensemble des successeurs d'un sommet x se définit par :

$$\Gamma^+(x) = \{ y \in X / \exists u \in U \text{ ou } T(u) = y \text{ et } I(u) = x \}.$$

Exemple :

Partant de la localité B, on peut atteindre directement les localités A et E, respectivement par deux routes BA et BE. Les sommets A et E forment ainsi l'ensemble des successeurs de B, noté $\Gamma^+(B) = \{A, E\}$.

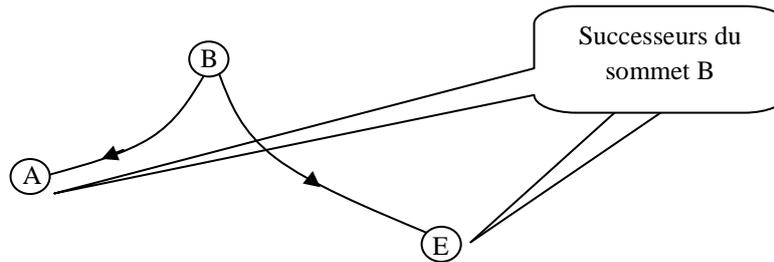


Figure II.12: Les successeurs d'un sommet

3. Les voisins :

L'ensemble des voisins d'un sommet x se définit par : $\Gamma(x) = \Gamma^-(x) \cup \Gamma^+(x)$.

Exemple :

L'ensemble des voisins du sommet B est égal à la réunion de l'ensemble de ses prédécesseurs et des ses successeurs, noté $\Gamma(B) = \{A, E, C\}$.

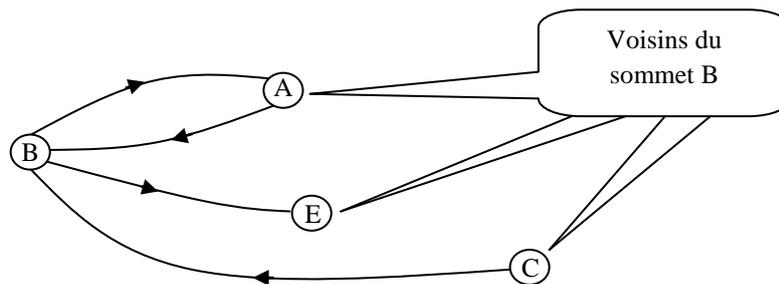


Figure II.13: Les voisins d'un sommet

II.2.1.5 Degré d'un sommet

Définition :

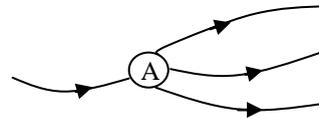
Soit $G = (X, U)$ un graphe orienté

- Le demi-degré extérieur d'un sommet x est égale au nombre d'arcs ayant le sommet x comme extrémité initiale, on dit aussi le nombre d'arcs incidents extérieurs au sommet x , on le note : $d_G^+(x) = |\{u \in U / I(u) = x\}|$.
- Le demi-degré intérieur d'un sommet x est égale au nombre d'arcs ayant le sommet x comme extrémité terminale, on dit aussi le nombre d'arcs incidents intérieurs au sommet x , on le note : $d_G^-(x) = |\{u \in U / T(u) = x\}|$.

- Le degré d'un sommet x est le nombre d'arcs ayant le sommet x comme extrémité initiale ou terminale, on dit aussi le nombre d'arcs adjacents au sommet x , on le note :

$$d_G(x) = d_G^+(x) + d_G^-(x).$$

Exemple : Considérons le sommet A suivant:



On a : un arc qui mène vers la localité A et 3 autres en partent, on définit ainsi la notation de degré d'un sommet dans un graphe comme suit :

- Le sommet A est l'extrémité initiale de 3 arcs, on dit alors que le demi-degré extérieur de A est 3, on le note $d_G^+(A) = 3$.
- De même le sommet A est l'extrémité terminale d'un seul arc. On dit dans ce cas que le demi-degré intérieur de A est 1 on le note $d_G^-(A) = 1$.
- La somme du demi-degré intérieur et du demi-degré extérieur du sommet A définit le degré du sommet A qu'on note $d(A) = 4$.

On note le plus grand degré des sommets d'un graphe par $\Delta(G)$ et le plus petit degré des sommets par $\gamma(G)$. Autrement dit : $Max d(x) = \Delta(G)$ et $Min d(x) = \gamma(G)$.

Sommet isolé :

On appelle un sommet dont le degré est égale à zéro ($d_G(x) = 0$) un sommet isolé.

Sommet pendent :

On appelle un sommet dont le degré est égale à un : ($d_G(x) = 1$) un sommet pendent.

II.2.1.6 Cheminement dans un graphe

Les cheminements dans la théorie des graphes sont de quatre types : la chaîne, le cycle, le chemin et le circuit.

Soit $G = (X, U)$ un graphe

1. Chaîne : Une chaîne joignant deux sommets x_0 et x_k dans un graphe G est une suite de sommets reliés par des arêtes (arcs) tel que, deux sommets successifs ont une arête (un arc) commune (commun). On la note : (x_0, x_1, \dots, x_k) . Les sommets x_0 et x_k sont les extrémités de la chaîne.

Une chaîne est dite simple si on passe une seule fois par ses arcs (arêtes).

2. Chemin : Un chemin du sommet x_0 à x_k dans un graphe G est une suite de sommets reliés successivement par des arcs orientés dans le même sens. Autrement dit, deux sommets successifs d'un chemin sont respectivement extrémité initiale et terminale du même arc, exception faite au premier, et au dernier sommet. On le note (x_0, x_1, \dots, x_k) .

Un chemin est dite simple si on passe une seule fois par ses arcs.

3. Cycle : Un cycle est une chaîne simple dont les deux extrémités coïncident (x_0 coïncide avec x_k), on le note : $(x_0, x_1, \dots, x_k = x_0)$.

4. Circuit : Un circuit est un chemin dont les deux extrémités sont confondues ; on le note par : $(x_0, x_1, \dots, x_k = x_0)$.

Exemple : Soit le graphe G de la figure (II.14)

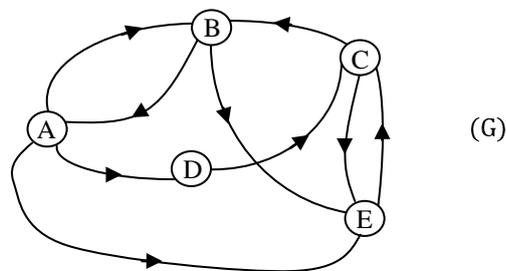


Figure II.14

- Les suites des sommets suivantes (A, B, E) , (A, E, B, A, E) sont des chaînes joignant les sommets A et E . La seconde chaîne n'est pas simple car l'arc (AE) est parcouru deux fois.
- Les suites des sommets suivantes (A, D, C, E) , (A, D, C, E, C, E) sont des chemins joignant les sommets A et E . Le second chemin n'est pas simple car l'arc (CE) est parcouru deux fois.
- La suite des sommets suivante (A, B, C, D, A) est un cycle.
- La suite des sommets suivante (A, D, C, B, A) est un circuit.

Remarque :

- Une chaîne (cycle-chemin-circuit) est dite élémentaire si on passe une seule fois par ses sommets (tous les sommets sont différents).
- La notion de chaîne et de cycle ne respecte pas l'orientation des arcs, par contre celle de chemin et de circuit la respecte.

Les cheminements remarquables :

- Un chemin eulérien est un chemin simple qui passe une et une seule fois par chaque arc du graphe.
- Une chaîne eulérienne est une chaîne empruntant une fois et une fois seulement chaque arête du graphe.
- Un cycle eulérien est un cycle passant une et une seule fois seulement par toutes les arêtes d'un graphe et revenant au sommet de départ.
- Un chemin hamiltonien (chaîne hamiltonienne) est un chemin (une chaîne) qui passe une et une seule fois seulement par chaque sommet du graphe.
- Cycle hamiltonien est un cycle passant une et une seule fois seulement par tous les sommets d'un graphe et revenant au sommet de départ.

II.2.2 Structure d'un graphe

Pour caractériser la structure d'un graphe, il est possible de rechercher des parties remarquables du graphe, en restreignant soit l'ensemble des sommets (sous-graphe), soit l'ensemble des arêtes (graphe partiel), ou bien les sommets et les arêtes (sous-graphe partiel) [2]. Soit le graphe $G = (X, U)$ de la figure (II.15), Et soit $\mathcal{A} = (A, D, C)$ et $\mathcal{W} = \{e_1, e_2, e_5\}$.

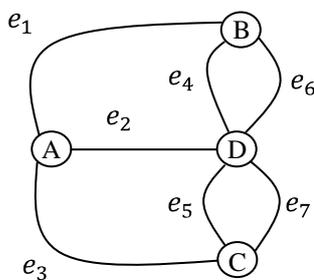


Figure II.15 : Graphe (G)

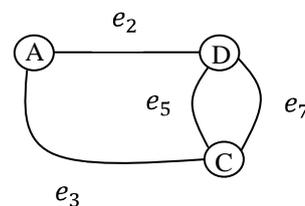


Figure II.16: $(G_{\mathcal{A}})$ Sous graphe

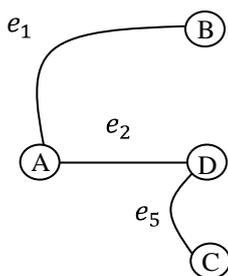


Figure II.17: Graphe partiel $(G_{\mathcal{W}})$

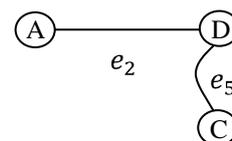


Figure II.18: Sous graphe partiel $(G_{\mathcal{A},\mathcal{W}})$

1. **Sous graphe** : Le sous graphe engendré par \mathcal{A} est le graphe $G_{\mathcal{A}} = (\mathcal{A}, E_{\mathcal{A}})$, avec $E_{\mathcal{A}} = \{e_2, e_3, e_5, e_7\}$, figure (II.16).
2. **Graphe partiel** : $(G_{\mathcal{W}})$ est le graphe partiel engendré par \mathcal{W} , figure (II.17).
3. **Sous graphe partiel** : $(G_{\mathcal{A},\mathcal{W}})$ est le sous graphe engendré par \mathcal{A} et \mathcal{W} , figure (II.18).

II.2.3 Les graphes particuliers

Les graphes particuliers ont un grand intérêt par leurs propriétés particulières ainsi pour les problèmes qu'ils posent, les plus connus sont les graphes : complet, complémentaire, planaire, biparti, eulérien, hamiltonien, arbres, arbre recouvrant minimal et arborescences [23].

II.2.3.1 Graphe complet

On appelle graphe complet un graphe dont tous les sommets sont adjacents.

Si le graphe est simple et complet, d'ordre n , on le note K_n



Figure II.19 Graphe complet et graphe non complet

- Les sommets A et C dans le graphe (G1) de la figure (II.19) ne sont pas adjacents, le graphe (G1) est donc non complet.
- Les sommets du graphe (G2) de la figure (II.19) sont tous adjacents, d'où le graphe (G2) est complet.
- Le graphe (G2) de la figure (II.19) est un graphe simple et complet, d'ordre 4, c'est un K_4 .

II.2.3.2 Graphe complémentaire

À un graphe simple $G = (X, U)$, on peut définir un graphe complémentaire $\bar{G} = (X, \bar{U})$ comme suit : $u \in \bar{U} \Leftrightarrow u \notin U$. C'est-à-dire, une arête (arc) appartient au graphe complètement (\bar{G}) si elle (il) n'appartient pas au graphe initiale G .

La figure (II.20) montre un exemple de graphe simple G et son graphe complémentaire \bar{G} . $G \cup \bar{G}$ est un graphe simple complet, donc c'est un K_4 .

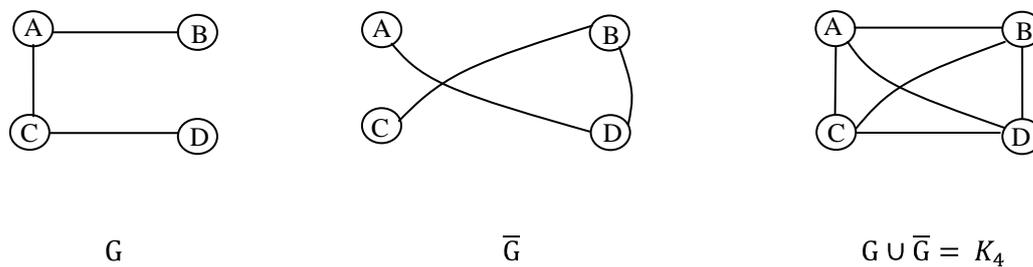


Figure II.20: Graphe complémentaire

II.2.3.3 Graphe planaire

Un graphe est dit planaire si on peut le dessiner sur un plan de telle façon que les arêtes ne se coupent pas en dehors de leurs extrémités. Ce type de graphe est particulièrement utilisé dans les problèmes de circuits imprimés.

Exemple :

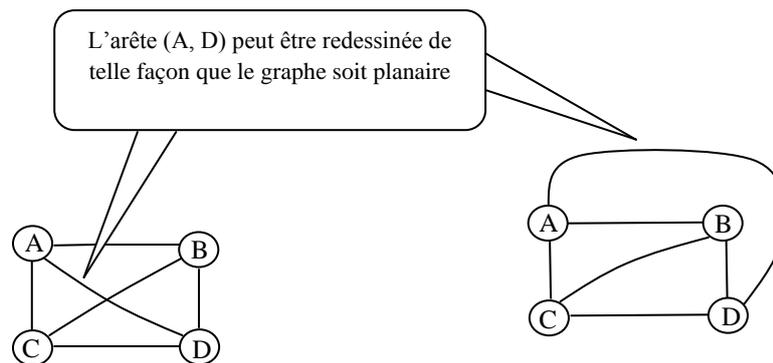


Figure II.21: Graphe planaire

On définit dans un graphe planaire:

- **Une face** : Une face d'un graphe planaire est par définition une région du plan limitée par des arêtes de telle sorte que deux points arbitraires, dans cette région, reliés par une arête ne rencontrent ni sommet, ni arête.
- **Une frontière** : La frontière d'une face est l'ensemble des arêtes qui l'entoure.
- **Face infini** : Une face infinie est une face illimitée, elle n'admet pas de contour et elle est unique. Les autres faces sont finies.
- Deux faces sont dites adjacentes si leurs frontières ont une arête commune.

À titre d'exemple, on considère le graphe planaire (G) de la figure (II.22):

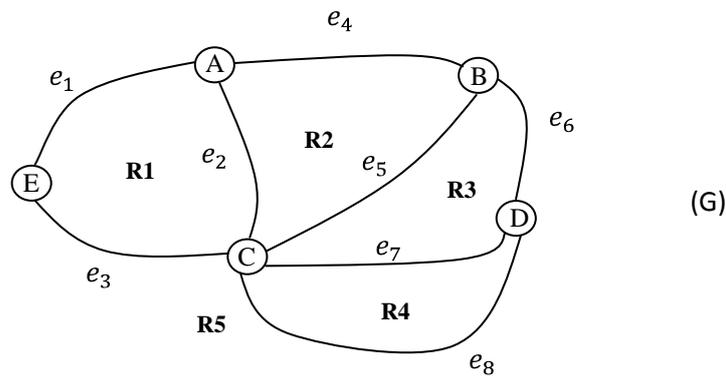


Figure II.22: Faces et frontières dans un graphe

- Les régions : **R1**, **R2**, **R3** et **R4** sont des faces finies.
- La région **R5** est une face infinie.
- Les arêtes e_1, e_2 et e_3 sont les frontières de la face **R1**.
- Les faces **R1** et **R2** sont adjacentes.

II.2.3.4 Graphe biparti

Un graphe est biparti si l'ensemble de ses sommets peut être reparti en deux classes X_1 et X_2 telles que, deux sommets de la même classe ne soient pas adjacents. On le note parfois $G = (X_1, X_2, U)$.

La figure (II.23) illustre la notion du graphe biparti.

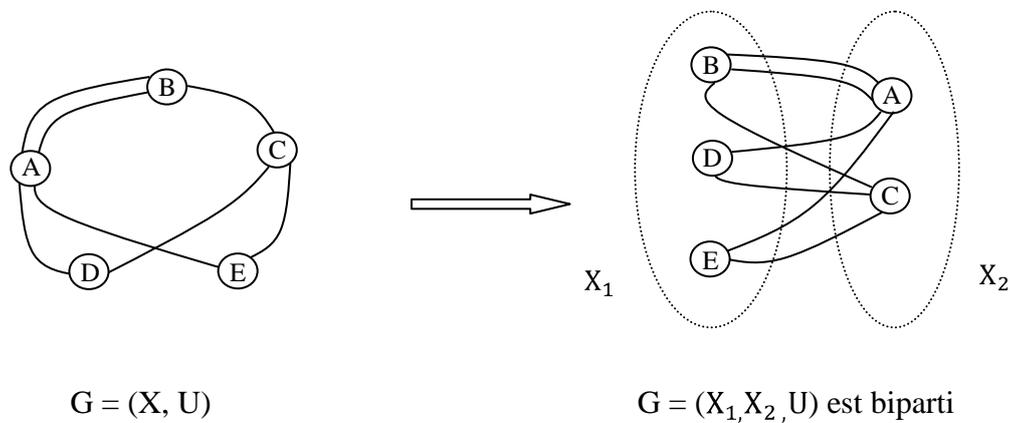


Figure II.23: Graphe biparti

- Un graphe G est biparti complet, si tout sommets de X_1 est adjacent à tout sommets de X_2 .
- Si de plus le graphe G est simple, alors G est un graphe simple biparti-complet, on le note K_{p-q} avec $p = |X_1|$ et $q = |X_2|$.

Le graphe G de la figure (II.24) est un graphe simple biparti-complet. Il est un K_{3-2} .

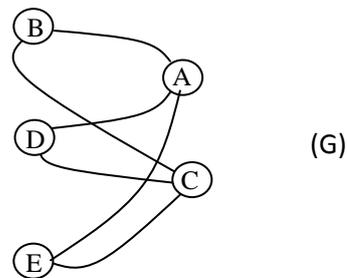


Figure II.24: Graphe simple, biparti-complet

II.2.3.5 Graphe eulérien

Un graphe est dit eulérien s'il est possible de trouver un cycle passant une et une seule fois par toutes ses arêtes. Plus simplement, on peut dire qu'un graphe est eulérien s'il est possible de dessiner le graphe sans lever le crayon.

Le graphe de la figure (II.25) est un graphe eulérien.

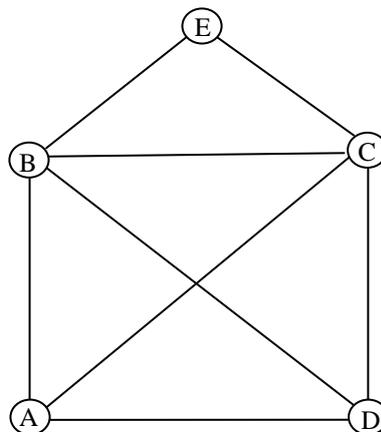


Figure II.25: Graphe eulérien

II.2.3.6 Graphe hamiltonien

Un graphe est dit hamiltonien s'il est possible de trouver un cycle passant une et une seule fois par tous ses sommets.

Le graphe de la figure (II.26) est un graphe hamiltonien.

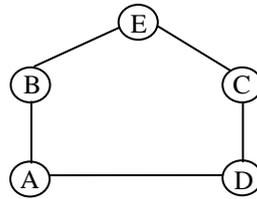


Figure II.26: Graphe hamiltonien

II.2.3.7 Arbres

Les arbres sont des graphes particulièrement importants. Un arbre peut être défini de plusieurs façons équivalentes : comme un graphe connexe sans cycle, ou bien comme un graphe tel que deux sommets distincts sont reliés par un et un seul chemin. Un arbre est donc un graphe connexe ayant un nombre d'arêtes (arcs) minimal. Si on enlève une (un) arête (arc) on déconnecte le graphe.

Un graphe non connexe dont chaque composante connexe est un arbre, c'est-à-dire sans cycle est appelé forêt.

Exemples :

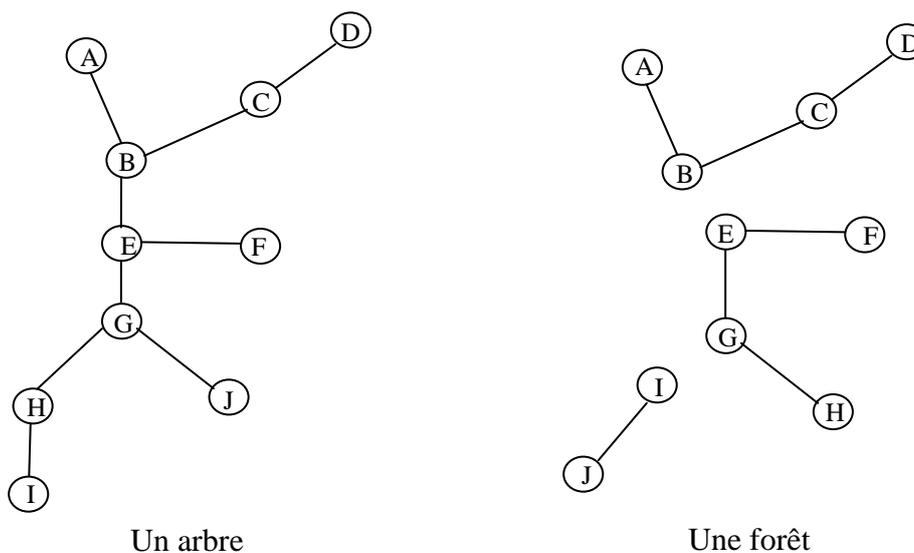


Figure II.27 : Exemples d'arbre et de forêt

II.2.3.8 Arbre recouvrant minimal

Etant donné un graphe non orienté et connexe. Un arbre recouvrant de ce graphe est un arbre qui relie tous les sommets ensemble [22]. Un graphe peut comporter plusieurs arbres recouvrant différents. On peut associer un poids à chaque arête (un nombre qui représente le coût de cette arête). Un arbre recouvrant minimal est un arbre recouvrant dont la somme des poids est plus petite ou égale à celles de tous les autres arbres recouvrant du graphe.

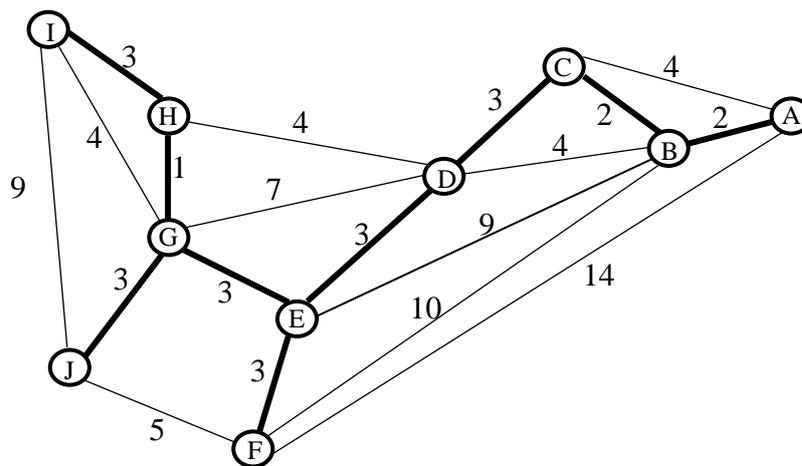


Figure II.28: Arbre recouvrant minimal

II.2.3.9 Arborescence

Une arborescence est un graphe sans cycles disposant d'un sommet nommé racine tel que pour tout sommet il existe un et un seul chemin orienté de cette racine à ce sommet. Cette racine est unique, autrement dit, un graphe $G = (X, U)$, avec $|X| = n \geq 2$ sommets est une arborescence de racine s si :

- G est un arbre.
- s est une racine de G .

Exemple : figure (II.29)

Soit $Ar = (X, U)$ un arbre admettant le sommet A comme une racine alors, Ar est une arborescence.

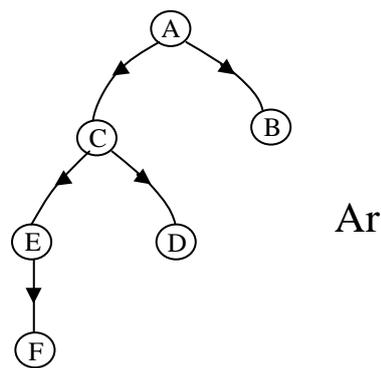


Figure II.29: Arborescence

II.3 Représentation matricielle d'un graphe [23]

Certains graphes peuvent être complexes, notamment s'ils contiennent plusieurs arcs ou plusieurs sommets. Un nouveau concept (celui des matrices) est introduit pour palier à certaines difficultés rencontrées dans la théorie des graphes.

À un graphe $G = (X, U)$ contenant n sommets et m arcs, c'est-à-dire : $|X| = n$ et $|U| = m$, on associera trois types de matrices. On définira ces matrices dans le cas du graphe G de la figure (II.30) ci-dessous :

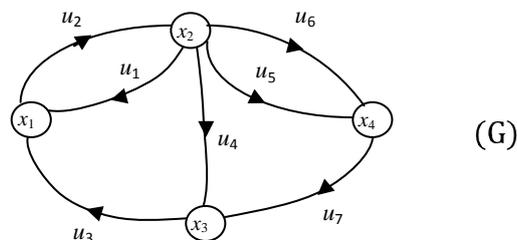


Figure II.30

II.3.1 Matrice d'adjacence

La matrice d'adjacence d'un graphe $G = (X, U)$ est une matrice $n \times n$; ses éléments prennent deux valeurs 1 ou 0. Chaque ligne et chaque colonne correspond à un sommet du graphe. Ainsi chaque élément de la matrice indique la relation qui existe entre deux sommets :

- (1) signifie que les deux sommets sont reliés par un arc.
- (0) signifie que les deux sommets ne sont pas reliés par un arc.

Exemple : Pour le graphe G de la figure (II.30), la matrice d'adjacence est:

$$A = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c|c|c|c} x_1 & x_2 & x_3 & x_4 \\ \hline 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \end{array}$$

$u = (x_1, x_2)$ est un arc du graphe G ; c'est à dire x_1 est une extrémité initiale de l'arc u et x_2 son extrémité terminale, donc l'élément a_{12} (l'élément de la première ligne et de la deuxième colonne) de la matrice d'adjacence du graphe G sera égal à 1. Par contre il n'existe pas d'arc ayant x_1 comme extrémité initiale et x_3 comme extrémité terminale ainsi l'élément a_{13} (élément de la première ligne et de la troisième colonne) de la matrice d'adjacence sera égal à 0.

II.3.2 Matrice associée

La matrice associée d'un graphe $G = (X, U)$ est une matrice $n \times n$, où chaque ligne et chaque colonne correspondent à un sommet du graphe. Chaque élément indique le nombre de relations entre deux sommets, c'est-à-dire le nombre d'arcs (arêtes) reliant deux sommets.

Pour le graphe G de la figure (II.30), la matrice associée est:

$$A = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c|c|c|c} x_1 & x_2 & x_3 & x_4 \\ \hline 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 2 \\ \hline 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \end{array}$$

Du graphe de la figure (II.30) on décèle 2 arcs de la forme (x_2, x_4) ; c'est-à-dire ayant x_2 comme extrémité initiale et x_4 comme extrémité terminale. Ainsi l'élément a_{24} correspondant sera égal à 2.

Remarque : De la matrice associée, on déduit:

- La somme des valeurs d'une ligne détermine le demi-degré extérieur ($d_G^+(x)$) du sommet x correspondant.
- La somme des valeurs d'une colonne détermine le demi-degré intérieur ($d_G^-(x)$) du sommet x correspondant.

II.3.3 Matrice d'incidence aux arcs

La matrice d'incidence aux arcs d'un graphe $G = (X, U)$ est une matrice à n lignes et m colonnes. Ses éléments prennent les valeurs 1 ou 0 ou -1. Chaque ligne de la matrice est associée à un sommet et chaque colonne à un arc. Chaque élément de la matrice indique la relation entre un sommet et un arc comme suit :

- (+1) signifie que le sommet est une extrémité initiale de l'arc.
- (-1) signifie que le sommet est une extrémité terminale de l'arc.
- (0) signifie qu'il n'existe pas de relations entre le sommet et l'arc.

La matrice d'incidence ne convient pas pour les graphes avec boucles.

Pour le graphe G de la figure (II.30), composé de 7 arcs, la matrice d'incidence aux arcs est:

$$A = \begin{array}{c|ccccccc} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \\ \hline x_1 & -1 & +1 & -1 & 0 & 0 & 0 & 0 \\ x_2 & +1 & -1 & 0 & +1 & +1 & +1 & 0 \\ x_3 & 0 & 0 & +1 & -1 & 0 & 0 & -1 \\ x_4 & 0 & 0 & 0 & 0 & -1 & -1 & +1 \end{array}$$

Remarque : De la matrice d'incidence aux arcs, on a :

- Le nombre des valeurs égales à (+1) d'une ligne donne le demi-degré extérieur ($d_G^+(x)$) du sommet x correspondant.
- Le nombre des valeurs égales à (-1) d'une ligne donne le demi-degré intérieur ($d_G^-(x)$) du sommet x correspondant.

II.4 Connexité dans un graphe

De manière générale, la connexité dans un graphe traduit les connexions d'un ensemble complexe et exprime les relations entre ses éléments [23].

II.4.1 Connexité

II.4.1.1 Notion de connexité

Deux sommets x et y dans un graphe ont une relation de connexité s'il existe une chaîne entre x et y .

Exemple : Soit le graphe (G) de la figure (II.31) suivante :

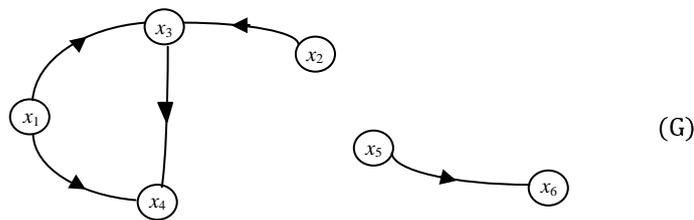


Figure II.31

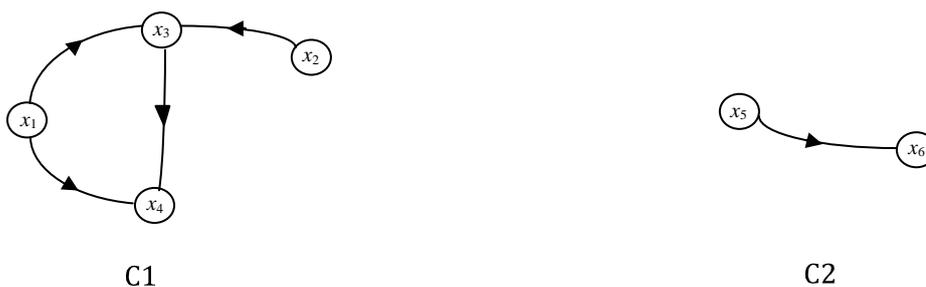
- Il existe une chaîne entre le sommet x_1 et x_2 notée $C = (x_1, x_3, x_2)$, par conséquent x_1 et x_2 ont une relation de connexité.
- Il n'existe pas de chaîne entre le sommet x_1 et x_5 , x_1 et x_5 n'ont alors pas une relation de connexité.

II.4.1.2 Composantes connexes

On appelle composante connexe un ensemble de sommets, qui ont deux à deux une relation de connexité. De plus, tout sommet en dehors de la composante n'a pas de relation de connexité avec les sommets de la composante.

Exemple :

Dans le graphe G de la figure (II.31), les sommets x_1, x_2, x_3, x_4 ont deux à deux une relation de connexité, donc l'ensemble $\{x_1, x_2, x_3, x_4\}$ forme ainsi la première composante connexe, on la note C1. L'ensemble $\{x_5, x_6\}$ forme la deuxième composante connexe C2. On constate que les sommets de C1 n'ont pas de relation de connexité avec les sommets de C2.



II.4.1.3 Graphe connexe

Un graphe $G = (X, U)$ est dit graphe connexe si tous ses sommets ont deux à deux une relation de connexité, autrement dit, si G contient une seule composante connexe.

Exemple : Le graphe G de la figure (II.32) contient une seule composante connexe $C = \{x_1, x_2, x_3, x_4, x_5\}$, donc le graphe G est connexe.

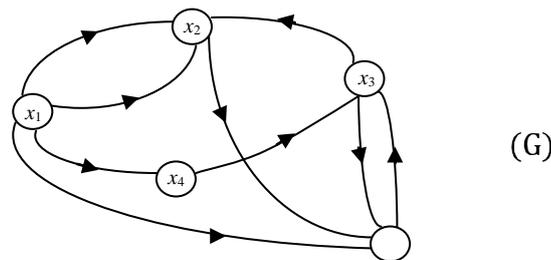


Figure II.32: Graphe connexe

Le graphe G de la figure (II.31) contient deux composantes connexes $C1 = \{x_1, x_2, x_3, x_4\}$ et $C2 = \{x_5, x_6\}$, donc le graphe G n'est pas connexe.

II.4.2 Forte connexité

II.4.2.1 Notion de forte connexité

Deux sommets x et y ont une relation de forte connexité s'il existe un chemin de x à y et un chemin de y à x .

Exemple : Soit le graphe $G = (X, U)$ suivant :

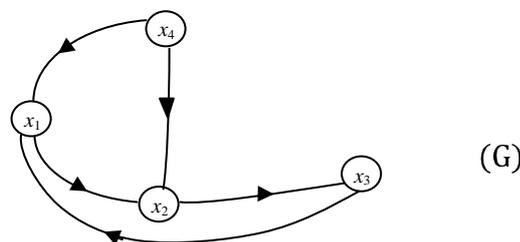


Figure II.33

- On a un chemin reliant le sommet x_1 au sommet x_3 et un chemin reliant le sommet x_3 au sommet x_1 alors x_1 et x_3 ont une relation de forte connexité.
- On a un chemin reliant le sommet x_4 au sommet x_3 , mais on a pas de chemin reliant le sommet x_3 au sommet x_4 alors x_4 et x_3 n'ont pas une relation de forte connexité.

II.4.2.2 Composantes fortement connexes

On appelle composante fortement connexe un ensemble de sommets, qui ont deux à deux une relation de forte connexité. De plus, tout sommet en dehors de la composante n'a pas de relation de forte connexité avec aucun élément de la composante.

Exemple : Soit le graphe $G = (X, U)$ suivant :

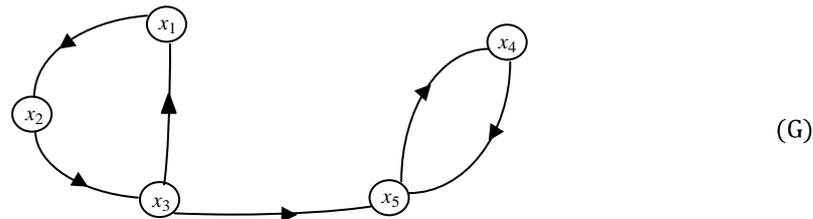


Figure II.34

Les sommets x_1, x_2, x_3 du graphe (G) ont deux à deux une relation de forte connexité, donc l'ensemble $\{x_1, x_2, x_3\}$ forme ainsi une composante fortement connexe, on la note C1.

Les sommets x_4, x_5 du graphe (G) ont deux à deux une relation de forte connexité, donc l'ensemble $\{x_4, x_5\}$ forme ainsi une composante fortement connexe, on la note C2.



De plus, on constate que les sommets de C1 n'ont pas de relation de forte connexité avec les sommets de C2.

II.4.2.3 Graphe réduit

On appelle graphe réduit du graphe $G = (X, U)$, le graphe $G_r = (X_r, U_r)$ dont les sommets représentent les composantes fortement connexes C_i du graphe G et s'il existe un arc (x, y) dans le graphe G avec le sommet x appartenant à C_i et le sommet y appartenant à C_j , alors il existera un arc (C_i, C_j) dans le graphe réduit G_r .

Exemple : Au graphe G de la figure (II.34) on fait correspondre un graphe réduit noté Gr représenté comme suit :

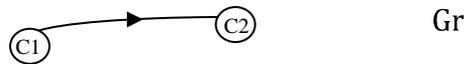


Figure II.35: Graphe Réduit

II.4.2.4 Graphe fortement connexe

Un graphe $G = (X, U)$ est dit graphe fortement connexe si tous ses sommets ont deux à deux une relation de fortement connexité, autrement dit, si G contient une seule composante fortement connexe.

Exemple : Le graphe G de la figure (II.34) contient deux composantes fortement connexe $C1 = \{x_1, x_2, x_3\}$ et $C2 = \{x_4, x_5\}$, le graphe G alors n'est pas fortement connexe.

Soit le graphe $G' = (X, U)$, avec $G' = G \cup \{(x_5, x_3)\}$.

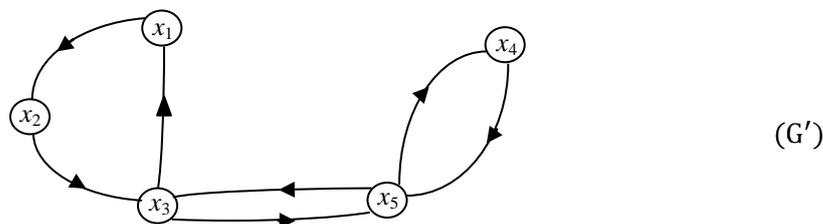


Figure II.36

Le graphe G' obtenu contient une seule composante fortement connexe $C = \{x_1, x_2, x_3, x_4, x_5\}$, alors G' est fortement connexe.

II.5 Domaine d'applications

Les applications liées aux graphes (et par conséquent à la théorie des graphes) sont très nombreuses et elles touchent plusieurs domaines. On peut citer quelques exemples, en :

- Économie et gestion (planning de livraisons, gestion de flots, ordonnancement, réseaux des informations dans une organisation, ...)

- Cartographie (réseaux routiers, réseaux internet, réseaux de téléphones, réseaux électriques, chemins de fer, ...)
- Chimie et biologie (modélisation de molécules, ADN, ...)
- Sciences sociales (généalogie, conflits, ...)
- Linguistique (grammaire, compilation, ...)
- Intelligence artificielle (comportement, certains jeux dames, échecs, ...)
- Technologie (ordonnancement des opérations de montage et démontage d'un ensemble technologique, ...)

II.6 Conclusion

Les graphes se proposent comme une méthode de pensée qui permet de modéliser une grande variété de problèmes en représentant les objets liés au problème par des sommets et dont les relations entre les objets sont représentés par des arcs (arêtes). L'étude des relations entre les objets peuvent être alors obtenue à partir de l'étude de la structure du graphe. Cette étude consiste à extraire certaines propriétés tels que le degré d'un sommet, la connexité dans un graphe ...etc.

Le prochain chapitre portera sur la présentation d'une méthode de classification automatique non supervisée appelée classification spectrale basée sur la théorie spectrale des graphes.

CHAPITRE III



Classification spectrale

III.1 Introduction

Dans ces dernières années, la classification spectrale (Spectral clustering) est devenue l'une des méthodes modernes de classification non supervisée les plus populaires. Cette dernière est de plus en plus usitée, à la fois en raison de son efficacité, et de sa simplicité relative d'implémentation. Elle peut être résolue efficacement par des méthodes d'algèbre linéaire. Par rapport à des algorithmes classiques comme celui de k -Means, elle offre de nombreux avantages fondamentaux. Les résultats obtenus par la classification spectrale sont souvent mieux que ceux obtenus par les approches traditionnelles.

Ce chapitre s'intéresse à la méthode de classification spectrale et à sa mise en œuvre. Tout d'abord, nous rappellerons quelques définitions liées à la classification spectrale, ainsi les différents graphes de similarité tels que le graphe de voisinage ε , le graphe des k plus proches voisins et le graphe totalement connecté. Ensuite, nous décrirons les différentes matrices laplaciennes caractérisant un graphe et leurs propriétés de base. Des algorithmes de classification spectrale et leurs différentes approches seront présentés, ainsi, leurs fondements théoriques. Puis, nous verrons comment le choix des paramètres pour la classification spectrale influe sur ses résultats. Enfin, nous terminerons ce chapitre par quelques avantages et inconvénients de classification spectrale.

III.2 Notions liées à la classification spectrale

Soient un ensemble d'objets x_1, x_2, \dots, x_n , où chaque objet est représenté par un ensemble de N paramètres, et une notion de similarité $s_{ij} \geq 0$ entre toutes les paires des vecteurs de données x_i et x_j . L'objectif de la classification est de diviser les vecteurs de données en plusieurs classes de telle manière que les vecteurs d'une même classe sont considérés « similaires » tandis que les vecteurs situés dans des classes distinctes soient définis comme « différents ». Une meilleure façon de représenter les données est sous forme de *graphe de similarité* $G = (V, E)$. Chaque sommet $v_i \in V$ du graphe représente un vecteur de données x_i . Deux sommets sont connectés si la similarité s_{ij} entre les points de données x_i et x_j correspondant est positive ou supérieure à un certain *seuil*. Toutes les arêtes du graphe sont alors pondérées par la valeur de s_{ij} .

Le problème de la classification peut être ainsi reformulé en utilisant un *graphe de similarité* de la manière suivante: on voudrait trouver une partition du graphe tels que les arêtes entre les différentes classes possèdent une faible pondération (ce qui signifie que les

vecteurs dans des différentes classes sont dissemblables les uns des autres) et les arêtes à l'intérieur d'une classe aient un poids élevé (ce qui signifie que les vecteurs dans une même classe sont similaires les uns aux autres) [26]. Pour être en mesure de formaliser cette intuition, il est nécessaire de rappeler quelques notations de base liées aux graphes qui seront indispensables pour l'étude de la classification spectrale.

III.2.1 Notations sur les graphes

Soit $G = (V, E)$ un graphe non orienté d'ensemble de sommets $V = \{v_1, \dots, v_n\}$. Dans la suite on supposera que le graphe G est pondéré, c'est-à-dire que chaque arête entre deux sommets v_i et v_j est associée un poids non-négatif $w_{ij} \geq 0$. La matrice d'adjacence du graphe est la matrice symétrique $W = (w_{ij})$ avec $i, j = 1, \dots, n$. Si $w_{ij} = 0$, cela signifie que les sommets v_i et v_j ne sont pas reliés par une arête. Comme G est non orienté, nous imposons $w_{ij} = w_{ji}$ (symétrie de W) [26].

Le degré d'un sommet $v_i \in V$ est défini comme:

$$d_i = \sum_{j=1}^n w_{ij} \quad (\text{III. 1})$$

Notez que, en fait, cette somme ne fonctionne que sur tous les sommets adjacents à v_i , ainsi pour tous les autres sommets v_j , le poids w_{ij} est égale à 0. La matrice degré D est définie comme étant la matrice *diagonale* avec les degrés d_1, d_2, \dots, d_n sur la diagonale.

Étant donné un sous-ensemble de sommets $A \subset V$, on note son complément $V \setminus A$ par \bar{A} .

On définit le vecteur indicateur $\vec{1}_A = (f_1, \dots, f_n)^T \in \mathbb{R}^n$ avec l'entrée $f_i=1$ si $v_i \in A$ et $f_i = 0$ autrement. Pour plus de commodité, on introduira la notation abrégée $i \in A$ pour l'ensemble des indices $\{i \mid v_i \in A\}$. On considérait deux manières différentes de mesurer la "taille" d'un sous-ensemble $A \subset V$:

$$\left\{ \begin{array}{l} |A| = \text{nombre de sommets de } A \\ \text{vol}(A) = \sum_{i \in A} d_i \end{array} \right. \quad (\text{III. 2})$$

Intuitivement, $|A|$ mesure la taille de A par le nombre de ses sommets, tandis que $vol(A)$ mesure la taille de A en additionnant les poids de toutes les arêtes attachées aux sommets de A .

Un sous-ensemble $A \subset V$ d'un graphe est *connexe* si deux sommets quelconques de A peuvent être rejoint par un chemin de telle sorte que tous les sommets intermédiaires se trouvent également dans A .

Un sous-ensemble A est appelé *composante connexe* s'il est connexe et s'il n'y a pas de connexions entre les sommets de A et \bar{A} .

Les ensembles non vides A_1, A_2, \dots, A_k forment une partition du graphe si $A_i \cap A_j = \emptyset$ et $A_1 \cup A_2 \cup \dots \cup A_k = V$.

III.2.2 Notion de fonction de similarité

Dans la littérature, plusieurs techniques permettent d'obtenir une mesure de similarité entre deux objets. Le choix de la fonction de similarité dépend essentiellement du domaine de provenance des données (par exemple, fouille de documents, fouille de données web, etc.), mais également du type de données (qui peuvent être décrites par des variables numériques, catégorielles, binaires, ...etc). Cependant, les deux formules les plus répandues et les plus utilisées sont [27]:

- **Noyau Gaussien** [28] :

$$s(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{2\sigma^2}\right) \quad (\text{III. 3})$$

Où $\| \cdot \|$ étant une mesure de distance (de type euclidienne, ...etc.) (*annexe D*), et σ un paramètre d'échelle dont la valeur est fixée par l'utilisateur.

- **Distance cosinus** [29] :

$$s = X^T X \quad (\text{III. 4})$$

Avec X désignant la matrice de données (constituée de n objets décrits chacun par N attributs)

III.3 Différents types de graphes de similarité (graphes de pondération)

Il existe plusieurs constructions pour transformer un ensemble des vecteurs de données x_1, x_2, \dots, x_n , avec une mesure de similarité s_{ij} ou de distance d_{ij} entre deux vecteurs de données d'un graphe. Le but lors de la construction des graphes de similarité est de modéliser les relations locales de voisinage entre les vecteurs de données. En outre, la plupart des constructions ci-dessous conduisent à une représentation clairsemée des données, qui représentent des avantages de calcul [26].

III.3.1 Graphe de voisinage - ϵ (The ϵ - neighborhood graph)

Il s'agit de relier deux à deux tous les nœuds dont les distances sont plus petites que ϵ . Comme les distances entre les nœuds connectés sont à peu près de la même échelle (au plus ϵ), la pondération des arêtes ne contiendra pas plus d'informations sur les données du graphe. Ainsi, le graphe de voisinage - ϵ est généralement considéré comme un graphe non pondéré. La figure ci-dessous montre un exemple d'un graphe de voisinage ϵ .

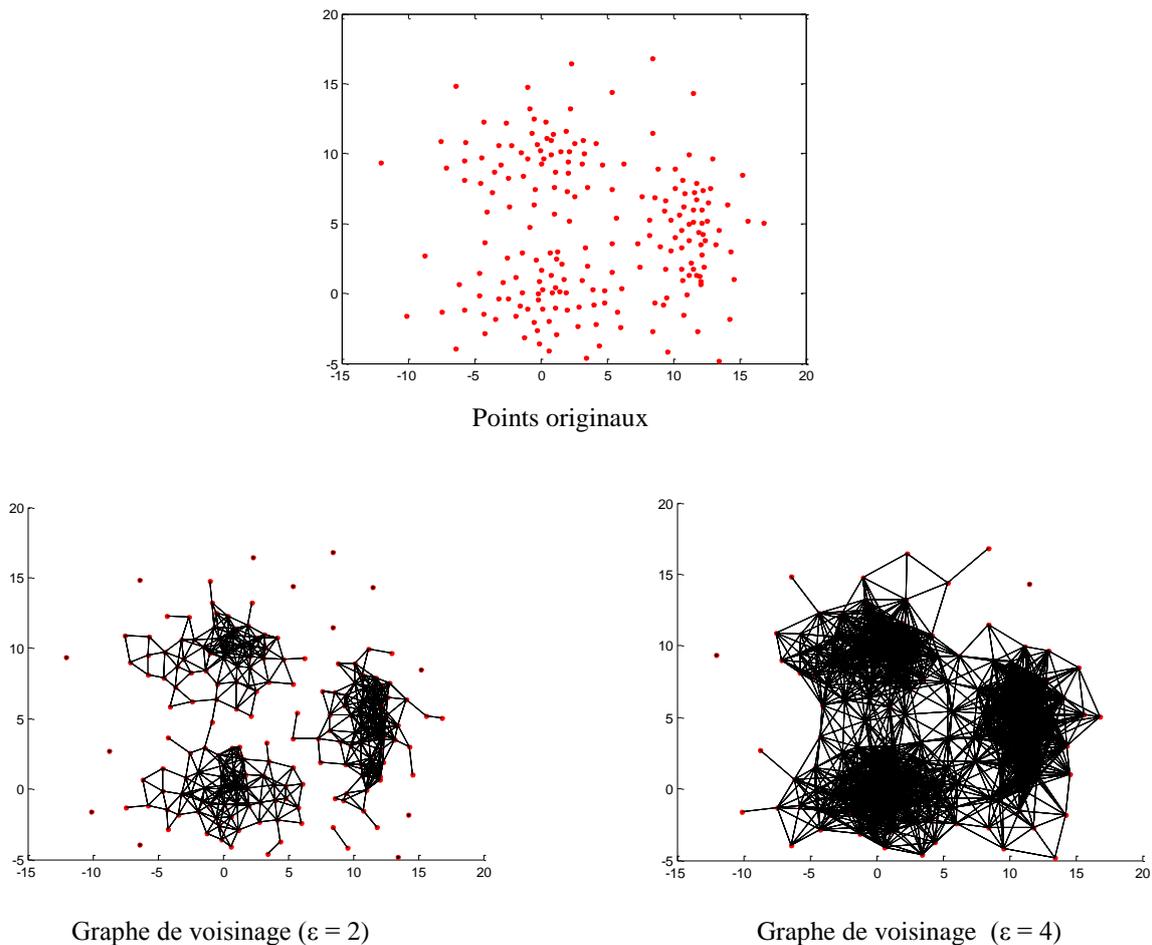


Figure III.1 : Graphe de voisinage ϵ

III.3.2 Graphe des K plus proches voisins (K -nearest neighbor graph)

Dans ce type de graphe, le sommet v_i est lié au sommet v_j si v_j est parmi les K plus proches voisins (K -PPV) de v_i . Cependant, cette définition conduit à un graphe orienté, et la relation de voisinage n'est pas symétrique.

À présent, il existe deux façons de rendre ce graphe un non orienté :

- La première façon consiste tout simplement à ignorer le sens des arêtes, en reliant le sommet v_i et le sommet v_j avec une arête non orientée « si v_i est parmi les K plus proches voisins de v_j ou si v_j est parmi les K plus proches voisins de v_i ». Le graphe qui résulte est généralement appelé « graphe des K -plus proches voisins *standard* ».
- Le second choix consiste à relier le sommet v_i avec le sommet v_j « si v_i est parmi les K plus proches voisins de v_j et si v_j est parmi les K plus proches voisins de v_i ». Le graphe qui résulte est appelé « graphe des K -plus proche voisins *mutuel* ».

Dans les deux cas, les arêtes sont pondérées par la similarité des points adjacents.

La figure ci-dessous montre les deux types de graphe des K plus proches voisins (*standard* et *mutuel*) pour $K=10$ (10 plus proches voisins).

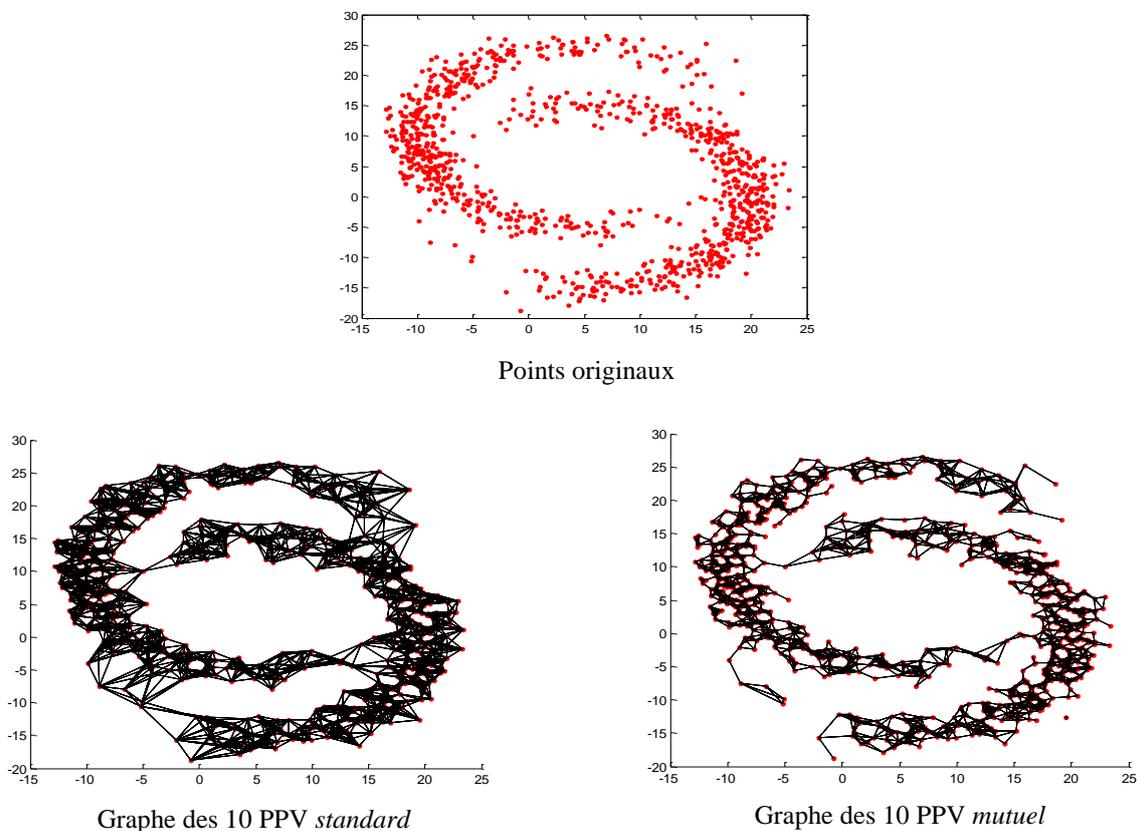


Figure III.2 : Graphes des K plus proches voisins *standard* et *mutuel*

III.3.3 Graphe totalement connecté (Fully connected graph)

Dans ce type de graphe, tous les sommets sont reliés entre eux. La similarité entre deux sommets est positive, et toutes les arêtes sont pondérées par les poids s_{ij} .

Un exemple d'une fonction de similarité est la *fonction de similarité gaussienne* de l'équation (III.3) où le paramètre σ contrôle la largeur de voisinage. La figure (III.3) montre un exemple de graphe totalement connecté.

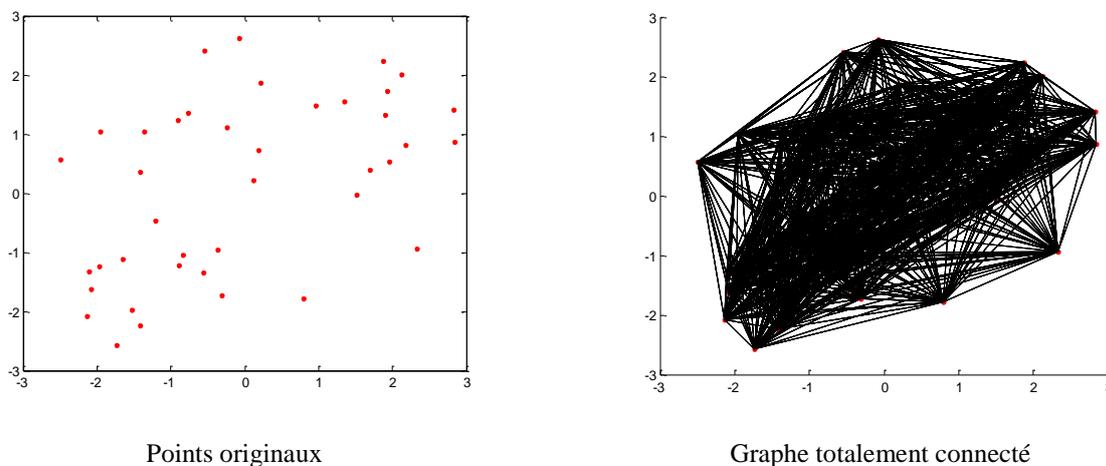


Figure III.3 : Graphe totalement connecté

III.4 Matrices laplaciennes d'un graphe et leurs propriétés de base [26]

La classification spectrale est principalement fondée sur l'étude des matrices laplaciennes d'un graphe. Il existe tout un champ dédié à l'étude de ces matrices, appelé *théorie spectrale des graphes* (voir [30]).

Dans cette partie, nous allons définir les différentes matrices laplaciennes d'un graphe et leurs propriétés les plus importantes. On prendra soin de distinguer entre les différentes variantes des matrices laplaciennes d'un graphe.

Plusieurs matrices laplaciennes d'un graphe sont proposées dans la littérature. Cependant, ces matrices peuvent être normalisées ou non normalisées.

Dans la suite, on supposera que G est un graphe *non orienté*, pondéré avec une matrice de pondération W , où $w_{ij} = w_{ji} \geq 0$.

En classification spectrale, on cherche toujours à déterminer les valeurs et vecteurs propres des matrices laplaciennes. Les valeurs propres seront toujours ordonnées par ordre croissant, tout en respectant les multiplicités. On réfère les k premiers vecteurs propres aux vecteurs propres correspondant aux k plus petites valeurs propres.

III.4.1 Matrice laplacienne *non normalisée* d'un graphe

La matrice laplacienne *non normalisée* d'un graphe est définie comme :

$$L = D - W \quad (\text{III.5})$$

Où D est la matrice degré et W la matrice de pondération.

Des détails sur les propriétés de cette matrice peuvent être trouvés dans [37] et [38].

La matrice L vérifie les propriétés suivantes:

1. Pour chaque vecteur $f \in \mathbb{R}^n$ nous avons :

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \quad (\text{III.6})$$

2. L est symétrique et semi-définie positive.
3. La plus petite valeur propre de L est égal à 0, le vecteur propre correspondant est le vecteur constant $\vec{1}$.
4. L possède n valeurs propres réelles non-négatives, $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.

Les valeurs propres et vecteurs propres de la matrice laplacienne *non normalisée* d'un graphe peuvent être utilisés pour décrire de nombreuses propriétés des graphes (voir [31] et [32]).

Le nombre de composantes connexes et le spectre de L :

Soit G un graphe non orienté avec des poids non-négatifs.

La multiplicité k de la valeur propre 0 de L est égale au nombre de composantes connexes A_1, \dots, A_k dans le graphe. L'espace propre de la valeur propre 0 est coupé par les vecteurs indicateurs $\vec{1}_{A_1}, \dots, \vec{1}_{A_k}$ de ces composantes.

Preuve :

On commence par le cas $k = 1$ (graphe connexe).

Supposons que f est un vecteur propre avec valeur propre 0, sachant que :

$$0 = f^T L f = \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$$

Comme les poids w_{ij} sont pas négatifs, cette somme ne peut s'annuler que si tous les termes $w_{ij}(f_i - f_j)^2$ s'annulent. Ainsi, si deux sommets v_i et v_j sont connectés ($w_{ij} > 0$), alors f_i doit être égale à f_j . En outre, comme tous les sommets d'une composante connexe dans un graphe non orienté peuvent être connectés par un chemin, f doit être constant sur toute la composante connexe.

Considérons maintenant le cas de k composantes connexes. On suppose que les sommets sont classés selon les composantes connexes aux quelles ils appartiennent.

Dans ce cas, la matrice d'adjacence W à une forme de matrice bloc-diagonale (ou diagonale par blocs), et la même chose est vraie pour la matrice L (remplie des blocs L_i sur la diagonale, et des 0 dans les autres positions) :

$$\begin{pmatrix} L_1 & \dots & 0 \\ \cdot & L_2 & \cdot \\ 0 & \dots & L_k \end{pmatrix}$$

□ Notons que chacun des blocs L_i est lui même une matrice laplacienne correspond au sous-graphe de la i -ème composante connexe. Comme c'est le cas pour toutes les matrices bloc-diagonale, le spectre de L est donné par l'union des spectres des L_i et les vecteurs propres de L sont les vecteurs propres des L_i .

□ **Remarque :** Une autre matrice laplacienne *non normalisée* existe dans la littérature donnée par [33] : $L = W$

III.4.2 Matrice laplacienne *normalisée* d'un graphe

On trouve dans la littérature deux types de « matrices laplaciennes *normalisées* » d'un graphe. Ces deux matrices sont étroitement liées l'une à l'autre et sont définies comme :

$$L_{sym} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2} \quad (\text{III.7})$$

$$L_{rw} = D^{-1}L = I - D^{-1}W \quad (\text{III.8})$$

I est la matrice identité. On note la première matrice par L_{sym} car c'est une matrice symétrique et la seconde par L_{rw} car elle est liée étroitement à une marche aléatoire (random walk).

Dans ce qui suit, nous résumerons plusieurs propriétés de L_{sym} et L_{rw} . Pour plus de détails sur les matrices laplaciennes *normalisées* d'un graphe, se référer à [30].

Les matrices laplaciennes *normalisées* L_{sym} et L_{rw} vérifient les propriétés suivantes:

1. Pour chaque vecteur $f \in \mathbb{R}^n$ nous avons :

$$f^T L_{sym} f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 \quad (\text{III.9})$$

2. λ est une valeur propre de L_{rw} de vecteur propre v si et seulement si λ est une valeur propre de L_{sym} de vecteur propre $w = D^{1/2}v$.

3. λ est une valeur propre de L_{rw} de vecteur propre v si et seulement si v et λ résolvent le problème propre généralisé $Lv = \lambda Dv$.

4. 0 est une valeur propre de L_{rw} de vecteur propre constant $\vec{1}$. 0 est une valeur propre de L_{sym} de vecteur propre $D^{1/2}\vec{1}$.

5. L_{sym} et L_{rw} sont semi-définies positives et possèdent n valeurs propres réelles non négatives $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.

Comme dans le cas de la matrice laplacienne *non normalisée*, la multiplicité de la valeur propre 0 de la matrice laplacienne *normalisée* est liée au nombre de composantes connexes.

Nombre de composantes connexes et spectres de L_{sym} et L_{rw} :

Soit G un graphe non orienté pondéré avec des poids non-négatifs.

La multiplicité k de la valeur propre 0 des deux matrices L_{rw} et L_{sym} est égale au nombre de composantes connexes A_1, \dots, A_k dans le graphe. Pour L_{rw} , l'espace propre de 0 est engendré

par les vecteurs indicateurs $\vec{1}_{A_i}$ de ces composantes. Pour L_{sym} , l'espace propre de 0 est engendré par les vecteurs $D^{1/2}\vec{1}_{A_i}$. La preuve est analogue à celle du cas *non normalisé*.

□ **Remarque :** Il existe d'autres matrices normalisées présentées dans la littérature [27]:

- Normalisation par division [40] : $L = D^{-1}W$
- Normalisation par division symétrique [36] : $L = D^{-1}WD^{-1}$
- Normalisation additive [41] : $L = \frac{(W+d_{max}I-D)}{d_{max}}$

Avec d_{max} désignant le degré maximum de D et I la matrice identité.

III.5 Algorithmes de classification spectrale [26]

Dans cette partie, nous allons décrire les algorithmes les plus couramment utilisés en classification spectrale. On suppose que nos données sont composées de n points x_1, \dots, x_n qui peuvent être des objets arbitraires et leurs similarité $s_{ij} = s(x_i, x_j)$, mesurée selon une *fonction de similarité* qui est symétrique et non négative. On note la matrice de similarité correspondante par :

$$S = (s_{ij})_{i,j=1\dots n}.$$

Algorithme de classification spectrale *non normalisée*

Entrée: Ensemble de données x_1, \dots, x_n , nombre k de classes que nous voulons obtenir.

- Calculer la matrice de similarité $s \in \mathbb{R}^{n \times n}$.
- Construire le graphe de similarité par l'une des méthodes décrites dans la partie (III.3).
- Calculer sa matrice d'adjacence pondérée $W \in \mathbb{R}^{n \times n}$.
- Calculer la matrice laplacienne *non normalisée* L .
- **Calculer les k premiers vecteurs propres v_1, \dots, v_k de L**
- Soit $V \in \mathbb{R}^{n \times k}$ la matrice contenant les vecteurs v_1, \dots, v_k sous forme de colonnes.
- Pour $i = 1, \dots, n$, créer le nouvel ensemble de points $y_i \in \mathbb{R}^k$ correspondant à la i -ème ligne de V .
- Regrouper en classes C_1, \dots, C_k les points $(y_i)_{i=1, \dots, n}$ dans \mathbb{R}^k avec l'algorithme k -Means.

Sortie: Classes (clusters) A_1, \dots, A_k , tel que x_j est affecté à A_i si y_j est affecté à C_i .

Il existe deux versions différentes de classification spectrale *normalisée* selon la matrice laplacienne utilisée.

Algorithme de classification spectrale *normalisée* selon Shi et Malik (2000)[35]

Entrée: Ensemble de données x_1, \dots, x_n , nombre k de classes que nous voulons obtenir .

- Calculer la matrice de similarité $s \in \mathbb{R}^{n \times n}$.
- Construire le graphe de similarité par l'une des méthodes décrites dans la partie (III.3).
- Calculer sa matrice d'adjacence pondérée $W \in \mathbb{R}^{n \times n}$.
- Calculer la matrice laplacienne *non normalisée* L .
- **Calculer les k premiers vecteurs propres v_1, \dots, v_k du problème propre généralisé $Lv = \lambda Dv$**
- Soit $V \in \mathbb{R}^{n \times k}$ la matrice contenant les vecteurs v_1, \dots, v_k sous forme de colonnes.
- Pour $i = 1, \dots, n$, créer le nouvel ensemble de points $y_i \in \mathbb{R}^k$ correspondant à la i -ème ligne de V .
- Regrouper en classes C_1, \dots, C_k les points $(y_i)_{i=1, \dots, n}$ dans \mathbb{R}^k avec l'algorithme k -Means.

Sortie: Classes (clusters) A_1, \dots, A_k , tel que x_j est affecté à A_i si y_j est affecté à C_i .

□ Notez que cet algorithme utilise les vecteurs propres généralisés de L , qui selon les propriétés de L_{rw} et L_{sym} correspondents aux vecteurs propres de la matrice L_{rw} . Donc, en fait, l'algorithme fonctionne avec les vecteurs propres de la matrice laplacienne *normalisée* L_{rw} , d'où son nom classification spectrale *normalisée*.

L'algorithme suivant utilise également une laplacienne *normalisée*, mais cette fois c'est la matrice L_{sym} qui est utilisée au lieu de L_{rw} .

**Algorithme de classification spectrale *normalisée* selon Ng, Jordan et Weiss
(2002) [36]**

Entrée: Ensemble de données x_1, \dots, x_n , nombre k de classes que nous voulons obtenir .

- Calculer la matrice de similarité $s \in \mathbb{R}^{n \times n}$.
- Construire le graphe de similarité par l'une des méthodes décrites dans la partie (III.3).
- Calculer sa matrice d'adjacence pondérée $W \in \mathbb{R}^{n \times n}$.
- Calculer la matrice laplacienne normalisée L_{sym} .
- **Calculer les k premiers vecteurs propres v_1, \dots, v_k de L_{sym}**
- Soit $V \in \mathbb{R}^{n \times k}$ la matrice contenant les vecteurs v_1, \dots, v_k sous forme de colonnes.
- **Formuler la matrice $T \in \mathbb{R}^{n \times k}$ en normalisant les lignes de V ,**

qui est définie :

$$t_{ij} = \frac{v_{ij}}{\left(\sum_{j=1}^k v_{ij}^2\right)^{1/2}} \quad (\text{III. 10})$$

- Pour $i = 1, \dots, n$, créer le nouvel ensemble de points $y_i \in \mathbb{R}^k$ correspondant à la i -ème ligne de T .
- Regrouper en classes C_1, \dots, C_k les points $(y_i)_{i=1, \dots, n}$ dans \mathbb{R}^k avec l'algorithme k -Means

Sortie: Classes (clusters) A_1, \dots, A_k , tel que x_j est affecté à A_i si y_j est affecté à C_i .

Tous les trois algorithmes indiqués ci-dessus semblent plutôt similaires, indépendamment du fait qu'ils utilisent trois types de matrices laplaciennes.

Dans les trois algorithmes, l'idée principale consiste à modifier la représentation des points de données x_i aux points $y_i \in \mathbb{R}^k$. Cela est dû aux propriétés de la matrice laplacienne du graphe. Ce changement de représentation va renforcer les propriétés de la classe dans les données, de sorte que les classes peuvent être aisément détectées dans la nouvelle représentation.

Par conséquent, un algorithme de classification simple comme k -Means n'aura pas de difficultés à détecter les classes dans cette nouvelle représentation [26].

III.6 Notions de coupes (Cuts) de graphe

Une façon de justifier la classification spectrale comme une technique de classification de données est d'appliquer une coupe de graphe (figure III.4). L'idée est de construire une partition du graphe de sorte que les arêtes entre les différentes classes ont des poids faibles, à l'intérieur d'une classe, les arêtes ont des poids élevés. Pour étudier les partitions possibles dans le graphe, on introduit une nouvelle mesure : la coupe de graphe.

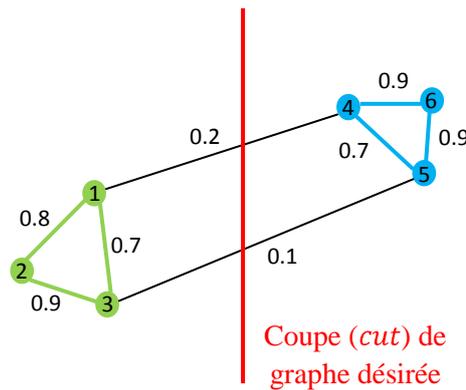


Figure III.4 :
Exemple de graphe pondéré des données

Soit deux sous ensembles disjoints $A, B \subset V$, on définit:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{III. 11}$$

Étant donné un graphe de similarité et une matrice de pondération W . La façon la plus simple et la plus directe de construire une partition du graphe est de résoudre un problème dit *mincut*. Pour un nombre k donné de sous-ensembles, le *mincut* consiste simplement à choisir une partition A_1, \dots, A_k qui minimise :

$$cut(A_1, \dots, A_k) = \sum_{i=1}^k cut(A_i, \bar{A}_i) \tag{III. 12}$$

Ou \bar{A} est le complément du sous ensemble $A \subset V$

En particulier pour $k = 2$, le *mincut* est un problème relativement facile et peut être résolu efficacement [34]. Cependant, dans la pratique les résultats sont très souvent décevants, car on a tendance à séparer simplement un sommet du reste du graphe.

Afin de remédier à ce problème, deux fonctions de coupe ont été proposées, la *RatioCut* [37] et la coupe normalisée *NCut* [35]. En *RatioCut*, la taille d'un sous-ensemble A d'un graphe est mesurée par le nombre de ses sommets $|A|$, tandis que dans *NCut* la taille du sous-ensemble A est mesurée par le poids de ses arêtes $vol(A)$. Les définitions sont les suivantes:

$$RatioCut = (A_1, \dots, A_k) = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{|A_i|} \quad (\text{III. 13})$$

$$NCut = (A_1, \dots, A_k) = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{vol(A_i)} \quad (\text{III. 14})$$

Sachons que l'objectif de ces deux fonctions est de prendre une petite valeur si la taille des sous ensembles A_i n'est pas trop petite. En fait, le minimum de la fonction $\sum_{i=1}^k (1/|A_i|)$ est atteint si tous les $|A_i|$ coïncident (identiques), et le minimum de $\sum_{i=1}^k (1/vol(A_i))$ est atteint si tous les $vol(A_i)$ coïncident (identiques). Ces deux fonctions essaient alors de produire des classes équilibrées en fonction de leurs degrés. Malheureusement, l'introduction de ces conditions de balance (d'équilibrage) fait que le problème devient NP-difficile [38].

La classification spectrale est un moyen de résoudre les versions relaxées de ce problème. Nous allons voir que la relaxation du *NCut* conduit à la classification spectrale *normalisée*, tandis que la relaxation de *RatioCut* conduit à la classification spectrale *non normalisée*.

Approximation de *RatioCut* pour $k = 2$:

On commence par le cas de *RatioCut* et $k = 2$, parce que la relaxation est plus facile à comprendre dans ce cadre. L'objectif est de résoudre le problème d'optimisation suivant :

$$\min_{A \subset V} RatioCut(A, \bar{A}) \quad (\text{III. 15})$$

Etant donné un sous-ensemble $A \subset V$, on définit le vecteur indicateur $f = (f_1, \dots, f_n)^T \in \mathbb{R}^n$ avec des entrées :

$$f_i = \begin{cases} \sqrt{\frac{|\bar{A}|}{|A|}} & \text{si } v_i \in A \\ -\sqrt{\frac{|A|}{|\bar{A}|}} & \text{si } v_i \in \bar{A} \end{cases} \quad (\text{III.16})$$

La fonction objectif *RatioCut* peut être facilement réécrite en utilisant la matrice laplacienne *non normalisée* :

$$\begin{aligned} f^T L f &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \\ &= \frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} \left(\sqrt{\frac{|\bar{A}|}{|A|}} + \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 + \frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} \left(-\sqrt{\frac{|\bar{A}|}{|A|}} - \sqrt{\frac{|A|}{|\bar{A}|}} \right)^2 \\ &= \text{cut}(A, \bar{A}) \left(\frac{|\bar{A}|}{|A|} + \frac{|A|}{|\bar{A}|} + 2 \right) \\ &= \text{cut}(A, \bar{A}) \left(\frac{|\bar{A}| + |A|}{|A|} + \frac{|A| + |\bar{A}|}{|\bar{A}|} \right) \\ &= (|\bar{A}| + |A|) \text{cut}(A, \bar{A}) \left(\frac{1}{|A|} + \frac{1}{|\bar{A}|} \right) \\ f^T L f &= |V| \times \text{RatioCut}(A, \bar{A}) \end{aligned} \quad (\text{III.17})$$

$|V|$: est le nombre de sommets de V

En outre, on a :

$$f^T \vec{1} = \sum_{i=1}^n f_i = \sum_{i \in A} \sqrt{\frac{|\bar{A}|}{|A|}} - \sum_{i \in \bar{A}} \sqrt{\frac{|A|}{|\bar{A}|}} = |A| \sqrt{\frac{|\bar{A}|}{|A|}} - |\bar{A}| \sqrt{\frac{|A|}{|\bar{A}|}} = 0$$

En d'autres termes, le vecteur f qui est défini dans l'équation (III.16) est orthogonale au vecteur constant $\vec{1}$.

Notons aussi que f satisfait :

$$\|f\|^2 = \sum_{i=1}^n f_i^2 = |A| \frac{|\bar{A}|}{|A|} + |\bar{A}| \frac{|A|}{|\bar{A}|} = |\bar{A}| + |A| = n$$

Finalement, le problème de minimisation (III. 15) peut se réécrire sous la forme suivante :

$$\min_{A \subset V} f^T L f$$

Sujet à : (III. 18)

$$\begin{aligned} f &\perp \vec{1} \\ \|f\| &= \sqrt{n} \end{aligned}$$

Il s'agit d'un problème d'optimisation discret dont les éléments du vecteur solution f ne sont autorisés à prendre que deux valeurs particulières, il est encore NP-difficile.

La relaxation la plus évidente dans ce contexte est de se défaire de l'état discret et de permettre à f_i de prendre des valeurs arbitraires dans \mathbb{R} . Il en résulte le problème d'optimisation relaxé suivant:

$$\min_{f \in \mathbb{R}^n} f^T L f$$

Sujet à : (III. 19)

$$\begin{aligned} f &\perp \vec{1} \\ \|f\| &= \sqrt{n} \end{aligned}$$

Selon le théorème de Rayleigh-Ritz (*annexe A*) [39] qui dit que la solution de ce problème est donnée par le vecteur f qui est le vecteur propre correspondant à la deuxième plus petite valeur propre de L (rappelons que la plus petite valeur propre de L est 0, le vecteur propre correspondant est $\vec{1}$). Donc, on peut approcher le minimum de *RatioCut* par le second vecteur propre de L . Toutefois, dans le but d'obtenir une partition du graphe nous avons besoin de retransformer les valeurs réelles du vecteur solution f du problème relaxé en valeurs discrètes correspondant au vecteur indicateur des classes.

La façon la plus simple pour le faire est d'utiliser le signe de f , on peut écrire :

$$\begin{cases} v_i \in A & \text{si } f_i \geq 0 \\ v_i \in \bar{A} & \text{si } f_i < 0 \end{cases} \quad (\text{III. 20})$$

Toutefois, dans le cas où $k > 2$, cette heuristique est très simple. Ce que font la plupart des algorithmes de classification spectrale est de considérer les coordonnées de f_i comme des points dans \mathbb{R} et de les regrouper en deux classes C et \bar{C} par l'algorithme de k -Means.

Puis on réalise par la suite le regroupement des vecteurs de données:

$$\begin{cases} v_i \in A & \text{si } f_i \in C \\ v_i \in \bar{A} & \text{si } f_i \in \bar{C} \end{cases} \quad (\text{III. 21})$$

C'est exactement l'algorithme de classification spectrale *non normalisée* pour $k = 2$.

Approximation de *RatioCut* pour k arbitraire :

Compte tenu d'une partition de V en k ensembles A_1, \dots, A_k , on définit les k vecteurs indicateurs $h_i = (h_{1,i}, \dots, h_{n,i})^T$ par :

$$h_{i,j} = \begin{cases} 1/\sqrt{|A_j|} & \text{si } v_i \in A_j \\ 0 & \text{sinon} \end{cases} \quad (i = 1, \dots, n; j = 1, \dots, k) \quad (\text{III. 22})$$

Puis on définit $H \in \mathbb{R}^{n \times k}$ comme la matrice contenant les k vecteurs indicateurs, sous forme de colonnes. Sachons que les colonnes de H sont orthonormales entre elles, (c.-à-d $H^T H = I$)

Par les mêmes calculs pour l'équation (III. 17), on aura :

$$h_i^T L h_i = \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|} \quad (\text{III. 23})$$

En outre, on peut vérifier que :

$$h_i^T L h_i = (H^T L H)_{ii} \quad (\text{III. 24})$$

Par combinaison de ces deux équations, on obtient :

$$\text{RatioCut}(A_1, \dots, A_k) = \sum_{i=1}^k h_i^T L h_i = \sum_{i=1}^k (H^T L H)_{ii} = \text{Tr}(H^T L H) \quad (\text{III. 25})$$

Où Tr désigne la trace d'une matrice.

Donc, le problème de minimisation de $\text{RatioCut}(A_1, \dots, A_k)$ peut se réécrire sous la forme suivante:

$$\begin{aligned} & \min_{A_1, \dots, A_k} \text{Tr}(H^T L H) \\ & \text{Sujet à :} \end{aligned} \quad (\text{III. 26})$$

$$H^T H = I$$

De la même manière que précédemment, on relaxe le problème afin de permettre aux éléments de la matrice H de prendre des valeurs réelles arbitraires, le problème devient :

$$\begin{aligned} & \min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(H^T L H) \\ & \text{Sujet à :} \end{aligned} \quad (\text{III. 27})$$

$$H^T H = I$$

C'est la forme standard d'un problème de minimisation de la trace, qui est encore une version du théorème de Rayleigh-Ritz (*annexe A*) [39] qui dit que la solution est donnée par la matrice H^* qui contient les k premiers vecteurs propres de L sous forme de colonnes.

On peut voir que la matrice H est équivalente à la matrice V utilisée dans l'algorithme de classification spectrale *non normalisée*. Toutefois, on a besoin de retransformer les valeurs solutions (réelles) de la matrice vers une partition discrète. Ensuite, nous pouvons appliquer l'algorithme de k -Means sur les lignes de H .

Approximation de $NCut$:

Les techniques employées précédemment pour $RatioCut$ peuvent être utilisées afin d'obtenir une classification spectrale *normalisée* par minimisation de $NCut$. Dans le cas où $k = 2$, on définit le vecteur indicateur des classes f par :

$$f_i = \begin{cases} \sqrt{\frac{vol(\bar{A}_i)}{vol(A_i)}} & \text{si } v_i \in A \\ -\sqrt{\frac{vol(A_i)}{vol(\bar{A}_i)}} & \text{si } v_i \in \bar{A} \end{cases} \quad (\text{III. 28})$$

Ensuite, par des calculs similaires à ceux utilisés pour $RatioCut$, on peut vérifier que :

$$\begin{cases} (Df)^T \vec{1} = 0 \\ f^T Df = vol(V) \\ f^T Lf = vol(V) \times NCut(A, \bar{A}) \end{cases}$$

Ainsi, le problème de minimisation de $NCut$ peut se réécrire sous la forme suivante :

$$\begin{aligned} & \min_{A \subset V} f^T Lf \\ & \text{Sujet à :} \end{aligned} \quad (\text{III. 29})$$

$$\begin{aligned} & Df \perp \vec{1} \\ & f^T Df = vol(V) \end{aligned}$$

De la même manière que précédemment, on relaxe le problème afin de permettre à f de prendre des valeurs réelles, le problème devient alors:

$$\begin{aligned} & \min_{f \in \mathbb{R}^n} f^T Lf \\ & \text{Sujet à :} \end{aligned} \quad (\text{III. 30})$$

$$\begin{aligned} & Df \perp \vec{1} \\ & f^T Df = vol(V) \end{aligned}$$

On pose $g = D^{1/2}f$, le problème devient :

$$\min_{g \in \mathbb{R}^n} g^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} g$$

Sujet à : (III. 31)

$$\begin{aligned} g &\perp D^{1/2} \vec{1} \\ \|g\|^2 &= vol(V) \end{aligned}$$

On remarque que $D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = L_{sym}$, $D^{1/2} \vec{1}$ est le premier vecteur propre de L_{sym} , $vol(V)$ une est constante. Comme dans les raisonnements précédents, la solution g du problème (III.31) est donnée par le théorème de Rayleigh-Ritz (*annexe A*) qui est le second vecteur propre de L_{sym} . En remplaçant $f = D^{-1/2}g$ et en utilisant les propriétés de L_{rw} et L_{sym} , on peut voir que f est alors le deuxième vecteur propre de L_{rw} , ou de manière équivalente c'est le vecteur propre généralisé de $Lv = \lambda Dv$.

Dans le cas ou $k > 2$, on définit les vecteurs indicateurs $h_i = (h_{1,i}, \dots, h_{n,i})^T$ par :

$$h_{i,j} = \begin{cases} 1/\sqrt{vol(A_j)} & \text{Si } v_i \in A_j \\ 0 & \text{Sinon} \end{cases} \quad (i = 1, \dots, n; j = 1, \dots, k) \quad \text{(III. 32)}$$

Puis, on définit H comme étant la matrice contenant les k vecteurs indicateurs sous forme de colonnes. Sachons que :

$$\begin{cases} H^T D H = I \\ h_i^T D h_i = 1 \\ h_i^T L h_i = Cut(A_i, \bar{A}_i)/vol(A_i) \end{cases}$$

Le problème de minimisation $NCut$ peut s'écrire comme suit:

$$\min_{A_1, \dots, A_k} Tr(H^T L H)$$

Sujet à : (III. 33)

$$H^T D H = I$$

Si on pose $T = D^{1/2}H$, on obtient le problème relaxé suivant :

$$\min_{T \in \mathbb{R}^{n \times k}} \text{Tr}(T^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} T)$$

Sujet à : (III.34)

$$T^T T = I$$

Là encore, c'est le problème de minimisation de la trace qui est résolu, par le théorème de Rayleigh-Ritz (*annexe A*), la solution est donnée par la matrice T^* qui contient les k premiers vecteurs propres de L_{sym} sous forme de colonnes. En remplaçant $H = D^{-1/2}T$ et en utilisant les propriétés de L_{rw} et L_{sym} , la solution H comporte les k premiers vecteurs propres de la matrice L_{rw} , ou bien les k premiers vecteurs propres généralisés de $Lv = \lambda Dv$. On aboutit alors à l'algorithme de classification spectrale *normalisée* selon Shi et Malik (2000).

III.7 Choix des paramètres pour la classification spectrale

Trouver de bons paramètres pour la classification spectrale n'est pas une tâche facile, car il y a plusieurs choix à faire et paramètres à définir. Donc, comment ces paramètres influent sur les résultats de la classification spectrale ? C'est ce que nous verrons dans cette partie.

III.7.1 Paramètre de la fonction de similarité σ

Avant même de penser à construire un graphe de similarité, on doit définir une fonction de similarité sur les données.

L'expression de la similarité *gaussienne* dépend du paramètre σ . Or, le principe de la classification spectrale repose sur la mesure de similarité, ce paramètre influe directement sur la méthode. En effet, σ influe sur la séparabilité des données dans l'espace de représentation.

En segmentation d'images, ce paramètre est souvent laissé libre et doit être fixé par l'utilisateur. Cependant, plusieurs interprétations sur le rôle de ce paramètre ont été développées afin de guider le choix de manière automatique ou semi-automatique. Les principales approches sont les suivantes :

III.7.1.1 Approche seuillage

Une première approche repose sur l'idée de seuillage. En effet, Perona et Freeman [42] définissent σ comme une distance seuil en dessous de laquelle deux points sont considérés comme similaires et au-dessus de laquelle ils sont dissemblables.

Von Luxburg [26] interprète σ comme un rayon du voisinage. Il joue donc un rôle équivalent à celui du ε , distance de voisinage pour la méthode de graphe de voisinage ε . Ces interprétations donnent des informations sur le rôle que joue le paramètre σ . Cependant, le choix du seuil reste ouvert et non automatisable.

III.7.1.2 Approche globale

Une seconde approche repose sur des définitions globales de ce paramètre, principalement basées sur l'analyse descriptive des données et des interprétations physiques. D'après Ng, Jordan, Weiss [36], σ contrôle la similarité entre les données et conditionne la qualité des résultats. Ainsi, parmi les différentes valeurs de σ , on sélectionne celle qui minimise la dispersion des points d'une même classe dans l'espace propre.

D'après Brand et Huang [43], σ doit être égal à la moyenne de la distance entre chaque point de l'ensemble et son plus proche voisin.

III.7.1.3 Approche locale

Une dernière classe de définitions basée sur des interprétations physiques privilégie des approches locales où un paramètre scalaire spécifique est défini pour chaque point x_i . Perona et Zelnik-Manor [44] ont adopté une approche locale, consiste à définir un scalaire pour chaque couple de point x_i, x_j . Ils assignent un paramètre scalaire σ_i différent à chaque point x_i de l'ensemble de données. σ_i est égal à la distance entre le point x_i et son k ième voisin le plus proche. Cette méthode donne de bons résultats, mais calculer une valeur de σ pour chaque point x_i peut être coûteux en termes de temps de calcul et la valeur k reste fixée empiriquement.

Une autre approche locale développée par Fischer et Poland [45] utilise la conductivité définie dans les réseaux électriques : la conductivité entre deux points dépend alors de tous les chemins entre eux. Le paramètre local est fixé tel que la somme des lignes de la matrice de similarité est égale à une valeur ζ . Donc σ dépend d'une autre valeur ζ représentant un rayon de voisinage à fixer empiriquement.

Ces définitions locales s'avèrent très efficaces pour des cas de données bruitées. Elles permettent de distinguer le bruit des données. Cependant, ces approches s'avèrent coûteuses et impliquent de définir de nouveaux paramètres, respectivement le nombre de voisins k ou la valeur ζ .

III.7.2 Choix du type de graphe de similarité

L'autre choix à faire concerne le type de graphe qu'on veut utiliser (graphe totalement connecté, graphe des K plus proches voisins ou le graphe de voisinage ε). Pour illustrer le comportement des différents graphes, nous utilisons l'exemple présenté dans la figure (III.5). Comme distribution, nous avons choisi une sur \mathbb{R}^2 avec trois classes: deux *croissants* et une *gaussienne* (l'une des cornes des croissants est choisie pour être plus dense que l'autre corne). La figure (III.5.a) montre un échantillon tiré de cette distribution. Les trois figures (b), (c) et (d) montrent les différents graphes de similarité appliqués sur cet échantillon.

Dans le graphe de voisinage ε , nous pouvons voir qu'il est difficile de choisir un paramètre ε convenable. Avec $\varepsilon = 1.8$ (figure (III.5.b)), les points des deux croissants sont déjà étroitement liés, tandis que les points de la *gaussienne* sont à peine connectés. Ce problème se produit toujours si on dispose de données qui sont sur des échelles différentes c'est-à-dire que les distances entre les points de données sont différentes dans les différentes régions de l'espace.

Le graphe des K plus proches voisins (figure (III.5.c) et (III.5.d)), d'autre part, peut relier des points sur des échelles différentes. Nous pouvons voir que les points de faible densité de la *gaussienne* sont connectés avec les points à haute densité, c'est une propriété générale des graphes des K plus proches voisins qui peut être très utile. On peut voir également que le graphe des K plus proches voisins peut se diviser en plusieurs composantes déconnectées. Le graphe des K plus proches voisins *mutuel* possède la propriété de relier des régions de densité constante, mais il ne permet pas de relier des régions de densité différentes. Donc, le graphe des K plus proches voisins *mutuel* peut être considéré comme étant entre le graphe de voisinage ε et le graphe des K plus proches voisins *standard*. Il est capable d'agir à des échelles différentes. Ainsi, le graphe des K plus proches voisins *mutuel* semble particulièrement bien adapté si l'on veut détecter des classes de densités différentes. Généralement, on peut observer que le graphe des K plus proches voisins *mutuel* à un nombre d'arêtes beaucoup moins faible que le graphe des K plus proches voisins *standard* pour le même paramètre K .

Pour le graphe totalement connecté, le paramètre σ joue le même rôle que le paramètre ε dans le graphe de voisinage ε . Les points dans un même voisinage sont reliés avec des arêtes qui ont des poids relativement élevés. Cependant, les arêtes des points les plus éloignés peuvent avoir des effets, mais leurs poids reste négligeable.

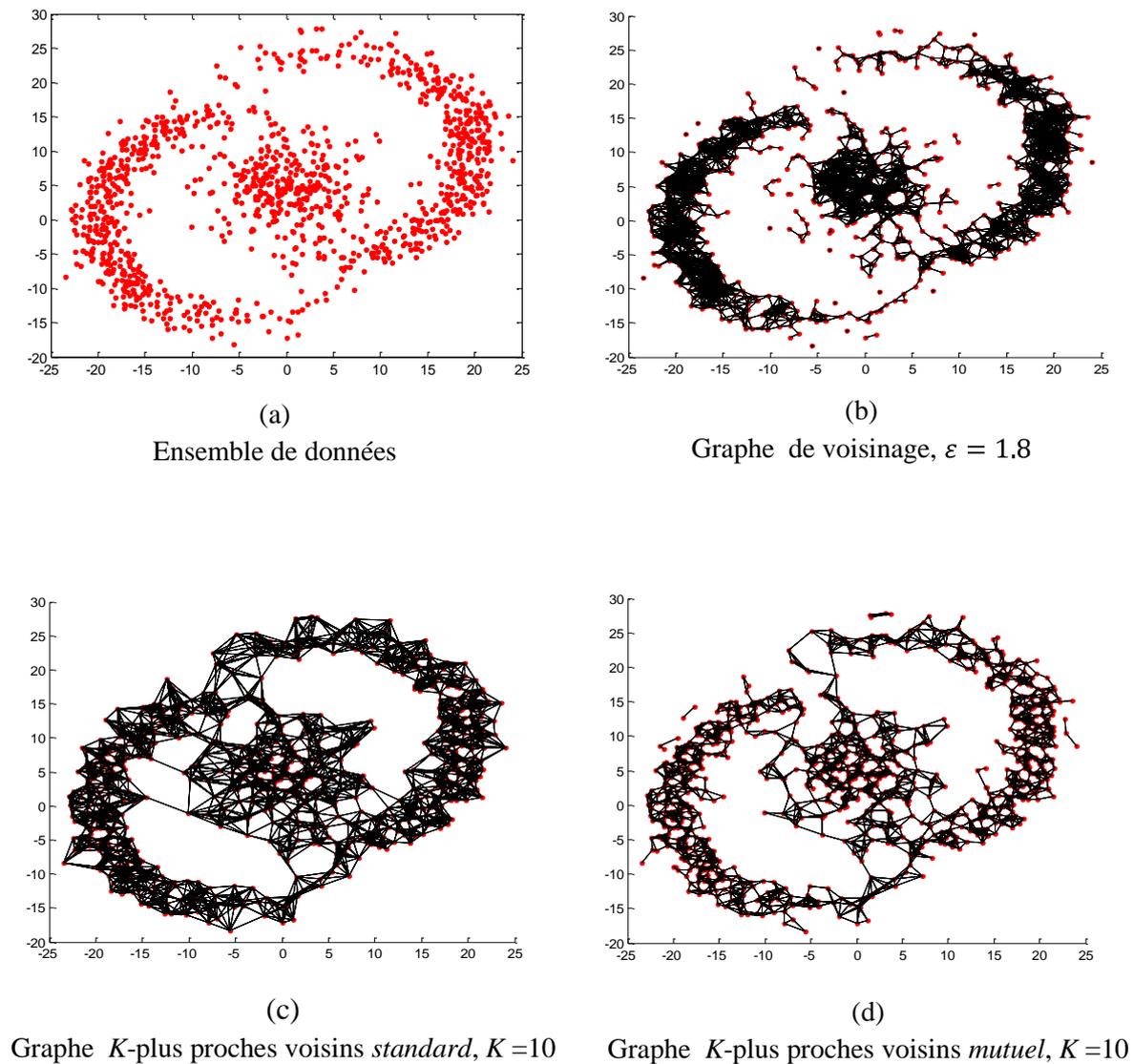


Figure III.5 : Différents graphes de similarité

Cependant, comme recommandation générale, il est préférable de travailler avec le graphe des K plus proches voisins comme un premier choix [26].

III.7.3 Paramètres du graphe de similarité

Une fois que le type du graphe de similarité est choisi, il faut fixer ses paramètres de connexion K ou ε . Il y a beaucoup de résultats théoriques sur la façon de relier les sommets entre eux dans un graphe, mais ces résultats ne sont valables que dans la limite de la taille des échantillons ($n \rightarrow \infty$).

Si on choisit K de l'ordre de $\log(n)$ dans le cas du graphe des K plus proches voisins *standard* ou *mutuel*, les n points de données seront reliés [46]. Des arguments similaires montrent que le paramètre ε dans le graphe de voisinage ε doit être choisi d'ordre de $(\log(n)/n)^d$ [47].

Pour le graphe de voisinage ε , nous suggérons de choisir ε tel que le graphe résultant est sûrement connexe. Pour déterminer la plus petite valeur de ε pour laquelle le graphe est connexe, il faut choisir ε comme la longueur de la plus longue arête dans un arbre recouvrant minimal du graphe totalement connecté. Celle-ci peut être facilement déterminée par n'importe quel algorithme d'arbre recouvrant minimal.

Enfin, si on utilise un graphe totalement connecté avec une fonction de similarité (par exemple la fonction de similarité *gaussienne*), alors l'échelle de la fonction de similarité devrait être choisie de telle sorte que le graphe obtenu a des propriétés similaires comme un graphe des K plus proches voisins ou un graphe de voisinage ε .

D'une manière générale, la classification spectrale peut être très sensible aux changements de graphe de similarité et au choix de ses paramètres. Malheureusement, il n'y a pas eu d'études systématiques qui examinent les effets du graphe de similarité et de ses paramètres sur la classification et aboutit à des règles bien justifiées. Trouver les règles qui ont une justification théorique est un thème de recherche qui est actuellement un champ très actif.

III.7.4 Choix du nombre de classes k

Le choix du nombre de classes k est un problème général pour tous les algorithmes de classification. En outre, un outil qui est particulièrement conçu pour la classification spectrale est l'heuristique d'écart propre (eigengap heuristic), qui peut être utilisé pour les trois matrices laplaciennes du graphe. L'objectif est de choisir le nombre k tel que toutes les valeurs propres $\lambda_1, \dots, \lambda_k$ sont très petites, mais λ_{k+1} est relativement grande [26].

III.7.5 Choix de la matrice laplacienne

Une question fondamentale liée à la classification spectrale est la quelle des trois matrices laplaciennes devrait être utilisée pour calculer les vecteurs propres. Avant de décider, il faut toujours comparer le degré de distribution du graphe de similarité. Si le graphe est très régulier et la plupart des sommets ont à peu près le même degré, alors toutes les laplaciennes seront très semblables les unes aux autres. Cependant, si les degrés dans le graphe sont très largement distribués, alors les laplaciennes seront considérablement différentes.

En littérature, il existe plusieurs arguments qui soutiennent l'utilisation de la matrice *normalisée* plutôt que la matrice *non normalisée*, et dans le cas *normalisé* à l'utilisation des vecteurs propres de L_{rw} plutôt que ceux de L_{sym} . La raison est que les vecteurs propres de L_{rw} sont des vecteurs indicateurs de classes $\vec{1}_{A_i}$, tandis que les vecteurs propres de L_{sym} sont en outre multipliés par $D^{1/2}$, qui pourrait conduire à des artefacts indésirables.

III.7.6 Exemple de classification spectrale

La figure (III.6) présente un exemple de classification spectrale par analyse des valeurs et vecteurs propres de la matrice laplacienne.

Soit un ensemble de données composé d'un échantillon aléatoire de 200 points $x_1, \dots, x_{200} \in \mathbb{R}$ établi selon un mélange de quatre *gaussiennes*.

La première ligne de cette figure montre l'histogramme d'un échantillon tiré à partir de cette distribution.

- Comme fonction de similarité sur cet ensemble de données, on choisit la fonction de similarité *gaussienne* (équation III.3) avec $\sigma = 1$.
- Comme graphe de similarité on considère : le graphe totalement connecté et le graphe des 10 plus proches voisins.

La figure (III.6) montre les premières valeurs propres et vecteurs propres de la matrice laplacienne *non normalisée* L et la matrice laplacienne *normalisée* L_{rw} . Dans la courbe des valeurs propres, on trace $\lambda_i = f(x_i)$ et dans les courbes des vecteurs propres $v = (v_1, \dots, v_{200})^T$ on trace $v_i = f(x_i)$.

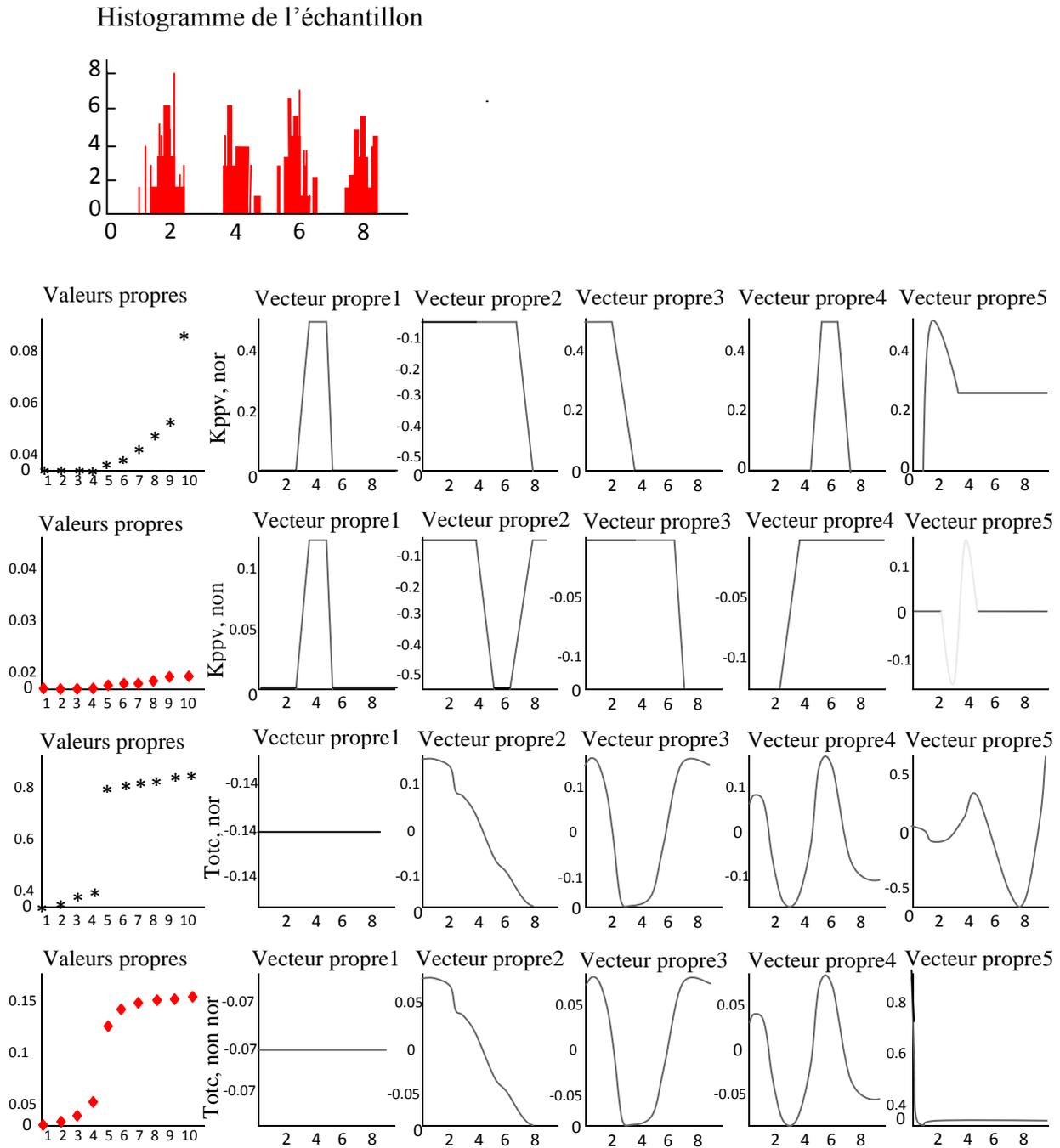


Figure III.6 [26]: Exemple de classification spectrale. La première et deuxième ligne représentent les valeurs propres et vecteurs propres de L_{r_w} et L utilisant le graphe des K plus proches voisins. La troisième et quatrième ligne représentent les valeurs propres et vecteurs propres de L_{r_w} et L utilisant le graphe totalement connecté.

La première et deuxième ligne de la figure (III.6) montrent les résultats du graphe des 10 plus proches voisins. On peut voir que les quatre premières valeurs propres sont égales à 0, et les vecteurs propres correspondants sont les vecteurs indicateurs des classes.

La raison est que les classes sont formées des groupes séparés dans le graphe des 10 plus proches voisins. La troisième et quatrième ligne présentent les résultats du graphe totalement connecté. Comme la fonction de similarité *gaussienne* est toujours positive, ce graphe est composé d'une seule composante connexe. Ainsi, la valeur propre 0 a une multiplicité 1, et le premier vecteur propre est le vecteur constant. Les vecteurs propres suivants portent l'information sur les classes. Par exemple dans le cas *non normalisé* (dernière ligne), si on coupe le *deuxième* vecteur propre à 0, alors la partie en dessous de 0 correspond à la classe 1 et 2, et la partie en dessus de 0 à la classe 3 et 4. De même, le *troisième* vecteur propre sépare la classe 1 et 4 des classes 2 et 3, et le *quatrième* vecteur propre sépare les classes 1 et 3 des classes 2 et 4. Donc, les quatre premiers vecteurs propres portent toute l'information sur les quatre classes. Dans tous les cas illustrés sur cette figure, la classification spectrale utilise la méthode de *k*-Means sur les quatre premiers vecteurs propres pour détecter d'une manière facile et exacte les quatre classes.

□ **Remarque** : les valeurs propres $\lambda_1, \dots, \lambda_4$ du graphe totalement connecté sont très petites, mais $\lambda_5, \dots, \lambda_{200}$ sont relativement grande (c'est un espace propre), donc *k* égale à 4.

III.8 Résumé de la méthode de classification spectrale

La figure (III.7) résume le principe de la méthode de classification spectrale

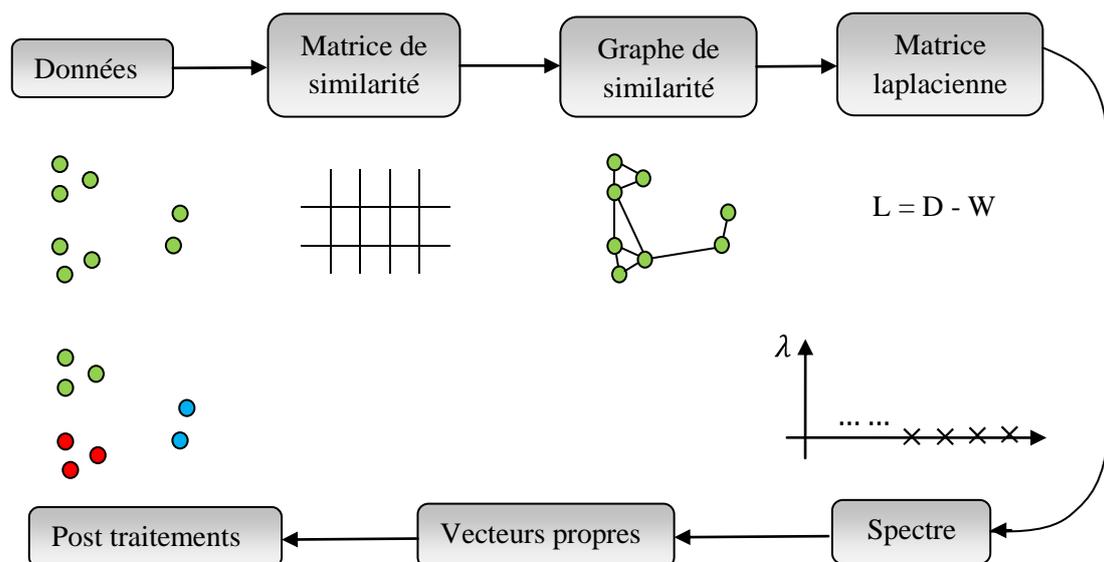


Figure III.7 : Différentes étapes de la classification spectrale

III.9 Avantages et inconvénients de la classification spectrale

La classification spectrale présente beaucoup d'avantages par rapport aux méthodes de regroupement classiques, on peut citer :

- La solution permet de résoudre un problème simple d'algèbre qui n'est pas difficile à implémenter.
- Peu coûteuse en termes de temps de calcul dans le cas de données de petite ou moyenne taille.
- Elle ne souffre pas du problème du minimum local inhérent aux méthodes d'optimisation.
- Elle peut détecter des classes non linéairement séparables.
- Capable de détecter des classes qui ne constituent pas des régions convexes dans l'espace, ce qui est presque impossible avec les méthodes de regroupement classiques (*k*-Means, *isodata*,... etc.) [49].
- Facilité de séparer des distributions qui se chevauchent.

Malgré tous les avantages que présente la classification spectrale, elle a quelques limites : la sensibilité aux choix des paramètres K , σ et ε qui définissent la construction du graphe, donc c'est une méthode qui est loin de détecter automatiquement le regroupement exact sur n'importe quel ensemble de données. Elle est très coûteuse en termes de temps de calcul dans le cas de données de grande taille (plusieurs méthodes sont proposées pour pallier à ce défaut)

III.10 Classification spectrale accélérée (Fast Spectral Clustering)

Un des objectifs des méthodes de classification non supervisée est de les appliquer sur de grands ensembles de données. Il a été démontré que la classification spectrale est plus efficace dans la recherche des classes que certains algorithmes traditionnels (*k*-Means) pour des petits ensembles de données.

Cependant, la classification spectrale souffre d'un problème de mémoire et de temps de calcul lorsque la taille des ensembles des données est très volumineuse.

Il existe deux approches fondamentales dans la littérature afin de pallier à ces défauts, la première est fondée sur une étape de prétraitement pour réduire la taille des ensembles de données par une sélection au hasard ou selon une certaine forme d'un certain nombre p d'individus (points représentatifs) fixé à *a priori* par l'utilisateur, les méthodes les plus connues

sont : la méthode KASP (*k*-Means-Based Approximate Spectral Clustering) et la méthode RASP (RP-tree-Based Approximate Spectral Clustering), se référer à [52]. La deuxième approche est fondée sur l'exploitation des techniques d'algèbre linéaire, les méthodes les plus connus sont : la méthode de Nyström [53] et la méthode d'Arnoldi [54] [55].

□ Rappelons que, dans le cadre de notre application (segmentation d'images couleurs) pour surpasser ces défauts nous choisissons la méthode de KASP de part sa simplicité et la facilité d'implémentation.

L'algorithme de cette méthode est donner par :

Algorithme de la méthode de classification spectrale accélérée (KASP)

Entrée: Ensemble des données x_1, \dots, x_n , le nombre des points représentatifs $\mathbb{p} < n$, nombre k de classes à construire.

1. Effectuer la méthode de *k*-Means sur l'ensemble des données x_1, \dots, x_n avec \mathbb{p} classes.
2. Calculer les centres des \mathbb{p} classes $y_1, \dots, y_{\mathbb{p}}$.
3. Exécuter un algorithme de classification spectrale sur les nouvelles données $y_1, \dots, y_{\mathbb{p}}$
4. Attribuer à chaque point de l'ensemble des données x_1, \dots, x_n la classe k qui a été attribuée au centre y_j **le plus proche** de x_i , avec $j = 1, \dots, \mathbb{p}$ et $i = 1, \dots, n$.

Sortie: Classes A_1, \dots, A_k .

III.11 Conclusion

Nous avons vu dans ce chapitre que le problème de classification spectrale est identique à un problème de partitionnement de graphe pour lequel les nœuds d'un même groupe sont similaires et les nœuds appartenant à des groupes différents sont dissimilaires. Nous avons vu aussi que la classification spectrale considérée comme une méthode de transformation des données d'un espace dans un autre espace où les classes sont facilement détectables. Elle se résume principalement à l'extraction des vecteurs propres de la matrice de similarité. Le succès de ses algorithmes est fondé sur le fait qu'ils ne font aucune supposition sur la forme des clusters. Cependant, un grand soin devrait être pris dans le choix des paramètres puisque la classification spectrale peut être très sensible à l'évolution du graphe de similarité et de ses paramètres.

CHAPITRE IV



Tests et résultats

IV.1 Introduction

Dans ce chapitre, nous présenterons en deux parties les résultats de classification spectrale. La première partie expose les résultats de la méthode spectrale appliquée sur des données artificielles à 2D puis à 3D. L'utilisation des données artificielles nous permettent de tester la méthode dans différentes situations. La deuxième partie est consacrée à la présentation des résultats de la segmentation des images couleurs en utilisant la méthode de classification spectrale.

Afin de démontrer l'intérêt de la méthode de classification spectrale, nous proposons de comparer les résultats de cette nouvelle approche aux ceux obtenus avec la méthode de k -Means (méthode classique très utilisée dans le domaine de classification).

Comme on a pu le voir dans le chapitre précédent, la classification spectrale utilise quatre types de graphes : graphe totalement connecté, graphe de voisinage ε , graphe des K plus proches voisins *standard* et *mutuel*. Pour chaque graphe, nous utiliserons trois types d'algorithmes : non normalisée, normalisée selon « Shi et Malik » et normalisée selon « Ng, Jordan et Weiss », dont l'utilisation nécessite le choix à *a priori* des paramètres k , ε , σ , K .

Pour chaque type d'algorithme, nous ferons varier le type de graphe ainsi que les paramètres cités précédemment. Nous afficherons et nous commenterons les résultats à chaque fois.

La qualité de classification est une question récurrente en analyse de données pour valider une nouvelle méthode ou pour mesurer la pertinence des résultats. Afin d'évaluer objectivement la qualité de classification ou de segmentation, nous proposerons de quantifier les résultats que nous allons obtenir grâce à :

- La matrice de confusion et le taux de classification (*annexe B*) pour le cas de classification des données artificielles.
- Le critère de Borsoetti (*annexe C*) pour le cas de la segmentation d'images couleurs par classification spectrale.

Nous avons implémenté les méthodes de classification spectrale sous **Matlab R2009a** sur une architecture du type PC avec un microprocesseur Pentium(R) Dual-Core cadencé à 2.00 GHz, avec 3 Go de RAM.

Les paramètres des algorithmes de classification spectrale à savoir le nombre de classes k , le rayon ε du graphe de voisinage ε , le nombre des plus proches voisins K du graphe des $KPPV$ et le paramètre de la fonction de similarité gaussienne σ ont été fixés de manière expérimentale de telle sorte que les méthodes fournissent un résultat correct.

Pour conclure ce chapitre, nous récapitulerons et nous discuterons les principaux éléments abordés et les résultats obtenus.

IV.2 Données artificielles à 2D

Les données artificielles choisies pour ce cas sont constituées de trois classes, deux ont une forme de croissant (en rouge et en bleue, 450 points pour chacune) et une classe gaussienne (en noire, 250 points), avec présence de chevauchement entre les trois classes pour bien tester l'efficacité de la méthode spectrale sur des données chevauchées. Pour le choix de la fonction de similarité, nous avons choisi le noyau gaussien. Le nombre de classes k est fixé à 3. Les paramètres σ , ε et K ont été fixés de manière expérimentale afin de comparer d'une part l'influence des différents graphes sur le résultat de classification et entre les algorithmes de classification spectrale d'une autre part.

Les figures (IV.1) et (IV.2) représentent les données originales. La deuxième figure représente les données de référence avec lesquelles nous pouvons comparer et évaluer les résultats de classification spectrale et ceux de la méthode de k -Means.

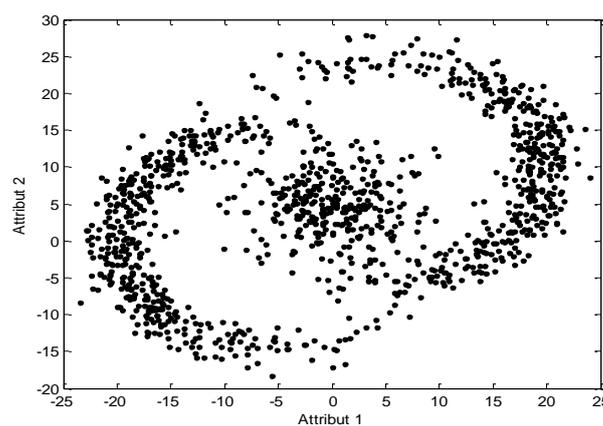


Figure IV.1 : Données originales brutes

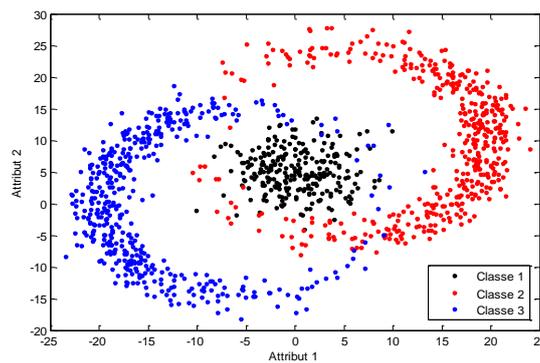


Figure IV.2 : Données de référence

La figure (IV.3) montre la difficulté de la méthode de *k*-Means à séparer les deux croissants, par contre, elle montre l'efficacité de cette méthode à détecter la gaussienne, cela est dû à son principe de fonctionnement fondé sur la séparation linéaire des classes. Ces constatations sont confirmées par la matrice de confusion du tableau (IV.1), 98.40 % d'individus bien classés pour la gaussienne, 72.88 % et 85.33 % pour les deux croissants.

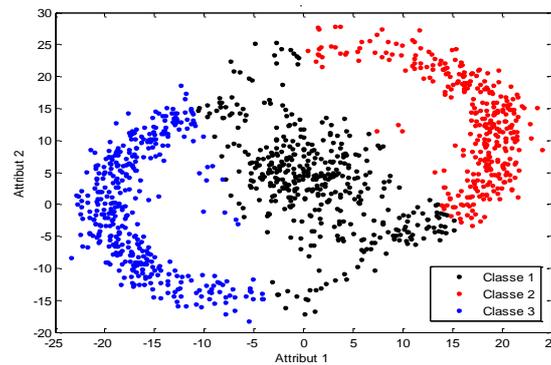


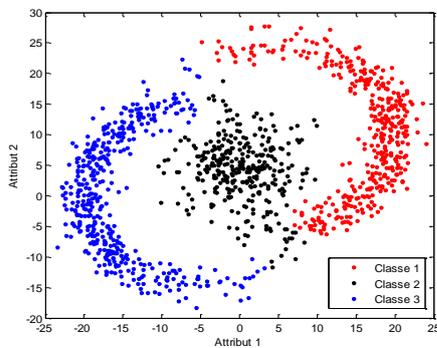
Figure IV.3 : Résultat de classification par la méthode de *k*-Means

	Matrice de confusion MC	Taux de classification T_c (%)	Temps de calcul t_c (s)
Résultat de la méthode de <i>k</i> -Means	$\begin{pmatrix} 246 & 2 & 2 \\ 113 & 328 & 9 \\ 64 & 2 & 384 \end{pmatrix}$	$T_c = 83.30 \%$	$t_c=0.92s$

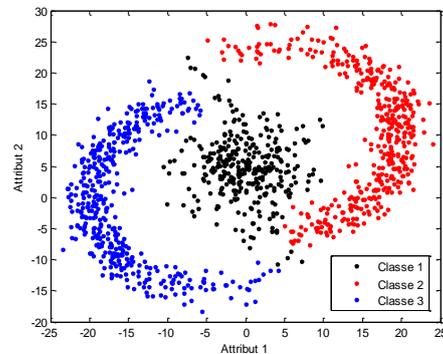
Tab IV.1 : Résultats de classification par la méthode de *k*-Means

IV.2.1 Classification spectrale non normalisée

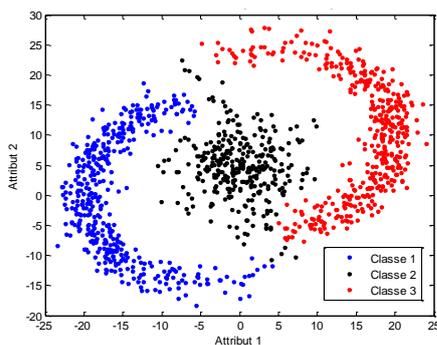
La figure (IV.4) donne les résultats de l'algorithme non normalisé utilisant la matrice laplacienne $L = D - W$ pour chaque type de graphe



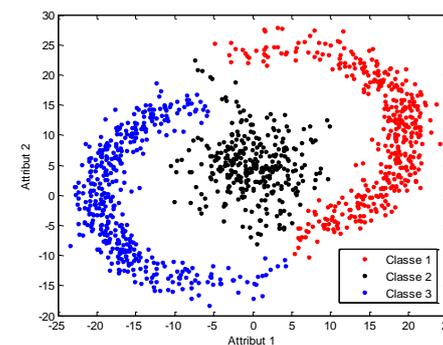
(a) Graphe totalement connecté



(b) Graphe de voisinage epsilon $\varepsilon = 6$



(c) Graphe des K plus proches voisins *standard* $K = 120$



(d) Graphe des K plus proches voisins *mutuel* $K = 70$

Figure IV.4 : Résultats de classification par la méthode spectrale non normalisée

Interprétation :

Les figures (IV.4.a), (IV.4.b), (IV.4.c) et (IV.4.d) montrent l'efficacité de la méthode spectrale à détecter les deux croissants ainsi que la gaussienne (contrairement à la méthode de k -Means). Le tableau (IV.2) montre que le taux de classification obtenu par la méthode spectrale est beaucoup meilleur que celui obtenu par la méthode de k -Means. Le taux d'erreur est dû au chevauchement entre les différentes classes. Nous constatons aussi que la procédure de construction des classes par la méthode spectrale prend plus de temps que la méthode de k -Means.

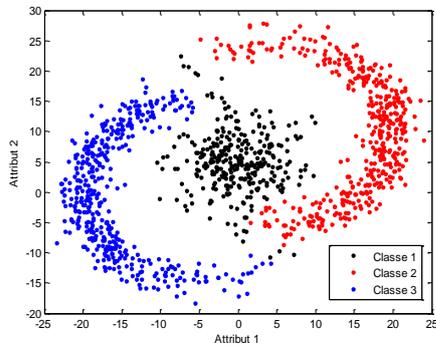
	Paramètre	Matrice de confusion MC	Taux de classification T_c	Temps de calcul t_c
Graphe totalement connecté	$\sigma = 0.4$	$\begin{pmatrix} 250 & 0 & 0 \\ 46 & 397 & 7 \\ 27 & 3 & 420 \end{pmatrix}$	$T_c = 92.78 \%$	$t_c = 25.80 \text{ s}$
	$\sigma = 0.38$	$\begin{pmatrix} 250 & 0 & 0 \\ 37 & 411 & 2 \\ 23 & 5 & 422 \end{pmatrix}$	$T_c = 94.17 \%$	$t_c = 27.25 \text{ s}$
Graphe de voisinage ε	$\sigma = 0.4$ $\varepsilon = 6$	$\begin{pmatrix} 250 & 0 & 0 \\ 42 & 406 & 2 \\ 24 & 5 & 421 \end{pmatrix}$	$T_c = 93.65 \%$	$t_c = 33.69 \text{ s}$
	$\sigma = 0.38$ $\varepsilon = 6$	$\begin{pmatrix} 250 & 0 & 0 \\ 37 & 411 & 2 \\ 23 & 5 & 422 \end{pmatrix}$	$T_c = 94.17 \%$	$t_c = 33.87 \text{ s}$
Graphe des KPPV <i>standard</i>	$\sigma = 0.4$ $K = 120$	$\begin{pmatrix} 250 & 0 & 0 \\ 42 & 406 & 2 \\ 24 & 5 & 421 \end{pmatrix}$	$T_c = 93.65 \%$	$t_c = 33.81 \text{ s}$
	$\sigma = 0.38$ $K = 169$	$\begin{pmatrix} 250 & 0 & 0 \\ 37 & 411 & 2 \\ 23 & 5 & 422 \end{pmatrix}$	$T_c = 94.17 \%$	$t_c = 34.03 \text{ s}$
Graphe des KPPV <i>mutuel</i>	$\sigma = 0.4$ $K = 70$	$\begin{pmatrix} 250 & 0 & 0 \\ 44 & 404 & 2 \\ 20 & 8 & 422 \end{pmatrix}$	$T_c = 93.56 \%$	$t_c = 33.24 \text{ s}$
	$\sigma = 0.38$ $K = 130$	$\begin{pmatrix} 250 & 0 & 0 \\ 37 & 411 & 2 \\ 23 & 5 & 422 \end{pmatrix}$	$T_c = 94.17 \%$	$t_c = 34.63 \text{ s}$

Tab IV.2 : Résultats de classification spectrale non normalisée

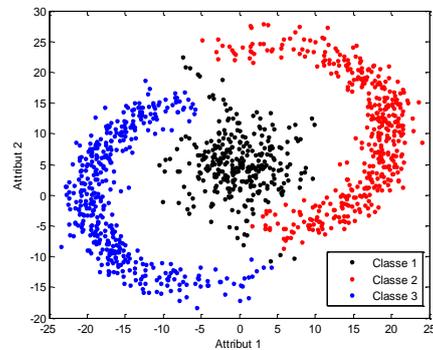
D'après le tableau (IV.2), nous remarquons que la meilleure classification est obtenue lorsque nous appliquons l'un des trois graphes suivants : graphe de voisinage ε , graphe des K plus proche voisins *standard* ou graphe des K plus proche voisins *mutuel*, avec un taux de classification $T_c \approx 93.65 \%$. Notons que nous pouvons améliorer ce taux en ajustant les paramètres σ , ε et K , le taux de classification maximal obtenu est : $T_{cMax} = 94.17 \%$

IV.2.2 Classification spectrale normalisée selon Shi et Malik

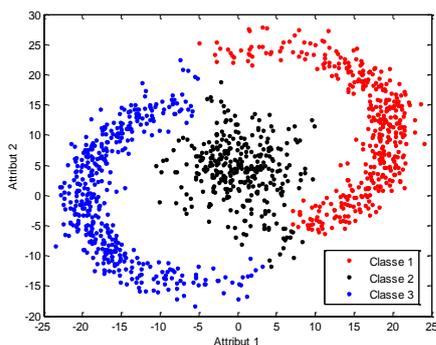
La figure (IV.5) donne les résultats de l'algorithme normalisé selon Shi et Malik utilisant la matrice laplacienne $L_{rw} = D^{-1} L$ pour chaque graphe



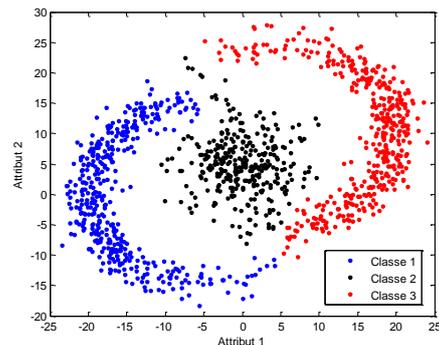
(a) Graphe totalement connecté



(b) Graphe de voisinage epsilon $\epsilon = 6$



(c) Graphe des K plus proches voisins *standard* $K = 120$



(d) Graphe des K plus proches voisins *mutuel* $K = 70$

Figure IV.5 : Résultats de classification par la méthode spectrale normalisée selon SM

Interprétation :

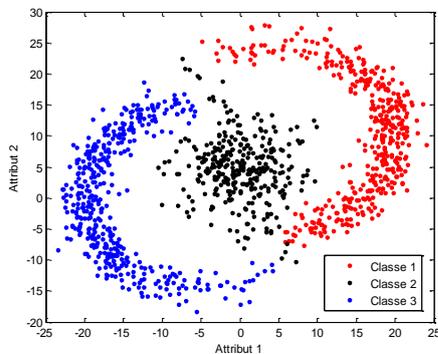
Avec les mêmes valeurs des paramètres que précédemment ($\sigma = 0.4$, $\epsilon = 6$, $K = 120$ pour le graphe des *KPPV standard*, $K = 70$ pour le graphe des *KPPV mutuel*), nous constatons que les résultats obtenus par l'algorithme normalisé selon Shi et Malik sont meilleurs que ceux obtenus par l'algorithme non normalisé quand nous appliquons le graphe totalement connecté et le graphe de voisinage ϵ (tableau IV.3). Le taux de classification maximal obtenu par cet algorithme est : $T_{cMax} = 94.43 \%$

	Paramètres du graphe	Matrice de confusion MC	Taux de classification T_c	Temps de calcul t_c
Graphe totalement connecté	$\sigma = 0.4$	$\begin{pmatrix} 249 & 1 & 0 \\ 32 & 416 & 2 \\ 23 & 6 & 421 \end{pmatrix}$	$T_c = 94.43 \%$	$t_c = 28.38 \text{ s}$
Graphe de voisinage ε	$\sigma = 0.4$ $\varepsilon = 6$	$\begin{pmatrix} 249 & 1 & 0 \\ 32 & 416 & 2 \\ 23 & 6 & 421 \end{pmatrix}$	$T_c = 94.43 \%$	$t_c = 36.87 \text{ s}$
Graphe des KPPV standard	$\sigma = 0.4$ $K = 120$	$\begin{pmatrix} 250 & 0 & 0 \\ 46 & 397 & 7 \\ 27 & 3 & 420 \end{pmatrix}$	$T_c = 92.78 \%$	$t_c = 36.52 \text{ s}$
	$\sigma = 0.4$ $K = 130$	$\begin{pmatrix} 249 & 1 & 0 \\ 32 & 416 & 2 \\ 23 & 6 & 421 \end{pmatrix}$	$T_c = 94.43 \%$	$t_c = 36.68 \text{ s}$
Graphe des KPPV mutuel	$\sigma = 0.4$ $K = 70$	$\begin{pmatrix} 250 & 0 & 0 \\ 44 & 404 & 2 \\ 20 & 8 & 422 \end{pmatrix}$	$T_c = 93.56 \%$	$t_c = 36.14 \text{ s}$
	$\sigma = 0.34$ $K = 75$	$\begin{pmatrix} 250 & 0 & 0 \\ 37 & 411 & 2 \\ 22 & 6 & 422 \end{pmatrix}$	$T_c = 94.17 \%$	$t_c = 36.29 \text{ s}$

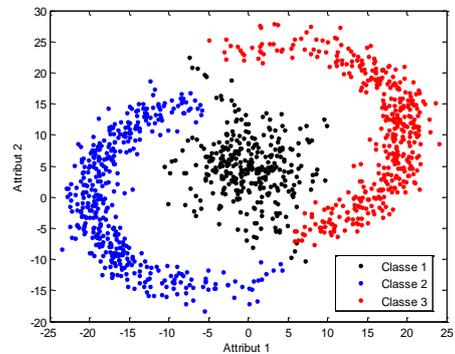
Tab IV.3 : Résultats de classification spectrale normalisée selon SM

IV.2.3 Classification spectrale normalisée selon Ng, Jordan et Weiss

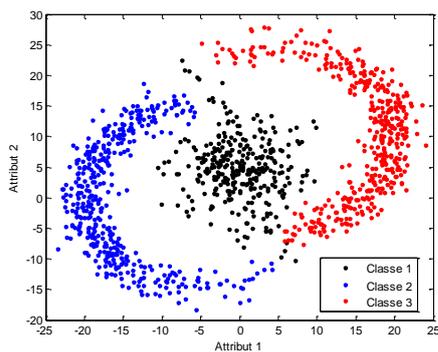
La figure (IV.6) donne les résultats de l’algorithme normalisé selon Ng, Jordan et Weiss utilisant la matrice laplacienne $L_{sym} = D^{-1/2} L D^{-1/2}$ pour chaque graphe



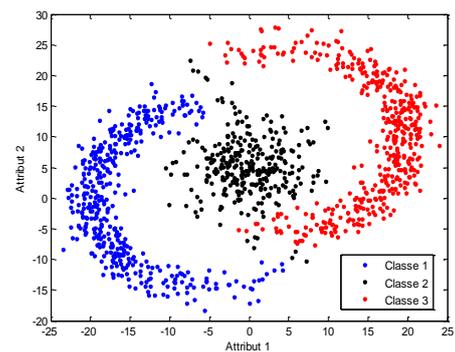
(a) Graphe totalement connecté



(b) Graphe de voisinage epsilon $\varepsilon = 6$



(c) Graphe des K plus proches voisins *standard* $K = 120$



(d) Graphe des K plus proches voisins *mutuel* $K = 70$

Figure IV.6 : Résultats de classification par la méthode spectrale normalisée selon NJW

Interprétation :

Avec les mêmes valeurs des paramètres ($\sigma = 0.4$, $\varepsilon = 6$, $K = 120$ pour le graphe des *KPPV standard* et $K = 70$ pour le graphe des *KPPV mutuel*), nous constatons que les résultats obtenus par l’algorithme normalisé selon Ng, Jordan et Weiss sont meilleurs que ceux obtenus par l’algorithme normalisé selon Shi et Malik utilisant le graphe des K plus proches voisins *standard* et *mutuel*.

	Paramètres du graphe	Matrice de confusion MC	Taux de classification T_c	Temps de calcul t_c
Graphe totalement connecté	$\sigma = 0.4$	$\begin{pmatrix} 250 & 0 & 0 \\ 44 & 404 & 2 \\ 24 & 4 & 422 \end{pmatrix}$	$T_c = 93.56 \%$	$t_c = 44.15 \text{ s}$
Graphe de voisinage ε	$\sigma = 0.4$ $\varepsilon = 6$	$\begin{pmatrix} 250 & 0 & 0 \\ 44 & 404 & 2 \\ 20 & 8 & 422 \end{pmatrix}$	$T_c = 93.56 \%$	$t_c = 52.14 \text{ s}$
Graphe des KPPV standard	$\sigma = 0.4$ $K = 120$	$\begin{pmatrix} 250 & 0 & 0 \\ 44 & 404 & 2 \\ 20 & 8 & 422 \end{pmatrix}$	$T_c = 93.56 \%$	$t_c = 54.11 \text{ s}$
	$\sigma = 0.44$ $K = 65$	$\begin{pmatrix} 245 & 2 & 3 \\ 24 & 424 & 2 \\ 19 & 6 & 425 \end{pmatrix}$	$T_c = 95.13 \%$	$t_c = 53.17 \text{ s}$
Graphe des KPPV mutuel	$\sigma = 0.4$ $K = 70$	$\begin{pmatrix} 248 & 2 & 0 \\ 29 & 419 & 2 \\ 23 & 5 & 422 \end{pmatrix}$	$T_c = 94.69 \%$	$t_c = 52.24 \text{ s}$

Tab IV.4 : Résultats de classification spectrale normalisée selon NJW

Nous constatons aussi que les résultats obtenus par l'algorithme normalisé selon Ng, Jordan et Weiss sont meilleurs que ceux obtenus par l'algorithme non normalisé lorsque nous utilisons le graphe totalement connecté et le graphe des K plus proches voisins *mutuel*. Nous remarquons aussi que cet algorithme prend un temps de calcul légèrement supérieur aux ceux des algorithmes précédents, ceci est dû à l'étape de normalisation des vecteurs propres. Le taux de classification maximal obtenu par cet algorithme est : $T_{cMax} = 95.13 \%$

□ Par comparaison des trois taux de classification maximaux T_{cMax} , nous concluons que pour ces données (deux croissants et une gaussienne), il est préférable d'utiliser l'algorithme normalisé selon Ng, Jordan et Weiss avec le graphe des K plus proches voisins *standard*.

IV.3 Données artificielles à 3D

Pour ce cas, les données sont constituées de deux classes (deux anneaux enchainés, un en rouge et un en noir). Un premier test est réalisé sur deux anneaux de 250 individus pour chacun soit 500 au total (figure (IV.7)). Le second test est réalisé sur deux anneaux de 1000 individus pour chaque anneau soit 2000 au total (figure (IV.8)). Pour le choix de la fonction de similarité, nous avons choisi le noyau gaussien. Le nombre de classes k est fixé à 2.

□ Notons que dans ces tests, nous présentons uniquement les résultats de classification spectrale non normalisée en faisant varier le type de graphe.

Les figures (IV.11) et (IV.12) montrent les résultats de la classification par la méthode de k -Means. Les taux de classification sont consignés dans le tableau (IV.5). Les résultats des méthodes spectrales sont illustrés sur les figures (IV.13) et (IV.14) et dans les tableaux (IV.6) et (IV.7).

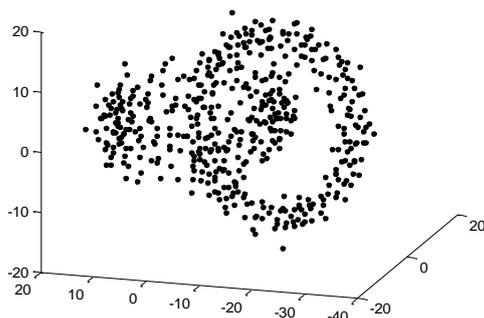


Figure IV.7 :

Données originaux brutes 500 points

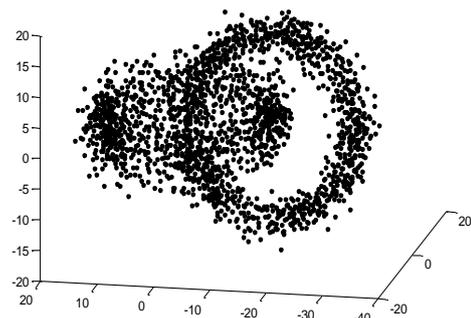


Figure IV.8 :

Données originaux brutes 2000 points

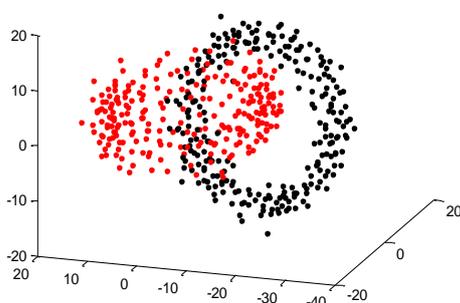


Figure IV.9 :

Données de référence (500 points)

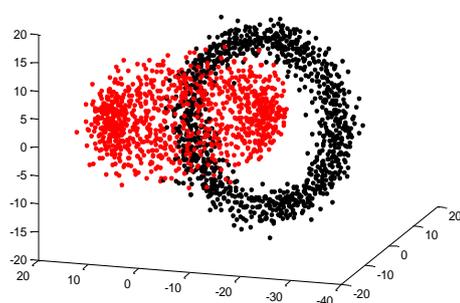


Figure IV.10 :

Données de référence (2000 points)

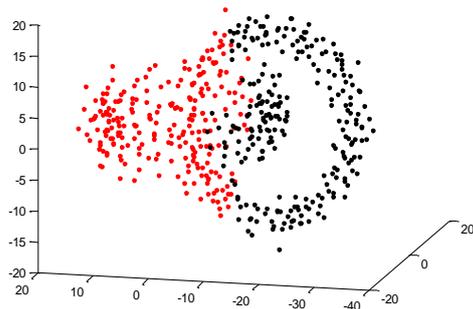


Figure IV.11 : Résultat de la méthode de *k*-Means (500 points)

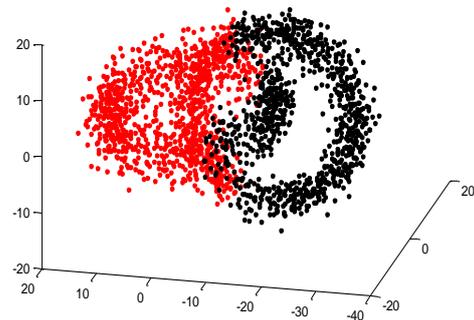


Figure IV.12 : Résultat de la méthode de *k*-Means (2000 points)

	Matrice de confusion MC	Taux de classification T_c (%)	Temps de calcul t_c (s)
Méthode de <i>k</i> -Means avec 500 points	$\begin{pmatrix} 161 & 89 \\ 88 & 162 \end{pmatrix}$	$T_c = 64.60 \%$	$t_c = 0.36$ s
Méthode de <i>k</i> -Means avec 2000 points	$\begin{pmatrix} 648 & 352 \\ 348 & 652 \end{pmatrix}$	$T_c = 65 \%$	$t_c = 0.65$ s

Tab IV.5 : Résultats de classification par la méthode de *k*-Means à 3D

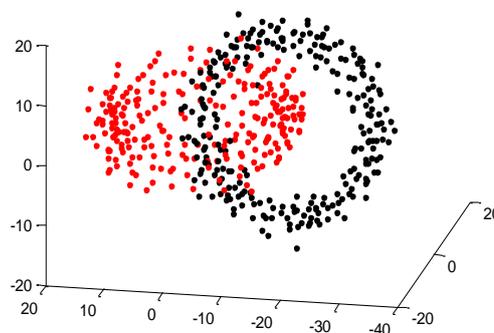


Figure IV.13 :
Classification spectrale non normalisée (500 points)

	Paramètres	Matrice de confusion MC	Taux de classification T_c	Temps de calcul t_c
Graphe totalement connecté	$\sigma = 0.5$	$\begin{pmatrix} 250 & 0 \\ 1 & 249 \end{pmatrix}$	$T_c = 99.80 \%$	$t_c = 1.04 \text{ s}$
Graphe de voisinage ε	$\sigma = 0.5$ $\varepsilon = 10$	$\begin{pmatrix} 250 & 0 \\ 1 & 249 \end{pmatrix}$	$T_c = 99.80 \%$	$t_c = 1.75 \text{ s}$
Graphe des KPPV standard	$\sigma = 0.5$ $K = 20$	$\begin{pmatrix} 250 & 0 \\ 1 & 249 \end{pmatrix}$	$T_c = 99.80 \%$	$t_c = 1.87 \text{ s}$
Graphe des KPPV mutuel	$\sigma = 0.5$ $K = 30$	$\begin{pmatrix} 250 & 0 \\ 1 & 249 \end{pmatrix}$	$T_c = 99.80 \%$	$t_c = 1.83 \text{ s}$

Tab IV.6 : Résultats de classification spectrale non normalisée à 3D (500 points)

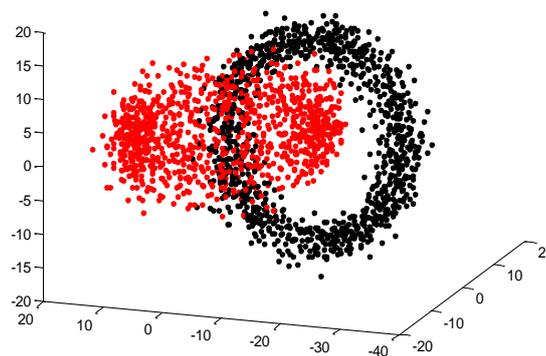


Figure IV.14 :
Classification spectrale non normalisée (2000 points)

	Paramètres	Matrice de confusion MC	Taux de classification T_c	Temps de calcul t_c
Graphe totalement connecté	$\sigma = 0.5$	$\begin{pmatrix} 1000 & 0 \\ 1 & 999 \end{pmatrix}$	$T_c = 99.95 \%$	$t_c = 46.83 \text{ s}$
Graphe de voisinage ε	$\sigma = 0.5$ $\varepsilon = 10$	$\begin{pmatrix} 1000 & 0 \\ 1 & 999 \end{pmatrix}$	$T_c = 99.95 \%$	$t_c = 89.13 \text{ s}$
Graphe des KPPV standard	$\sigma = 0.5$ $K = 60$	$\begin{pmatrix} 1000 & 0 \\ 1 & 999 \end{pmatrix}$	$T_c = 99.95 \%$	$t_c = 90.19 \text{ s}$
Graphe des KPPV mutuel	$\sigma = 0.5$ $K = 80$	$\begin{pmatrix} 1000 & 0 \\ 1 & 999 \end{pmatrix}$	$T_c = 99.95 \%$	$t_c = 90.51 \text{ s}$

Tab IV.7 : Résultats de classification spectrale non normalisée à 3D (2000 points)

Interprétation :

D'après les résultats obtenus dans figures (IV.11) et (IV.12), la méthode de k -Means n'a pas réussi à discerner les deux anneaux, cela est dû à son principe de fonctionnement fondé sur la séparation linéaire des classes (incapacité de détecter des classes non linéairement séparables).

Ces constatations sont confirmées par le taux de classification ($T_c \approx 65 \%$) (Tab IV.5). Par contre, les figures (IV.13) et (IV.14) nous montrent l'efficacité de la méthode spectrale à détecter les deux anneaux (taux de classification $T_c \approx 100 \%$).

D'après les tableaux (IV.6) et (IV.7), nous constatons que le choix du type de graphe n'influe pas sur les résultats de classification spectrale, cela revient à la nature des données choisies (degré de chevauchement très faible).

Nous constatons aussi que la procédure de construction des classes par la méthode spectrale à 3D prend plus de temps que la méthode spectrale à 2D (contrairement à la méthode de k -Means où la dimension des données n'influe pas sur le temps de calcul).

□ Notons que les algorithmes de classification spectrale normalisée à 3D selon Shi et Malik et selon Ng, Jordan, et Weiss ont donné pratiquement le même résultat que celui de l'algorithme de classification spectrale non normalisée.

□ Notons aussi que, si nous augmentons le nombre de points à 4000 (individus), l'espace mémoire occupé dans la mémoire RAM par les différentes matrices (matrice de distance, matrice laplacienne, matrice qui contient les vecteurs propres) devient important et le temps de calcul passe de quelques dizaines de secondes pour des données inférieures à 2000 individus à quelques heures pour des données supérieures à 2000 individus).

Les deux problèmes précédents représentent les inconvénients fondamentaux de la méthode de classification spectrale.

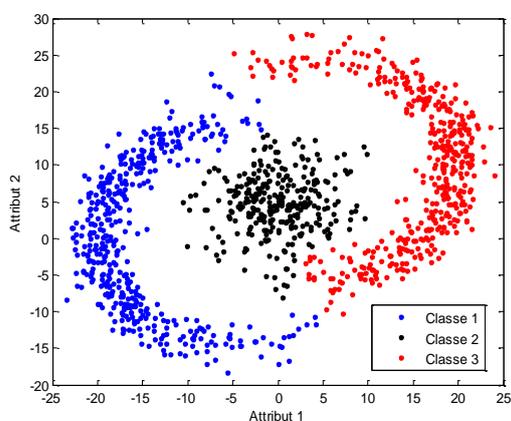
Afin de pallier à ces problèmes, nous proposons de sélectionner un nombre p d'individus qui serviront comme des échantillons, puis nous appliquerons la méthode spectrale sur ces p individus, ensuite nous utiliserons la méthode du plus proche voisin pour classer les individus non classés, c'est le principe de l'algorithme de classification spectrale accélérée KASP (chapitre III).

IV.4 Classification spectrale accélérée KASP sur des données artificielles

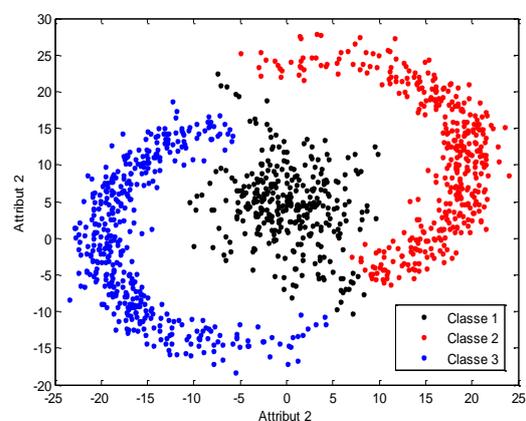
Avant d'appliquer la méthode de classification spectrale accélérée KASP sur des grands ensembles de données réelles, et afin d'étudier son comportement, nous proposons de la tester sur des données artificielles. Pour le cas $2D$, nous choisissons les données utilisées dans la partie IV.2, et pour le cas $3D$, nous choisissons les données utilisées dans la partie IV.3.

IV.4.1 Cas $2D$

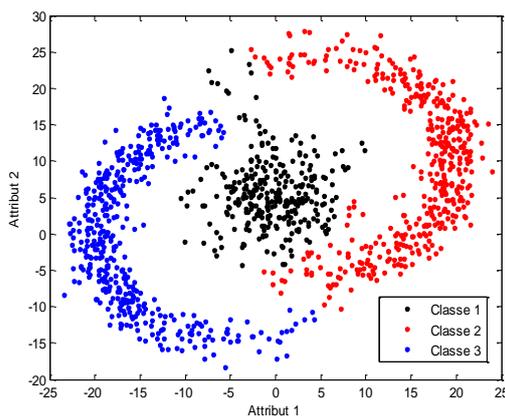
Les résultats de classification obtenus par la méthode spectrale accélérée KASP sont illustrés sur la figure (IV.15) et le tableau (IV.8).



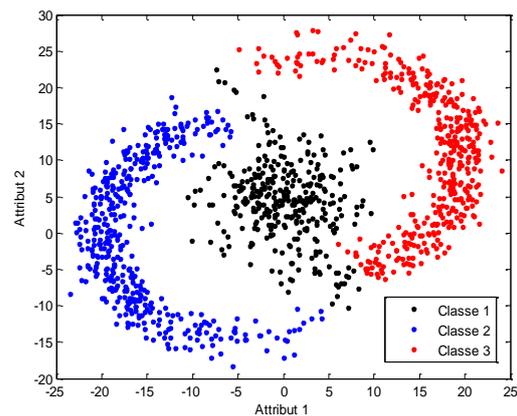
(a) Graphe totalement connecté



(b) Graphe de voisinage epsilon $\epsilon = 6$



(c) Graphe des K plus proches voisins *standard* $K = 120$



(d) Graphe des K plus proches voisins *mutuel* $K = 70$

Figure IV.15: Résultats de classification spectrale KASP non normalisée

	Paramètres	Matrice de confusion MC	Taux de classification T_c	Temps de calcul t_c
Graphe totalement connecté	$\sigma = 0.4$ $\mathbb{p} = 200$	$\begin{pmatrix} 249 & 1 & 0 \\ 27 & 415 & 8 \\ 15 & 8 & 427 \end{pmatrix}$	$T_c = 94.86 \%$	$t_c = 1.74 \text{ s}$
Graphe de voisinage ε	$\sigma = 0.4$ $\varepsilon = 6$ $\mathbb{p} = 200$	$\begin{pmatrix} 250 & 0 & 0 \\ 53 & 395 & 2 \\ 25 & 3 & 422 \end{pmatrix}$	$T_c = 92.78 \%$	$t_c = 1.69 \text{ s}$
Graphe des KPPV standard	$\sigma = 0.4$ $K = 120$ $\mathbb{p} = 200$	$\begin{pmatrix} 246 & 4 & 0 \\ 29 & 419 & 2 \\ 17 & 11 & 422 \end{pmatrix}$	$T_c = 94.52 \%$	$t_c = 1.78 \text{ s}$
Graphe des KPPV mutuel	$\sigma = 0.4$ $K = 70$ $\mathbb{p} = 200$	$\begin{pmatrix} 250 & 0 & 0 \\ 52 & 396 & 2 \\ 25 & 3 & 422 \end{pmatrix}$	$T_c = 92.86 \%$	$t_c = 1.86 \text{ s}$

Tab IV.8 : Résultats de classification spectrale KASP non normalisée

Interprétation :

D'après le tableau (IV.8), nous constatons que la procédure de construction des classes par la méthode spectrale accélérée à 2D prend un temps de calcul beaucoup moins que la méthode spectrale classique à 2D ($t_c \approx 2 \text{ s}$ pour la méthode KASP et $t_c \approx 35 \text{ s}$ pour la méthode spectrale classique).

Nous remarquons aussi que la méthode de classification spectrale KASP présente peu de dégradation au niveau de la précision de classification par rapport à la méthode spectrale classique, cela est dû au nombre de points représentatifs \mathbb{p} choisis. Autrement dit, si nous augmentons \mathbb{p} on a tendance à se rapprocher des résultats de classification spectrale classique.

Nous voyons aussi que parfois, les résultats de la méthode KASP sont améliorés par rapport à ceux de la méthode spectrale classique, cela est probablement dû à l'initialisation aléatoire des points représentatifs \mathbb{p} par la méthode de k -Means, ainsi qu'à l'effet de l'affectation par la méthode du plus proche voisin (étape 4 de l'algorithme de classification spectrale accélérée KASP).

IV.4.2 Cas 3D

Le résultat de classification obtenus par la méthode spectrale accélérée KASP sont illustrés sur la figure (IV.16) et le tableau (IV.9).

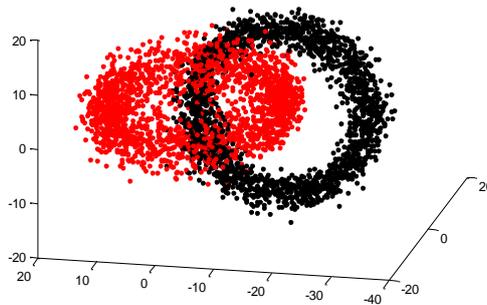


Figure IV.16 :

Résultats de classification spectrale accélérée KASP non normalisée (4000 points)

	Paramètres	Matrice de confusion MC	Taux de classification T_c	Temps de calcul t_c
Graphe totalement connecté	$\sigma = 0.5$ $\mathbb{p} = 200$	$\begin{pmatrix} 2000 & 0 \\ 2 & 1998 \end{pmatrix}$	$T_c = 99.95 \%$	$t_c = 18.78 \text{ s}$
Graphe de voisinage ε	$\sigma = 0.5$ $\varepsilon = 10$ $\mathbb{p} = 200$	$\begin{pmatrix} 2000 & 0 \\ 3 & 1997 \end{pmatrix}$	$T_c = 99.92 \%$	$t_c = 11.36 \text{ s}$
Graphe des KPPV standard	$\sigma = 0.5$ $K = 60$ $\mathbb{p} = 200$	$\begin{pmatrix} 2000 & 0 \\ 4 & 1996 \end{pmatrix}$	$T_c = 99.90 \%$	$t_c = 14.66 \text{ s}$
Graphe des KPPV mutuel	$\sigma = 0.5$ $K = 80$ $\mathbb{p} = 200$	$\begin{pmatrix} 2000 & 0 \\ 2 & 1998 \end{pmatrix}$	$T_c = 99.95 \%$	$t_c = 17.17 \text{ s}$

Tab IV.9 : Résultats de classification spectrale accélérée KASP non normalisée en 3D (4000 points)

Interprétation :

D'après le tableau (IV.9), nous constatons que la procédure de construction des classes par la méthode spectrale accélérée KASP en 3D pour 4000 points prend très peu de temps que la méthode spectrale classique à 3D pour 4000 points ($t_c \approx 18$ s pour la méthode KASP et quelques heures pour la méthode spectrale classique). Notons aussi que la méthode spectrale accélérée ne souffre pas du problème de stockage mémoire, contrairement à la méthode spectrale classique.

□ Compte tenu de cet avantage indéniable, nous utiliserons la méthode de classification spectrale accélérée KASP pour la segmentation d'images couleurs (données volumineuses).

IV.5 Segmentation d'images couleurs par classification

Dans cette partie, nous présentons les résultats de segmentation d'images couleurs par classification non supervisée de pixels en utilisant la méthode KASP. Et avant tout, nous devons rappeler quelques notions fondamentales sur la segmentation d'images.

IV.5.1 Rappels sur la segmentation d'images

La segmentation d'images est une étape importante pour l'extraction des informations qualitatives de l'image. Elle consiste à partitionner l'image en régions disjointes selon un critère prédéfini (niveau de gris, couleur, texture, ...) afin d'extraire les différents objets présents dans l'image. Une multitude de méthodes de segmentation sont présentées dans la littérature, elles se repartissent en trois grandes familles :

- La segmentation par approche région
- La segmentation par approche contour
- La segmentation par approche coopérative

L'approche région a pour objectif de décomposer l'image en un ensemble de régions connexes les plus homogènes possible. Plusieurs méthodes appartiennent à cette catégorie, on peut citer la segmentation par: division-fusion de régions, croissance de régions, classification de pixels. Cette dernière est l'approche qui nous intéresse dans ce travail. Elle consiste à classer les pixels en un certain nombre de classes en fonction de leurs attributs (couleurs). Chaque pixel est affecté à une classe unique, mais il est classifié indépendamment de ces voisins. Les critères employés sont en effet, des distances entre vecteurs d'attributs et ne tiennent pas compte, en principe, de la proximité entre les pixels dans l'image.

Duale à l'approche région, l'approche contour ne cherche plus à trouver les régions elles-mêmes mais leurs frontières. L'approche contour consiste à identifier les transitions entre les régions, qui peuvent se lier à une variation géométrique ou physique de la scène ou de l'objet observé. De nombreuses techniques d'extraction de contours existent dans la littérature, les plus utilisées sont les approches dérivatives (le *gradient* et le *laplacien*).

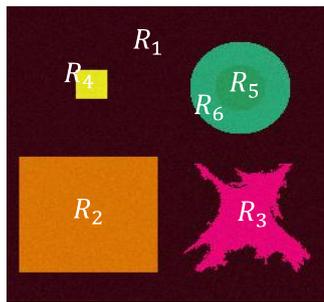
Les approches par coopération consistent à coopérer plusieurs méthodes, notamment celles de détection de contours et celles d'extraction de régions afin d'aboutir à un résultat satisfaisant et meilleur que celui obtenu en appliquant une seule méthode de segmentation. Trois approches sont souvent empruntées pour mettre en évidence l'aspect coopératif de la segmentation : l'approche séquentielle, l'approche parallèle et l'approche hybride.

IV.5.2 Représentation d'une image couleur

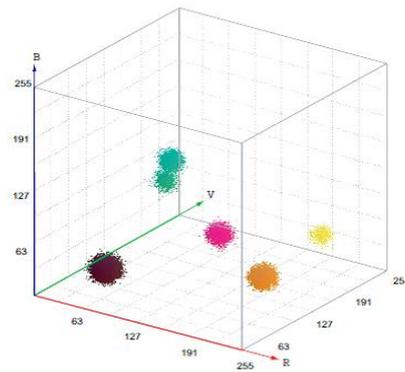
Une image couleur se définit comme une matrice $2D$ de pixels caractérisés par leurs composantes dans un espace couleur. L'espace le plus utilisé est l'espace R, V, B (R, G, B en anglais) dans lequel les coordonnées d'un point-couleur sont les niveaux des composantes couleurs, rouge, verte et bleue du pixel correspondant.

Généralement, les méthodes de segmentation d'images couleurs par classification considèrent que les régions de l'image de couleurs homogènes constituent des nuages de points-couleur dans l'espace couleur, chaque nuage correspondant à une classe de pixels qui partagent des propriétés colorimétriques similaires. Les classes sont construites par une procédure d'identification des nuages effectuée par un algorithme de classification. Quand les classes sont construites, les pixels sont assignés à l'une d'elles par une règle de décision. Les étiquettes des régions sont ensuite affectées aux pixels connexes assignés aux mêmes classes afin de construire l'image segmentée.

À titre d'exemple, la figure (IV.17.a) montre une image couleur de synthèse, constituée de six régions R_i de formes et tailles différentes. La figure (IV.17.b) présente les points-couleur représentant les régions dans le plan couleur R, V, B . Les points-couleur représentatifs des régions R_1, R_2, R_3 et R_4 forment des nuages bien séparés, ce qui faciliterait leur identification. Par contre, les points-couleur provenant des régions R_5 et R_6 donnent naissance à deux nuages qui se chevauchent.



(a) Image originale

(b) Points couleurs représentant les pixels de l'image (IV.17.a) dans l'espace (R, V, B) Figure IV.17 Représentation de l'image de synthèse dans l'espace couleur R, V, B

IV.5.3 Segmentation d'images couleurs utilisant la méthode spectrale KASP

Afin de tester et d'évaluer les performances des algorithmes de classification spectrale décrits dans le chapitre précédent, nous proposons de les appliquer à la segmentation des images couleurs réelles et de synthèse. Les performances de la segmentation sont estimées à partir de l'analyse de l'image segmentée. Pour ce faire, nous nous sommes basés sur le critère de Borsotti (*annexe C*) pour quantifier les résultats obtenus de la segmentation. Plus la valeur de ce critère est faible et plus le résultat de la segmentation est satisfaisant. En vue d'une éventuelle comparaison, les tests vont s'effectuer avec les trois algorithmes suivants:

- Algorithme non normalisé
- Algorithme normalisé selon Shi et Malik
- Algorithme normalisé selon Ng, Jordan et Weiss

Pour chaque algorithme, nous appliquerons quatre types de graphes : graphe totalement connecté, graphe de voisinage ε , graphe des K plus proches voisins *standard* et *mutuel*. Les paramètres à varier sont : le paramètre de la fonction de similarité gaussienne σ , le paramètre ε du graphe de voisinage ε , le paramètre K du graphe des K -PPV ainsi le nombre \mathbb{p} de points représentatifs. Le nombre de classes k à construire doit être fixé à *priori*. Afin de démontrer l'intérêt de la méthode spectrale accélérée, nous proposons de comparer les résultats de segmentation obtenus avec ceux obtenus par la méthode de k -Means.

IV.5.3.1 Segmentation de l'image "synthétique"

La figure (IV.18) représentant une image de synthèse comprenant 256×256 pixels soit $n = 65536$ points, codée dans le système RVB, a été utilisée comme image test pour tester les différentes méthodes de classification spectrale. Elle est constituée de six régions de couleur et de forme différentes, parmi elles, deux régions sont choisies de couleurs très proches dans le but de montrer de façon précise les performances de la méthode de segmentation à discriminer ces deux nuances de couleur : vert et vert légèrement plus foncé. Les résultats de la segmentation diffèrent selon les paramètres intervenant dans chacune des méthodes.

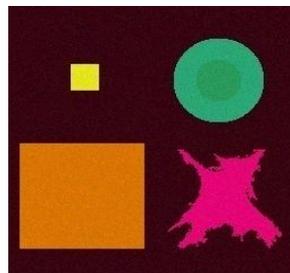


Figure IV.18 :L'image synthétique

1. Résultat de segmentation par la méthode de *k*-Means :

Le résultat de cette méthode dépend d'un seul paramètre à savoir le nombre de classes k fixé par l'utilisateur. Dans le cas de l'image synthétique, il est fixé à 6. La figure (IV.19) présente le résultat de segmentation par *k*-Means. Le critère de Borsotti a donné comme valeur : $Q = 472.4659$

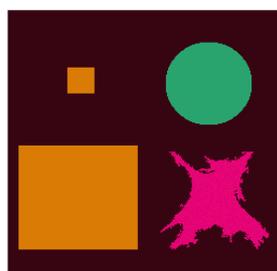


Figure IV.19 : Résultat obtenu de segmentation par la méthode de *k*-Means

2. Résultats de segmentation par la méthode de classification spectrale :

Les résultats obtenus par les différentes méthodes de classification spectrale accélérée pour un nombre de classes k égale à 6 sont visuellement identiques (figure (IV.20)), mais différent selon le critère de Borsotti comme le montre les tableaux (IV.10), (IV.11) et (IV.12).

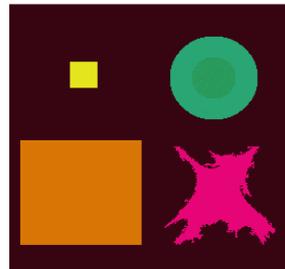


Figure IV.20 : Résultat obtenu de segmentation par les différents algorithmes de classification spectrale accélérée

Les valeurs du critère de Borsotti sont regroupées dans les tableaux ci-dessous :

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 150, \sigma = 0.38$	Q = 180.7747
Graphe de voisinage ε	$\mathbb{p} = 150, \varepsilon = 150, \sigma = 0.38$	Q = 158.8840
Graphe des <i>KPPV standard</i>	$\mathbb{p} = 150, K = 140, \sigma = 0.38$	Q = 140.4928
Graphe des <i>KPPV mutuel</i>	$\mathbb{p} = 150, K = 100, \sigma = 0.38$	Q = 142.7858

Tab IV.10 : Résultats obtenus par l’algorithme non normalisé

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 150, \sigma = 0.38$	Q = 175.2951
Graphe de voisinage ε	$\mathbb{p} = 150, \varepsilon = 150, \sigma = 0.38$	Q = 142.7877
Graphe des <i>KPPV standard</i>	$\mathbb{p} = 150, K = 140, \sigma = 0.38$	Q = 141.6390
Graphe des <i>KPPV mutuel</i>	$\mathbb{p} = 150, K = 100, \sigma = 0.38$	Q = 140.4911

Tab IV.11 : Résultats obtenus par l’algorithme normalisé selon SM

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 150, \sigma = 0.38$	Q = 151.5590
Graphe de voisinage ε	$\mathbb{p} = 150, \varepsilon = 150, \sigma = 0.38$	Q = 148.3297
Graphe des KPPV <i>standard</i>	$\mathbb{p} = 150, K = 140, \sigma = 0.38$	Q = 150.5008
Graphe des KPPV <i>mutuel</i>	$\mathbb{p} = 150, K = 100, \sigma = 0.38$	Q = 150.5156

Tab IV.12 : Résultats obtenus par l'algorithme normalisé selon NJW

3. Discussion :

Le résultat de segmentation obtenu par la méthode de k -Means (figure (IV.19)) nous montre que cette méthode ne parvient pas à séparer les deux régions représentant les deux disques concentriques. Ce même résultat nous montre aussi que les pixels constituant les deux carrés (jaune et orange) sont assignés à une seule classe des classes construites.

Comparé aux résultats des différents algorithmes de classification spectrale accélérée (figure (IV.20)), nous pouvons remarquer que les six régions de formes et de couleurs différentes de l'image originale ont été bien retrouvées. Ce résultat montre que cette méthode est capable de discriminer des classes de pixels dont les distributions des points-couleur se chevauchent fortement dans l'espace couleur (les deux disques).

Selon le critère de Borsotti (tableaux (IV.10), (IV.11) et (IV.12)), la segmentation par la méthode de classification spectrale accélérée est beaucoup meilleure que celle de la méthode de k -Means.

Les variations légères du critère de Borsotti pour les méthodes spectrales accélérées sont dues essentiellement au changement du type de graphe ainsi aux quelques erreurs d'affectation par la méthode du plus proche voisin.

D'une manière générale, vu que les résultats obtenus par les différentes méthodes spectrales accélérées sont visuellement très proches, et la ressemblance constatée entre l'image originale et celle reconstruite, ceci, nous autorise à dire que nous avons abouti à des résultats satisfaisants.

IV.5.3.2 Segmentation des images réelles

Afin de vérifier l'efficacité des algorithmes de classification spectrale, et de tester leurs robustesses, nous les appliquerons sur des images réelles beaucoup plus complexes. Les résultats de la segmentation diffèrent selon les paramètres intervenant dans chacune des méthodes.

IV.5.3.2.1 Segmentation de l'image de "fleurs"

La figure (IV.21) représente une image réelle de fleurs comprenant 150×150 pixels à savoir $n = 22500$ points, codée dans le système RVB a été utilisée comme image test réelle pour tester les méthodes de classification. Elle contient des fleurs de couleurs et de formes différentes, certaines présentent une variation graduelle de teinte. Le nombre de classes k doit être fixé à *a priori* par l'utilisateur, dans notre cas, il est égal à 8.



Figure IV.21 :L'image de fleurs

1. Résultat de segmentation par la méthode de k -Means :

La figure (IV.22) représente le résultat de segmentation par la méthode de k -Means pour un nombre de classes égale à 8. Le critère de Borsotti a donné comme résultat: $Q = 278.9290$



Figure IV.22 : Résultat de segmentation obtenu par la méthode de k -Means

2. Résultats de segmentation par la méthode de classification spectrale :

Les figures suivantes représentent les résultats de segmentation par les des différentes méthodes de classification spectrale accélérée:



Graphe totalement
connecté

Graphe de
voisinage ε

Graphe des K -PPV
standard

Graphe des K -PPV
mutuel

Figure IV.23 Résultats de segmentation obtenus par l'algorithme non normalisé



Graphe totalement
connecté

Graphe de
voisinage ε

Graphe des K -PPV
standard

Graphe des K -PPV
mutuel

Figure IV.24 Résultats de segmentation obtenus par l'algorithme normalisé selon SM



Graphe totalement
connecté

Graphe de
voisinage ε

Graphe des K -PPV
standard

Graphe des K -PPV
mutuel

Figure IV.25 Résultats de segmentation obtenus par l'algorithme normalisé selon NJW

Les valeurs du critère de Borsotti sont regroupées dans les tableaux ci-dessous :

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 150, \sigma = 3$	Q = 729.5417
Graphe de voisinage ε	$\mathbb{p} = 150, \varepsilon = 100, \sigma = 3$	Q = 255.2209
Graphe des K-PPV <i>standard</i>	$\mathbb{p} = 150, K = 70, \sigma = 3$	Q = 472.4595
Graphe des K-PPV <i>mutuel</i>	$\mathbb{p} = 150, K = 70, \sigma = 3$	Q = 299.7229

Tab IV.13 : Résultats obtenus par l'algorithme non normalisé

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 150, \sigma = 3$	Q = 336.3870
Graphe de voisinage ε	$\mathbb{p} = 150, \varepsilon = 100, \sigma = 3$	Q = 250.7577
Graphe des K-PPV <i>standard</i>	$\mathbb{p} = 150, K = 70, \sigma = 3$	Q = 281.3373
Graphe des K-PPV <i>mutuel</i>	$\mathbb{p} = 150, K = 70, \sigma = 3$	Q = 285.9627

Tab IV.14 : Résultats obtenus par l'algorithme normalisé selon SM

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 150, \sigma = 3$	Q = 272.7697
Graphe de voisinage ε	$\mathbb{p} = 150, \varepsilon = 100, \sigma = 3$	Q = 225.6221
Graphe des K-PPV <i>standard</i>	$\mathbb{p} = 150, K = 70, \sigma = 3$	Q = 266.5034
Graphe des K-PPV <i>mutuel</i>	$\mathbb{p} = 150, K = 70, \sigma = 3$	Q = 268.9943

Tab IV.15 : Résultats obtenus par l'algorithme normalisé selon NJW

3. Discussion :

Selon les résultats obtenus par l'image de fleurs, nous pouvons remarquer que les régions détectées sont relativement bien homogènes, malgré la complexité de cette image en matière de nombre de couleurs, de leurs rapprochements et de leurs finesses. Comparées à l'image originale, les formes des fleurs sont bien décrites et les détails du lys central sont distingués notamment lorsque nous utilisons le graphe de voisinage ε .

Le critère d'évaluation de Borsotti (tableaux (IV.13), (IV.14) et (IV.15)) nous montre que les résultats de segmentation sont proches excepté le cas de l'algorithme non normalisé utilisant graphe totalement connecté qui donne un résultat moins satisfaisant.

IV.5.3.2 Segmentation de l'image "house"

La figure IV.26 représente une image classique intitulée "house" comprenant 255×192 pixels soit $n = 48960$ points, codée dans le système couleur RVB. La segmentation de cette image présente plusieurs difficultés car l'image présente des effets d'ombres sous les toits de la maison et sur les arbres qui l'entourent. Par ailleurs, les régions qui constituent les murs ou les toits présentent des couleurs non homogènes. Le nombre de classes k à construire doit être fixé à *priori* par l'utilisateur.



Figure IV.26 :L'image "house"

1. Résultat de segmentation par la méthode de k -Means :

La figure (IV.27) représente le résultat de segmentation par la méthode de k -Means pour un nombre de classes k égale à 7. Le critère de Borsotti a donné comme valeur : $Q = 457.5207$



Figure IV.27 : Résultat obtenu de segmentation par la méthode de k -Means

2. Résultat de segmentation par la méthode de classification spectrale :

Les figures suivantes représentent les résultats de segmentation obtenus par les différentes méthodes de classification spectrale accélérée :

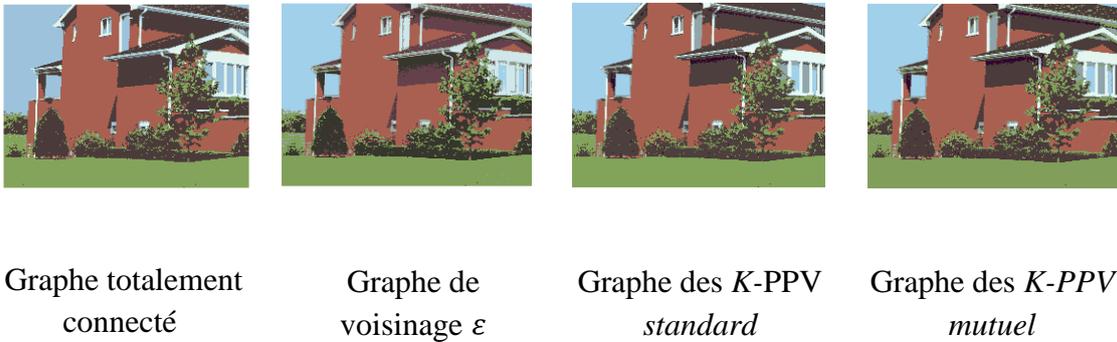


Figure IV.28 Résultats de segmentation obtenus par l’algorithme non normalisé

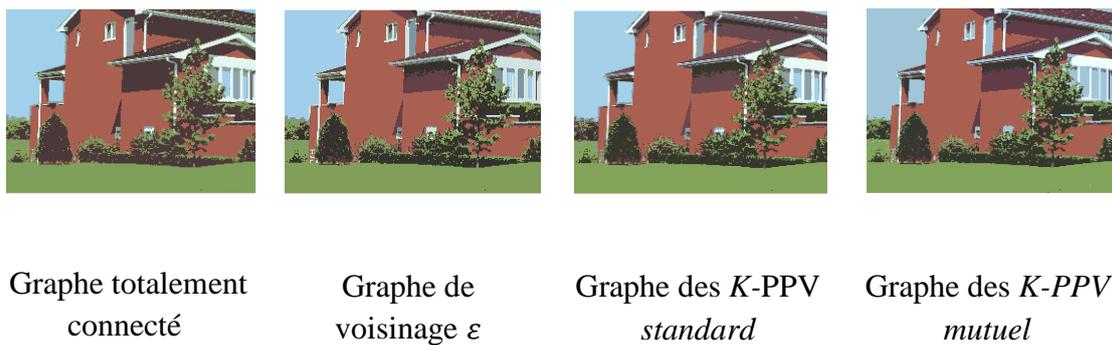


Figure IV.29 Résultats de segmentation obtenus par l’algorithme normalisé selon SM

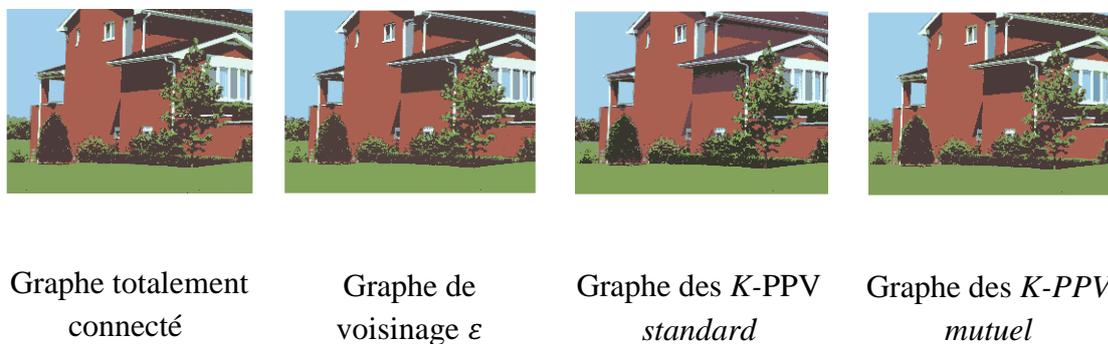


Figure IV.30 Résultats de segmentation obtenus par l’algorithme normalisé selon NJW

Les valeurs du critère de Borsotti sont regroupées dans les tableaux ci-dessous :

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 350, \sigma = 0.7$	Q = 592.3166
Graphe de voisinage ε	$\mathbb{p} = 350, \varepsilon = 35, \sigma = 0.7$	Q = 540.0568
Graphe des K-PPV <i>standard</i>	$\mathbb{p} = 350, K = 50, \sigma = 0.7$	Q = 478.7744
Graphe des K-PPV <i>mutuel</i>	$\mathbb{p} = 350, K = 180, \sigma = 0.7$	Q = 461.2499

Tab IV.16 : Résultats obtenus par l'algorithme non normalisé

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 350, \sigma = 0.7$	Q = 513.7136
Graphe de voisinage ε	$\mathbb{p} = 350, \varepsilon = 35, \sigma = 0.7$	Q = 524.5655
Graphe des K-PPV <i>standard</i>	$\mathbb{p} = 350, K = 50, \sigma = 0.7$	Q = 482.7783
Graphe des K-PPV <i>mutuel</i>	$\mathbb{p} = 350, K = 180, \sigma = 0.7$	Q = 482.4638

Tab IV.17 : Résultats obtenus par l'algorithme normalisé selon SM

	Paramètres	Critère de Borsotti Q
Graphe totalement connecté	$\mathbb{p} = 350, \sigma = 0.7$	Q = 509.8543
Graphe de voisinage ε	$\mathbb{p} = 350, \varepsilon = 35, \sigma = 0.7$	Q = 504.0820
Graphe des K-PPV <i>standard</i>	$\mathbb{p} = 350, K = 50, \sigma = 0.7$	Q = 519.3671
Graphe des K-PPV <i>mutuel</i>	$\mathbb{p} = 350, K = 180, \sigma = 0.7$	Q = 510.2738

Tab IV.18 : Résultats obtenus par l'algorithme normalisé selon NJW

3. Discussion :

Nous pouvons remarquer que les pixels constituant les différentes régions homogènes de l'image ont été bien affectés. En effet, les pixels représentant le mur en brique sont assignés à la même classe, tout comme le mur situé sur le côté de la maison. Ainsi, les pixels représentant le gazon sont bien assignés à une seule classe.

Par rapport au résultat de k -Means, les méthodes spectrales sont meilleures d'un point de vue visuel. Nous constatons que la méthode de k -Means n'a pas pu discriminer le ciel en bleu des bouts des toits contrairement aux méthodes spectrales accélérées.

La segmentation obtenue en utilisant le graphe totalement connecté nous montre que ce dernier n'arrive pas à distinguer les deux nuances de l'ombre au dessous des toits. Cependant, les meilleurs résultats sont obtenus avec les méthodes normalisées en utilisant le graphe des *KPPV standard*.

IV.6 Conclusion

Dans ce chapitre, nous avons présenté les différents résultats de classification obtenus par les différents algorithmes de classification spectrale décrits dans le chapitre précédent. Dans un premier temps, nous les avons appliqués sur des données artificielles, puis à la segmentation des images couleurs réelles et de synthèse représentées dans l'espace couleur RVB.

Pour mieux situer la méthode spectrale, des tests ont été effectués dans le but de comparer ses performances avec d'autres méthodes. La méthode que nous avons choisie est celle de k -Means. Cette comparaison est réalisée via la matrice de confusion pour le cas des données artificielles et via le critère d'évaluation de Borsotti pour la segmentation des images couleurs.

Les résultats expérimentaux ont montrés que la méthode de classification spectrale ne suppose aucune forme sur les classes et reste performante et efficace même lorsque les distributions des points présentent des chevauchements dans l'espace de représentation, contrairement à la méthode de k -Means qui n'arrive pas à les discriminer.

Toutefois, nous avons montré que cette méthode souffre de quelques inconvénients tels que le problème de stockage mémoire et de temps de calcul dans le cas de données de grande taille. Pour surpasser ce problème, surtout dans le cas de la segmentation d'images couleurs, nous avons proposé d'utiliser une méthode spectrale approximative (KASP) plus rapide que la méthode spectrale classique. Cependant, la méthode KASP nécessite plusieurs essais pour obtenir une bonne solution, car, elle reste sensible aux choix des paramètres σ , ε , K , ainsi qu'à l'initialisation des points représentatifs \mathbb{p} dû essentiellement à la méthode de k -Means et à l'affectation par la méthode du plus proche voisin.

Conclusion générale

Conclusion générale

Dans ce travail, nous avons étudié les éléments des méthodes de classification de données et plus particulièrement la méthode de classification spectrale (spectral clustering). L'objectif était de comprendre le fonctionnement de cette méthode afin de l'appliquer dans un cadre non supervisé à la segmentation des images couleurs. Cette méthode a été formulée à l'aide de la théorie spectrale des graphes et des coupes de graphes. Elle permet de projeter implicitement les données dans un espace de faible dimension dans lequel les données seront linéairement séparables. Elle comprend essentiellement trois étapes: (1) *prétraitement* : à partir de la base de données, construire la matrice de similarité, (2) *décomposition* : analyser les valeurs et vecteurs propres (spectre) de la matrice de similarité et (3) *groupement* : obtenir une classification à partir des vecteurs propres dominants.

La deuxième étape laisse apparaître plusieurs manières de construire les matrices de similarité selon le type du graphe utilisé (graphe totalement connecté, graphe de voisinage ϵ , graphe des *KPPV standard* et *KPPV mutuel*) et selon le type de la matrice laplacienne considérée (non normalisée, normalisée selon Shi et Malik et normalisée selon Ng, Jordan et Weiss). Il en découle 12 variantes de la méthode de classification spectrale.

Nous avons implémenté et testé les 12 méthodes de classification spectrale. Les résultats obtenus ont permis de démontrer que la méthode spectrale est capable de distinguer les classes d'objets présentant des chevauchements dans l'espace de représentation. En effet, nous avons pu constater que le succès de ces algorithmes est fondé sur le fait qu'ils ne font aucune supposition sur la forme des clusters et permettent de séparer non-linéairement des classes, contrairement aux autres méthodes comme la méthode *k*-Means qui permet de retrouver seulement les classes globulaires linéairement séparables.

Toutefois, nous avons montré que cette méthode souffre de quelques inconvénients tels que l'ajustement du paramètre de la fonction de similarité gaussienne σ et les paramètres ϵ et K qui définissent la construction du graphe, ainsi que le choix à *priori* du nombre de classe k .

Dans le cadre de la segmentation d'images couleurs, nous avons affaire à un grand flot de données. Or, les étapes de calcul de la matrice de similarité et d'extraction des vecteurs propres deviennent très coûteuses en termes de temps de calcul et en stockage mémoire.

Afin de surpasser ces problèmes, plusieurs travaux ont été développés principalement basés sur des techniques d'algèbre linéaire ou bien sur des techniques de réduction de la taille des données pour réduire le coût numérique. Dans notre application, nous avons fait appel à la méthode de classification spectrale accélérée KASP qui consiste à réduire la taille des ensembles de données par une sélection d'un certain nombre d'individus en utilisant la méthode de k -Means avec un nombre de classes élevé. Cette méthode nécessite une étape supplémentaire pour affecter les données originales aux classes détectées en utilisant la règle du plus proche voisin. Les résultats obtenus sont en général très satisfaisants. Cependant, les résultats de segmentation obtenus par quelques graphes ont montrés que certaines classes ne sont pas correctement reconnues ou mal construites, cela est dû essentiellement à une mauvaise initialisation des centres des classes ainsi qu'à la règle d'affectation du plus proche voisin.

Références bibliographiques

Références bibliographiques

- [1] **Stéphane TUFFERY.**, *“Data mining et statistique décisionnelle, l’intelligence des données”*, édition TECHNIP, ISBN 13 : 978-2-7108-0888-6.
- [2] **Hédia Mhiri Sellami, Ali Jaoua.**, *“Une méthode de classification de données binaires basée sur la décomposition rectangulaire ”*, la revue du MODULAD, décembre 2000.
- [3] **Emmanuel ZENOU.**, *“Localisation topologique, amers visuels et treillis de gallois”*, thèse de doctorat, Ecole Nationale Supérieure de l’Aéronautique et de l’Espace, décembre, 2004.
- [4] **Nicolas TURENNE.**, *“Apprentissage statistique pour l’extraction de concepts à partir de textes. Application au filtrage d’information textuelles”*, thèse de doctorat, Université LOUIS PASTEUR de Strasbourg, novembre 2000.
- [5] **Amel Boulemnadjel.**, *“Partitionnement neuronal et validité des classes, Application à la segmentation d’images”*, thèse de Magister en électronique, Université Mentouri de Constantine, 2009.
- [6] **Kohonen, T.**, *“Self-organized and Associative Memory Analysis and Machine intelligence”*. Springer-Verlag, 1990.
- [7] **I. RIVALS, L. PERSONNAZ, G. DREYFUS.**, *“Modélisation, classification et commande par réseaux de neurones : principes fondamentaux, méthodologie de conception et illustrations industrielles”*, Ecole Supérieure de Physique et de Chimie Industrielles de la Ville de Paris, Lavoisier Technique et Documentation, Paris.
- [8] **Jean-Pierre Nakache , Josiane Confais.**, *“Approche pragmatique de la classification, Arbres hiérarchiques, Partitionnements”*, édition TECHNIP.

- [9] **François-Xavier Jollois.**, "*Contribution de la classification automatique à la Fouille de Données* ", thèse de doctorat, Université de Metz, décembre 2003.
- [10] **Michaël FONTAINE.**, "*Segmentation non supervisée d'images couleur par analyse de la connexité des pixels*", thèse de doctorat en Automatique et Informatique Industrielle, Université de Lille I, décembre 2001.
- [11] **Lotfi KHODJA.**, "*Contribution à la Classification Floue non Supervisée*", thèse de doctorat, Université de Savoie.
- [12] **David FREZZATO.**, "*Classification de fonds de futures*", rapport de stage, 31 mai 2001.
- [13] **Belacel Nabil.**, "*Méthodes de Classification Multicritère Méthodologie et Applications à l'Aide au Diagnostic Médical*", thèse de doctorat en Sciences orientation Recherche Opérationnelle, Université libre de Bruxelles, 2000.
- [14] **Michel Grabisch.**, "*Classification*", december 14, 2000.
- [15] **J-C. Bazdek.**, "*Pattern recognition with fuzzy objective function algorithms*", Plenum Press, New York, 1981.
- [16] **J-C. Dunn.**, "*A fuzzy relative of the isodata process and its use in detecting compact well-seperated clusters*", J.Cybernetics, pages 32-57, 1973.
- [17] **A. Dempster, A. P., Laird, N. M. et Rubin, D. B.**, "*Maximum likelihood from incomplete data via the EM algorithm* ",Journal of the Royal Statistical Society, Series B, 34:1-38, 1977.
- [18] **Celeux, G. et Diebolt, D.**, "*The SEM algorithm : a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem*",*Computational Statistics Quarterly*, 2(1) :73–82, 1985.
- [19] **Celeux, G. et Govaert, G.**, "*A classification EM algorithm for clustering and two stochastic versions*", Rapport de recherche RR-1364, Inria, Institut National de Recherche en Informatique et en Automatique, 1991.

- [20] **Gilles Celeux, Jean Diebolt, Jean-Claude Biscarat.**, “*Stochastic versions of the EM algorithm*“, technical report No. 227, January 1992.
- [21] **Wei, G. C. G. et Tanner, M. A.**, “*A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms*“, *Journal of the American Statistical Association* 85:699-704, 1990.
- [22] **THEORIE DES GRAPHERS .**, «www.wikipedia.com ».
- [23] **N. BELHARRAT .**, “*Théorie des Graphes en Recherche opérationnelle*“, éditions, pages bleues, 2009.
- [24] **W.L. PRICE.**, traduction en français par H.GRUNSPAN., “*Introduction aux graphes et aux réseaux*“.
- [25] **Michel GONDRAN, Michel MINOUX.**, “*Graphes et algorithmes*“, 3eme édition, édition EYROLLES, 1995.
- [26] **Ulrike von luxburg .**, “*A tutorial on spectral clustering*“. Max Planck Institute for Biological Cybernetics, Statistics and Computing, Germany, 2007.
- [27] **Classification spectrale (intelligence artificielle).**, “www.wikipedia.com”.
- [28] **Zelnik-Manor L., Perona P.**, “*Self-tuning spectral clustering*“. *Advances in Neural Information Processing Systems*, vol. 17, p. 1601-1608, 2004.
- [29] **Salton G.**, “*Automatic Text Processing: the transformation, analysis and retrieval of information by computer*“. Addison- Wesley, 1989.
- [30] **Chung, F.**, “*Spectral graph theory*” (Vol. 92 of the CBMS Regional Conference Series in Mathematics). Conference Board of the Mathematical Sciences, Washington, 1997.
- [31] **Mohar, B.**, “*The Laplacian spectrum of graphs*“. In *Graph theory, combinatorics, and applications*. Vol. 2 (Kalamazoo, MI, 1988) (pp. 871 – 898). New York: Wiley, 1991.

- [32] **Mohar, B.**, “*Some applications of Laplace eigenvalues of graphs*”. In G. Hahn and G. Sabidussi (Eds.), *Graph Symmetry: Algebraic Methods and Applications* (Vol. NATO ASI Ser. C 497, pp. 225 – 275). Kluwer, 1997.
- [33] **Deerwester S., Dumais S., Landauer T., Furnas G., Harshman R.**, “*Indexing by latent semantic analysis*”. *Journal of the American Society of Information Science*, vol. 41(6), p. 391-407, 1990.
- [34] **Stoer, M. and Wagner, F.**, “*A simple min-cut algorithm*”. *J. ACM*, 44 (4), 585 – 591, 1997.
- [35] **Shi, J. and Malik, J.,.** “*Normalized cuts and image segmentation*”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (8), 888 – 905, 2000.
- [36] **Ng, A., Jordan, M., and Weiss, Y.**, “*On spectral clustering: analysis and an algorithm*”. In T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14* (pp. 849 – 856). MIT Press, 2002.
- [37] **Hagen, L. and Kahng, A.**, “*New spectral methods for ratio cut partitioning and clustering*”. *IEEE Trans. Computer-Aided Design*, 11 (9), 1074 – 1085, 1992.
- [38] **Wagner, D. and Wagner, F.**, “*Between min cut and graph bisection*”. In *Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science (MFCS)* (pp. 744 – 750). London: Springer, 1993.
- [39] **Lutkepohl, H.**, “*Handbook of Matrices*”. Chichester: Wiley, 1997.
- [40] **Meila M., Shi J.**, “*Learning segmentation by random walks*”. *Advances in Neural Information Processing Systems*, p.470-477, 2000.
- [41] **Kamvar S., Klein D., Manning C.**, “*Spectral Learning*”. *18th International Joint Conference on Artificial Intelligence*, p. 561-566, 2003.
- [42] **Freeman W.T. Perona, P.**, “*A factorization approach to grouping*”. *European Conference on Computer Vision*, 1998.

- [43] **M. Brand and K. Huang,** “*A unifying theorem for spectral embedding and clustering*”. 9th International Conference on Artificial Intelligence and Statistics, 2002.
- [44] **L. Zelnik-Manor and P. Perona,** “*Self-tuning spectral clustering*”. Advances in Neural Information Processing Systems, 17(1601-1608) :16, 2004.
- [45] **I. Fischer and J. Poland,** “*Amplifying the block matrix structure for spectral clustering*”. In Proceedings of the 14th Annual Machine Learning Conference of Belgium and the Netherlands, pages 21–28. Citeseer, 2005.
- [46] **Brito, M., Chavez, E., Quiroz, A., and Yukich, J.,** “*Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection*”. Statistics and Probability Letters, 35, 33 – 42, 1997.
- [47] **Penrose, M.,** “*A strong law for the longest edge of the minimal spanning tree*”. Ann. of Prob., 27 (1), 246 – 260, 1999.
- [48] **Algorithme Espérance-Maximisation** ., «www.wikipedia.com ».
- [49] **Aarti Singh,** “*Spectral Clustering*”. Machine Learning 10-701/15-781 Nov 22, 2010.
- [50] **Sylvie Philipp-Foliguet, Laurent Guigues,** “*Evaluation de la segmentation d’images : état de l’art, nouveaux indices et comparaison*”. Rapport Technique, 2001.
- [51] **M. Borsotti, P. Campadelli, R. Schettini,** “*Quantitative evaluation of color image segmentation results*”. Pattern Recognition Letters, Milano, Italy, 1998.
- [52] **Donghui Yan, Ling Huang, Michael I. Jordan,** “*Fast Approximate Spectral Clustering*”. Technical Report, University of California.
- [53] **C. Fowlkes, S. Belongie, F. Chung, and J. Malik,** “*Spectral grouping using the Nystrom method*”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(2) :214–225, 2004.

- [54] **Wen-Yen Chen et Song Yangqiu et Hongjie Bai et Chih-Jen Lin et Edward Y. Chang.,** “*Parallel Spectral Clustering in Distributed Systems*”. Preprint of IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.
- [55] **Y. Song, W.Y. Chen, H. Bai, C.J. Lin, and E.Y. Chang.,** “*Parallel spectral clustering*”. In Proceedings of European Conference on Machine Learning and Pattern Knowledge Discovery. Springer, 2008.
- [56] **S. OUATARA.,** “*Stratégies de segmentation d’images multi-composantes par analyse d’histogramme multidimensionnels*”. Thèse de doctorat, école doctorale d’Angers STIM, France, 2009.
- [57] **M. Lezoray Olivier.,** “*Segmentation d’images couleurs par morphologie mathématique et classification de données par réseaux de neurones : application à la classification de cellules en cytologie des séreuses*”. Thèse de doctorat, université de CAEN/BASSE-NORMANDIE, France, 2000.

Annexes

Annexe A : Théorème de Rayleigh-Ritz

Soit $A, B \in \mathbb{C}^{n \times n}$ deux matrices et $R: \mathbb{C}^n \setminus \{0\} \rightarrow \mathbb{R}$ tel que :

$$R(x) = \frac{x^T A x}{x^T B x}, \quad \text{avec } \|x\| \neq 0$$

Cette expression est appelée le quotient de Rayleigh, il définit les vecteurs propres comme des points critiques dans \mathbb{R} . Alors :

$$\left\{ \begin{array}{l} \lambda_{max}(A) = \max_{x^T x=1} (x^T A x) \\ \lambda_{min}(A) = \min_{x^T x=1} (x^T A x) \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{l} \lambda_{max}(AB^{-1}) = \max_{\|x\| \neq 0} R(x) \\ \lambda_{min}(AB^{-1}) = \min_{\|x\| \neq 0} R(x) \end{array} \right.$$

Annexe B : Evaluation des résultats de la classification des données artificielles

Afin d'évaluer les résultats d'une méthode de classification, on calcule les indices suivants :

- **Matrice de confusion MC:** une matrice de confusion sert à évaluer la qualité d'une classification (taux d'erreur et taux de classification). Elle est obtenue en comparant les données classées avec des données de référence. Elle suppose donc que les classes sont connues à *priori*.
- **Taux de classification T_c :** c'est le nombre d'individus bien classés divisé par le nombre total d'individus (en pourcentage). Il se calcul à partir de la matrice de confusion en divisant sa trace sur la somme de tous ses éléments (tous les individus).

Annexe C : Critère d'évaluation des résultats de segmentation sur une image couleurs

Comme il est difficile d'apprécier visuellement la qualité de segmentation d'une image, et face à ce constat, la comparaison entre les différentes segmentations devient importante afin de choisir une stratégie appropriée. La qualité de la segmentation est une question récurrente en analyse d'images, pour valider une nouvelle méthode ou pour mesurer la pertinence des résultats d'une chaîne de traitement.

Les principaux critères que nous retrouvons le plus souvent dans la littérature sont :

- Critère de Liu et Yang
- Critère de Borsotti
- Contraste de ZEBOUJ
- Critère d'uniformité intra-région de LEVINE ET NAZIF
- Contraste inter-région de LEVINE ET NAZIF

Les différents critères sont décrits de façon détaillée dans [50].

Critère de Borsotti :

Afin d'évaluer les résultats de différentes segmentations d'une image couleur, M. Borsotti, P. Campadelli et R. Schettini [51] proposent un critère qui ne nécessite aucun réglage de paramètres et qui correspond de manière assez proche au jugement visuel, ce critère est défini par l'équation suivante:

$$Q(I) = \frac{1}{10000 \times (N \times M)} \sqrt{R} \times \sum_{i=1}^R \left(\frac{e_i^2}{1 + \log A_i} + \frac{R(A_i)^2}{A_i} \right)$$

Où :

- $N \times M$ est la taille de l'image.
- R est le nombre de régions reconstruites dans l'image segmentée.
- A_i est la surface de la i -ème région reconstruite. En pratique, c'est le nombre de pixels qui constituent la i -ème région.
- e_i est la somme des distances euclidiennes entre la couleur (R, G, B) de chaque pixel de la i -ème région reconstruite et la couleur moyenne de cette même région reconstruite.
- $R(A_i)$ est le nombre de régions reconstruites dont la surface est égale à A_i .

En résumé, plus la valeur de $Q(I)$ est faible, plus le résultat de la segmentation est satisfaisant.

Annexe D : Les métriques en classification

Les métriques permettent de mesurer les distances séparant deux positions d'un espace multidimensionnel. Une métrique est une fonction qui décrit la distance entre deux points d'un ensemble E . Cette distance est une application de $E \times E$ dans R^+ telle que, $\forall i, j, k \in E$ [56]:

- $d(x_i, x_j) = d(x_j, x_i)$ (symétrie)
- $d(x_i, x_j) \geq 0$
- $d(x_i, x_j) = 0 \Leftrightarrow x_i = x_j$
- $d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j)$ (inégalité triangulaire)

De nombreuses distances présentées dans la littérature vérifiant ces propriétés. Mais aucune ne peut être considérée comme optimale pour opérer une classification. Tout dépend des données à traiter et des algorithmes de classification mis en œuvre. En classification, les métriques les plus utilisés sont la métrique euclidienne et la métrique de Mahalanobis [57].

Soient deux positions x et y de l'espace nD , repérées respectivement par les coordonnées (x_1, \dots, x_n) et (y_1, \dots, y_n) . La distance entre ces positions peut s'exprimer de différentes façons, dont les plus courantes sont indiquées ci-dessous.

1. Distance de Minkowsky :

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^P \right)^{\frac{1}{P}}$$

2. Distance de Manhattan :

Dans le cas particulier où $P = 1$, la formule de *Minkowsky* fournit la distance de Manhattan :

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

3. Distance Euclidienne :

Dans le cas particulier où $P = 2$, la formule de *Minkowsky* fournit la distance Euclidienne :

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^2 \right)^{\frac{1}{2}}$$

4. *Distance de Chebychev :*

Lorsque P tend vers l'infini, la formule de **Minkowsky** fournit la distance de Chebychev, encore appelée distance Queen-wise ou métrique maximum :

$$d(x, y) = \max_{i=1}^n |x_i - y_i|$$

5. *Distance de Mahalanobis :*

Elle mesure la distance entre une position x et y :

$$d(x, y) = \sqrt{(x - y)^T M^{-1} (x - y)}$$

Où M est la matrice de variance-covariance définie sur l'ensemble d'apprentissage. Si la matrice de variance-covariance est la matrice identité, cette distance est alors la même que la distance euclidienne.

6. *Distance des cordes carrées :*

$$d(x, y) = \sum_{i=1}^n (\sqrt{x_i} - \sqrt{y_i})^2$$

