Université de Tunis
Institut supérieur de gestion

Classification Recouvrante Basée sur les Méthodes de Noyaux

Mémoire en vue de l'obtention du Mastère
En informatique de gestion

Présenté par
Chiheb-Eddine Ben N'Cir

Sous la direction de

Dr. Nadia Essoussi

Session : Juin 2010

Remerciements

Je remercie tout spécialement Madame Nadia Essoussi pour son soutien constant, pour son aide inestimable, pour sa présence, pour la qualité de son encadrement et pour ses remarques pertinentes qui m'ont aidé à l'amélioration continue de ce mémoire.

Je tiens également à remercier Monsieur Mohamed limam pour m'avoir guidé vers une plus grande maturité et une meilleure vision sur le domaine de la recherche.

J'exprime ma profonde gratitude à Monsieur Patrice Bertrand, qui m'a présenté le domaine de recherche de la classification recouvrante et qui m'a guidé dans ce travail.

Je tiens à remercier Monsieur Ghazi Belmofti et Monsieur Riadh Khanchel qui ont contribué à l'avancée de ces recherches.

Je tiens aussi à exprimer ma reconnaissance à mes parents et mes frères pour leur soutien inconditionnel.

Enfin, mes remerciements vont également aux membres de jury d'avoir accepter de juger notre travail.

Table des matières

Introduction générale 1

Chapitre 1 : Etat de l'art de la classification automatique 3

Introduction 3
1.1. Classification automatique : le processus d'apprentissage non supervisé 3
1.2. Catégorisation des méthodes de classification 5
1.2.1. Méthodes de classification Stricte 5
1.2.1.1. Méthodes Stratifiées 6
1.2.1.2. Méthodes par partitionnement 7
1.2.2. Méthodes de classification floue 9
1.2.3. Méthodes de classification recouvrante 10
1.3. Les Problématiques dans la classification automatique 12
1.3.1. Notion de similarité 12
1.3.2. Détermination du nombre de classes 12
1.3.3. Identification des classes de formes complexes et de densités variées 13
1.3.4. Traitement des données éloignées 15
1.3.5. Evaluation et validation des résultats de la classification 15
1.4. Les méthodes de noyaux 15
1.4.1. La méthode « Kernel k-means » 17
1.4.2. SKK: Soft Kernel K-means 18
Conclusion 20

Chapitre 2 : Méthodes de classification recouvrante 21

Introduction 21
2.1. Méthodes recouvrantes basées sur la réallocation dynamique 21
2.1.1. La méthode « OKM : Overlapping K-Means » 22
2.1.2. La méthode « OKMED: Overlapping K-Medoid » 23
2.1.3. Synthèse des méthodes recouvrantes basées sur la réallocation dynamique.. 24
2.2. Méthodes basées sur les modèles de mélange de lois 26
2.2.1. La méthode « MOC: Model based Overlapping Clustering » 26
2.2.2. IOMM: infinite overlapping mixture model 29
2.2.3. Synthèse des méthodes recouvrantes basées sur les modèles de mélange de lois 31
2.3. Méthodes basées sur la théorie des graphes 32
Conclusion 32

Chapitre 3 : Estimation du nombre de recouvrements et identification des classes de formes complexes 33

Introduction 33
3.1. Estimation du nombre de classes 33
3.1.1. Premier cas : classes bien séparées et non recouvrantes 35
3.1.2. Deuxième cas : classes qui se chevauchent et de densité non variée 36
3.1.3. Troisième cas : clusters qui se chevauchent et qui sont concentriques 36
3.1.4. Quatrième cas : clusters de densité variée 37
3.2. Utilisation des Méthodes de noyaux comme une mesure de distance 38
3.3. Autocritique de la solution proposée 40
3.4. Solution proposée « KOKM : Kernel Overlapping K-Means » 41
3.4.1. Critère objectif de « KOKM » 41
3.4.2. Algorithme de classification de « KOKM » 42
3.4.3. Le calcul des centres des recouvrements dans « KOKM » 43
3.4.4. L'affectation des objets à un ou plusieurs clusters 45
3.4.5. Exemple illustratif d'exécution de « KOKM » 48
Conclusion 50

Chapitre 4 : Evaluation empirique des solutions proposées 51

Introduction 51
4.1. Méthode et Mesures de validation utilisées 51
4.1.1. Distances utilisées dans « OKM » 52
4.1.2. Méthode de validation 52
4.1.3. Environnement de travail 54
4.2. Expérimentations sur le dataset « Iris » 54
4.2.1. Détermination du nombre de classes du dataset « Iris » 56
4.2.2. Comparaison entre « OKM » et « KOKM » sur le dataset « Iris » 56
4.3. Expérimentations sur le dataset « Ionosphère » 58
4.3.1. Détermination du nombre de classes du dataset « Ionosphère » 59
4.3.2. Comparaison entre « OKM » et « KOKM » sur le dataset « Ionosphère » 60
4.4. Expérimentations sur le dataset « Eachmovie » 60
4.4.1. Détermination du nombre de classes dans le dataset « Eachmovie » 62
4.4.2. Comparaison entre « OKM » et « KOKM » sur le dataset « Eachmovie » 63
Conclusion 64

Conclusion générale 65

Références Bibliographiques **67**
Annexe 1 : Algorithme de la méthode « OKM » **72**
Annexe 2 : Algorithme de la méthode « OKMED » **75**
Annexe 3 : Algorithme de la méthode « Kernel k-means » **78**
Annexe 4 : Algorithme de la méthode « SKK » **79**

Tables des figures

Chapitre 1 : Etat de l'art de la classification automatique
Figure 1. Les étapes du processus de classification automatique 4
Figure 2. Exemple de partitions formées par les méthodes de classification stricte. 6
Figure 3. Dendrogramme des partitions emboîtées d'un ensemble de 6 objets. 7
Figure 4. Mise à jour des centres des classes dans la méthode k-moyennes. 8
Figure 5. Exemple de groupes formés par les méthodes de classification floue 10
Figure 6. Exemple de recouvrements formés par les méthodes de classification recouvrante 11
Figure 7. Estimation du nombre de clusters par densité 13
Figure 8. Clusters de formes complexes : (a) clusters sous formes de chaine (b) clusters de formes concentriques 14
Figure 9. Clusters de densités variées 14
Figure 10.Clusters de formes concentriques et de densités variées 14
Figure 11. Clusters non linéairement séparables projetés sur une troisième dimension où ils deviennent séparables [Schölkopf and Smola 2002] 16

Chapitre 2 : Méthodes de classification recouvrante
Figure 12. Problématiques de recouvrements concentriques dans la méthode « OKM » 25
Figure 13. Instanciation du modèle « PRM » avec 2 gènes (données), 2 conditions (dimensions), 3 processus (classes). 27
Figure 14. Le produit de deux distributions gaussiennes (en bleu) de paramètre N_1 (-1, 0.2) et N_2 (1,0.6) est une distribution gaussienne (en rouge). 30

Chapitre 3 : Estimation du nombre de recouvrements et identification des classes de formes complexes
Figure 15. Exemple de matrice de noyaux *K(6,6)* construite à partir d'un ensemble de données (6 objets) de dimension 3 (3 attributs) avec un noyau RBF de paramètre $\sigma = 2$ 34
Figure 16. Méthode de détermination du nombre de recouvrements 35
Figure 17. Classes bien séparées et non recouvrantes 36
Figure 18. Classes recouvrantes de densité non variée 36
Figure 19. Classes recouvrantes concentriques 37
Figure 20. Classe recouvrante de densité variée 37
Figure 21. Distance entre deux représentations d'objets dans l'espace de redescription 38
Figure 22. Algorithme de la fonction « *K*_ASSIGN » utilisant les distances basées sur les méthodes de noyaux 40
Figure 23. Algorithme principal de la méthode « KOKM ». 43
Figure 24. Algorithme de la fonction « *AFFECT* » 47
Figure 25. Distribution des objets test sur les deux dimensions définies 48

Chapitre 4 : Evaluation empirique des solutions proposées
Figure 26. Illustration de la mesure du rappel et de la mesure de précision 52
Figure 27. Représentation du dataset « Iris » sur les 2 premiers axes en utilisant la méthode PCA 55
Figure 28. Représentation du dataset « Iris » sur les 2 premiers axes avec un noyau RBF $\sigma = 2$ en utilisant la méthode PCA 55

Figure 29 : Diagramme en bâton des valeurs propres les plus significatives dans le dataset « Iris » 56
Figure 30. Représentation de la base « Ionosphère » sur les deux premiers axes en utilisant la méthode PCA 58
Figure 31. Représentation du dataset Ionosphère sur les deux premiers axes avec un noyau RBF σ = 20 en utilisant la méthode PCA 59
Figure 32. Diagramme en bâton des valeurs propres les plus significatives dans le dataset « Ionosphère » en utilisant un noyau RBF de paramètre σ = 20 59
Figure 33. Représentation d'un sous ensemble du dataset « Eachmovie » contenant 3 genres de films sur les deux premiers axes en utilisant la méthode PCA 61
Figure 34. Représentation d'un sous ensemble de la base « Eachmovie » contenant 3 genres de films avec un noyau RBF σ = 20 sur les deux premiers axes en utilisant la méthode PCA 62
Figure 35. Diagramme en bâton des valeurs propres les plus significatives d'un sous ensemble du dataset « Eachmovie » contenant 3 genres de films en utilisant un noyau RBF de paramètre σ = 20 62

Annexe 1 : Algorithme de la méthode « OKM »

Figure 36. Algorithme principal de la méthode « OKM » 73
Figure 37. Description de la fonction « Prototype » utilisée dans la méthode « OKM » 73
Figure 38. Description de la fonction « ASSIGN » utilisée dans la méthode « OKM » 74

Annexe 2 : Algorithme de la méthode « OKMED »

Figure 39. Algorithme principal de la méthode « OKMED » 76
Figure 40. Description de la fonction « *MEDOID* » utilisée dans la méthode « OKMED »... 76
Figure 41. Algorithme de la fonction « ASSIGN » utilisée dans « OKMED » 77

Annexe 3 : Algorithme de la méthode « Kernel k-means »

Figure 42. Algorithme principal de la méthode « kernel k-means » 78

Annexe 4 : Algorithme de la méthode « SKK »

Figure 43. Algorithme principal de la méthode « SKK » 80

Liste des tableaux

Chapitre 1 : Etat de l'art de la classification automatique

Tableau 1. La distribution des données entre les 3 partitions A, B et C……5

Tableau 2. La distribution des données entre les 3 partitions A, B et C par une méthode de classification floue…..……9

Tableau 3. La distribution des données entre les 3 recouvrements A, B et C par une méthode de classification recouvrante…….11

Chapitre 4 : Evaluation empirique des solutions proposées

Tableau 4. Comparaison entre la méthode « OKM » et la méthode « KOKM » avec même importance de la précision et du rappel sur le dataset « Iris »……..57

Tableau 5. Comparaison entre la méthode « OKM » et la méthode « KOKM » en donnant une double importance de la précision par rapport au rappel sur le dataset « Iris »…….57

Tableau 6. Comparaison entre la méthode « OKM » et la méthode « KOKM » avec même importance de la précision et du rappel sur le dataset « Ionosphère »……..60

Tableau 7. Comparaison entre la méthode « OKM » et la méthode « KOKM » avec même importance de la précision et du rappel sur le dataset « Eachmovie »……..63

Tableau 8. Comparaison entre la méthode « OKM » et la méthode « KOKM » en donnant une double importance de la précision par rapport au rappel sur le dataset « Eachmovie »…64

Introduction générale

.

La classification automatique désigne le groupement des objets dans une ou plusieurs partitions de telle manière que les objets qui sont similaires appartiennent à la même partition alors que les objets dissimilaires appartiennent à des partitions différentes. Cette définition pourrait être non applicable dans certains problèmes réels qui nécessite qu'un objet puisse appartenir à la fois à plusieurs partitions. Par exemple, en biologie, un gène peut participer à plusieurs processus ; en recherche d'information, un document peut aborder plusieurs thématiques ou appartenir à plusieurs genres différents ; en traitement du langage, un mot peut avoir plusieurs interprétations ; en réseaux sociaux, un individu peut appartenir à plusieurs communautés. A cet effet, une tendance dans la réalisation des algorithmes de clustering appelée classification recouvrante (overlapping clustering) a essayé de résoudre ces problèmes qui se sont posés naturellement.

Les approches traditionnelles proposées dans la littérature étendent les schémas de classification floue où un objet appartient à plusieurs classes avec des différents degrés d'appartenance. En fixant un seuil minimal sur ces degrés, les objets sont affectés à une ou à plusieurs classes. Ces approches ne permettent pas de traiter tous les schémas de recouvrement possibles. A cet effet, les nouvelles approches dans la classification recouvrante cherchent directement des recouvrements optimaux et non pas des partitions optimales. De nombreuses approches ont été proposées. L'éventail de ces approches comprend à minima des généralisations des méthodes de réallocation dynamique tel que « K-Means » [MacQueen 1967] et « PAM » [Kaufman and Rousseeuw 1990], des adaptations des méthodes des mélanges de lois et des méthodes fondées sur la théorie des graphes pour produire des schémas recouvrants.

Ce mémoire présente une étude détaillée de quelques méthodes de classification recouvrante qui ont été proposées dans la littérature. L'objectif de cette étude détaillée consiste à déterminer les problématiques dans ce type de schéma de classification. Nous avons concentré notre travail sur les méthodes de réallocation dynamique autour des centres mobiles. Nous avons ensuite essayé de proposer des solutions pour certaines de ces

problématiques. Tout d'abord, nous avons proposé une estimation du nombre de classes utile pour toutes les méthodes qui considèrent que le nombre de classes est fixé a priori. Cette estimation est basée sur l'extraction des valeurs propres à partir d'une matrice de noyaux. Ensuite, pour résoudre la problématique des classes ayant des formes concentriques et des formes variées, nous avons adapté en premier lieu une distance basée sur les méthodes de noyaux sur une des méthodes de classification recouvrante qui est la méthode « OKM : Overlapping K-Means »[Cleuziou 2007]. En deuxième lieu, nous avons proposé une nouvelle méthode de classification recouvrante désignée sous l'appellation « KOKM :Kernel Overlapping K-Means » qui pourrait être considérée comme une amélioration qui s'ajoute à celle des distances basées sur les méthodes de noyaux. Enfin, pour valider les solutions proposées, nous avons testé ces solutions sur deux bases de données du benchmark à savoir « Iris » et « Ionosphère » et une base recouvrante réelle « Eachmovie » portant sur la classification des vidéos appartenant à plusieurs genres.

Le premier chapitre de ce mémoire définit la classification automatique, énumère des différentes problématiques existantes dans ce domaine et présente la technique des méthodes de noyaux qui est de plus en plus utilisées dans les méthodes de classification automatique. Les méthodes étudiées de classification recouvrante font l'objet du deuxième chapitre. La problématique du nombre de recouvrements et la problématique des classes de formes concentriques et de formes variées seront traitées dans le troisième chapitre où les méthodes de noyaux seront utilisées comme solution permettant de résoudre ces deux problématiques. Le dernier chapitre présente les différentes expérimentations que nous avons réalisées pour valider les différentes solutions proposées.

Chapitre 1
Etat de l’art de la classification automatique

Introduction

La Classification automatique est un domaine de recherche étudié depuis de nombreuses années. Plusieurs disciplines se réunissent et adaptent leurs techniques, leurs théories et leurs outils pour produire des schémas de classification. La communauté de la modélisation de données considère que la classification automatique est dirigée par la science des mathématiques, de statistiques et de l’analyse numérique. La communauté « Machine Learning » considère que la classification automatique consiste en un apprentissage non supervisé pour extraire des modèles cachés [Xu and Wunsch 2005].

Ce chapitre présente le domaine de la classification automatique de manière générale et présente les concepts de la technique des fonctions de noyaux sur laquelle repose la solution que nous allons proposer. La première section définit la notion de classification automatique et d’apprentissage automatique. Les différents types de schémas de classification seront présentés dans la seconde section alors que la troisième section énumère les problématiques actuelles dans la classification automatique qui ont fait l’objet de plusieurs publications et de plusieurs travaux de recherches. Enfin la quatrième section présente une description de la technique des fonctions de noyaux qui est de plus en plus utilisée dans les algorithmes de classification.

1.1. Classification automatique : le processus d’apprentissage non supervisé

L’objectif général de la classification est d’étiqueter des données en leur associant une classe. Si les classes possibles sont connues a priori et si les exemples sont fournis avec les étiquettes de leur classe, on parle d’apprentissage supervisé ou d’analyse discriminante. Dans ce cas, l’apprentissage est basé sur les exemples déjà classés pour apprendre un modèle qui permet ensuite de prédire la classe d’un nouvel exemple. Si la classe est représentée par une variable numérique non catégorique, on parle de régression [Berkhin 2002].

Au contraire, si les exemples sont non étiquetés et si les classes et leur nombre sont inconnus, on parle d’apprentissage non supervisé ou classification automatique. L’objectif général consiste à séparer un ensemble d’objets en différents groupes en fonction d’une certaine notion de similarité. Les objets qui sont considérés comme similaires sont associés au même cluster alors que ceux qui sont considérés comme différents sont associés à des clusters différents. Donc, le but est de maximiser les ressemblances intra cluster et de minimiser les ressemblances inter clusters.

D’autres types d’apprentissage sont apparus progressivement comme que l’apprentissage semi-supervisé. Ce type d’apprentissage se base sur des données étiquetées et des données non étiquetées. Son principe est que les exemples étiquetés vont servir pour classifier les exemples non étiquetés.

Un autre type d’apprentissage est l’apprentissage par renforcement où l'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage [Cornuéjols and *al.* 2002].

Dans ce mémoire, nous nous intéressons seulement à la classification automatique (apprentissage non supervisé). La classification automatique ne constitue pas une simple tâche à appliquer directement sur les données. Elle consiste en un processus complet qui englobe plusieurs étapes. La négligence d’une étape de ce processus peut avoir une influence négative sur la qualité de classification.

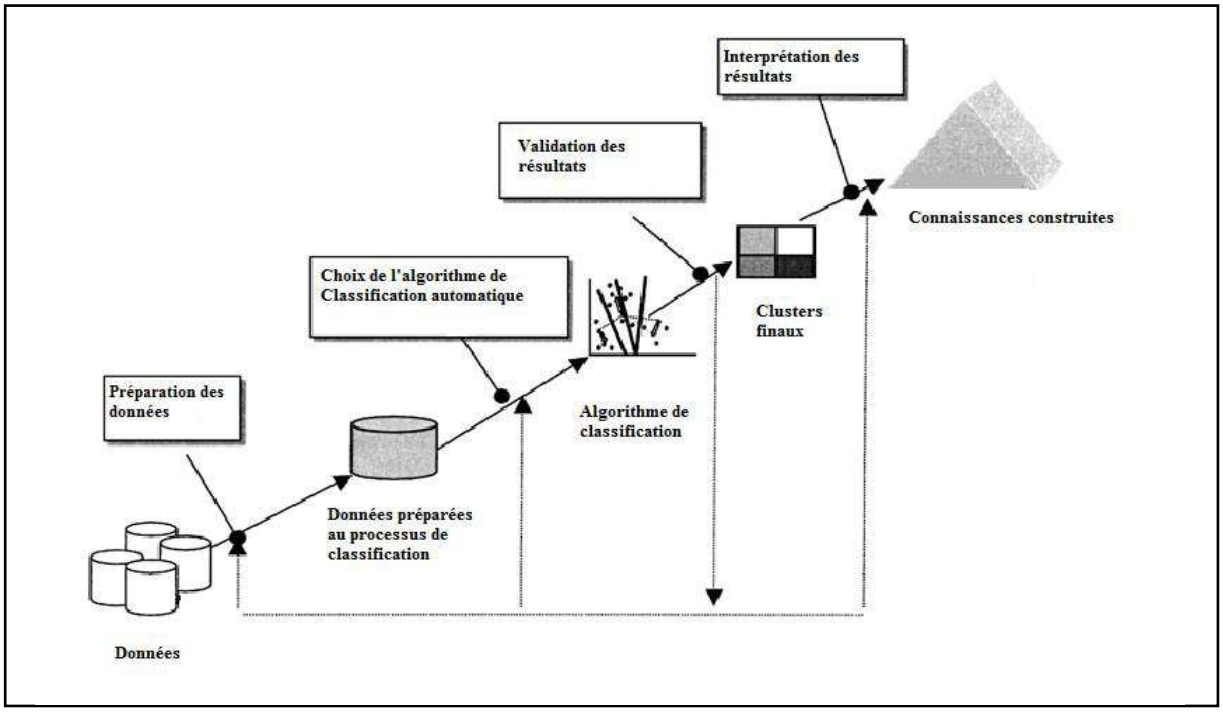


Figure 1. Les étapes du processus de classification automatique [Fayyad and *al.* 1996].

Le processus de classification consiste au moins en quatre étapes non négligeables [Fayyad and *al.* 1996]. La figure 1. illustre ces différentes étapes : la première, est l’étape de préparation des données. Cette étape englobe la représentation des individus qui a pour but de

déterminer le nombre d'individus disponibles, le type et l'échelle des attributs de données. Elle englobe aussi la sélection des attributs les plus discriminants pour réduire la dimensionnalité des données. La deuxième étape consiste à choisir l'algorithme de classification qui répond aux spécificités des données et aux spécificités de l'application désirée. Le choix de l'algorithme peut être accompagné par le choix des métriques de distance entre les objets selon leur nature. L'étape suivante consiste en la validation du résultat qui vise à déterminer si les classes fournies sont significatives en utilisant un critère spécifique d'optimalité. Cependant, un tel critère est souvent subjectif ce qui explique la difficulté de la validation des algorithmes de classification automatique. La dernière étape est celle de l'interprétation des résultats. Elle est généralement faite par un expert du domaine qui intègre les résultats obtenus au cours de la classification avec d'autres informations ou d'autres résultats pour en conclure les meilleures interprétations.

1.2. Catégorisation des méthodes de classification

Plusieurs méthodes de classification non supervisée existent dans la littérature. Il n'existe aucun critère ni aucune méthode qui est considéré meilleur que les autres sur l'ensemble des problématiques et des problèmes envisageables. Par contre, certains critères et certaines méthodes sont plus appropriés que d'autres dans certains domaines d'application et face à certains problèmes.

Ces méthodes peuvent être catégorisées selon plusieurs manières. Nous pouvons classer ces méthodes selon le type de schéma obtenu après classification. Trois grands types de schémas de classification sont retenus [Cleuziou and *al.* 2004] à savoir : les schémas de classification stricte, les schémas de classification floue et les schémas de classification recouvrante.

1.2.1. Méthodes de classification Stricte

Le principe des méthodes de classification stricte est de créer des partitions disjointes de telles sortes que chaque objet appartient à une seule partition. L'appartenance d'un objet à une partition exclut totalement la probabilité d'appartenance de cet objet à une autre partition. Le tableau 1 montre une répartition stricte de six objets sur trois partitions formées. Nous remarquons que chaque ligne de la matrice de distribution contient exactement un seul '1' indiquant l'appartenance de l'objet en question à la partition correspondante.

	Classification Stricte		
Objets	A	B	C
1	0	1	0
2	1	0	0
3	0	1	0
4	1	0	0
5	0	1	0
6	0	0	1

Tableau 1. La distribution des données entre les 3 partitions A, B et C.

La Figure 2. illustre les partitions disjointes formées par les méthodes de classification stricte.

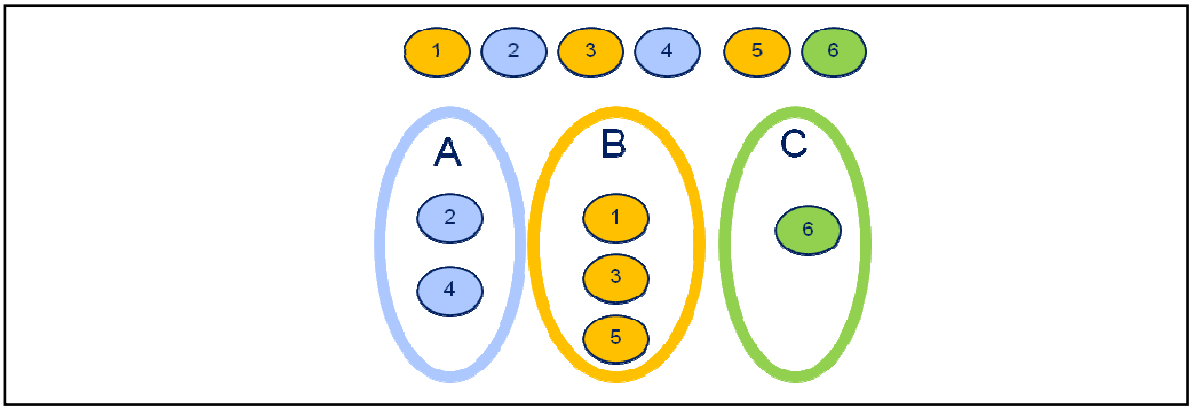


Figure 2. Exemple de partitions formées par les méthodes de classification stricte.

Nous pouvons distinguer deux grandes familles de méthodes de classification automatique stricte à savoir les méthodes stratifiées et les méthodes par partitionnement.

1.2.1.1. Méthodes Stratifiées

Les méthodes stratifiées se présentent comme la succession de partitions emboîtées et peuvent être représentées graphiquement à l'aide de dendrogrammes. La figure 3. présente un dendrogramme construit par une méthode de classification hiérarchique sur un ensemble de six objets. Il existe deux grands types de méthodes hiérarchiques [Berkhin 2002] :

- Les méthodes hiérarchiques ascendantes : à partir des singletons (objets), un dendrogramme est construit successivement jusqu'à ce que tous les objets soient groupés dans un seul cluster. Nous pouvons citer à titre d'exemples les méthodes : « SLINK » [Sibson 1973], « CLINK » [Defays 1977], « Voorhees method » [Voorhees 1986], « CURE » [Guha and *al.* 1998], « CHAMELEON » [Karypis and *al.* 1999].

- Les méthodes hiérarchiques descendantes : à partir de l'ensemble initial des objets, des itérations successives sont effectuées jusqu'à ce que chaque objet soit représenté dans un cluster à part. Cette technique est très peu utilisée en pratique. Nous pouvons citer à titre d'exemples les méthodes: « Principal Direction Divisive Partitioning » [Boley 1998], « COBWEB » [Fisher 1987], « CLASSIT » [Gennari and *al.*1989].

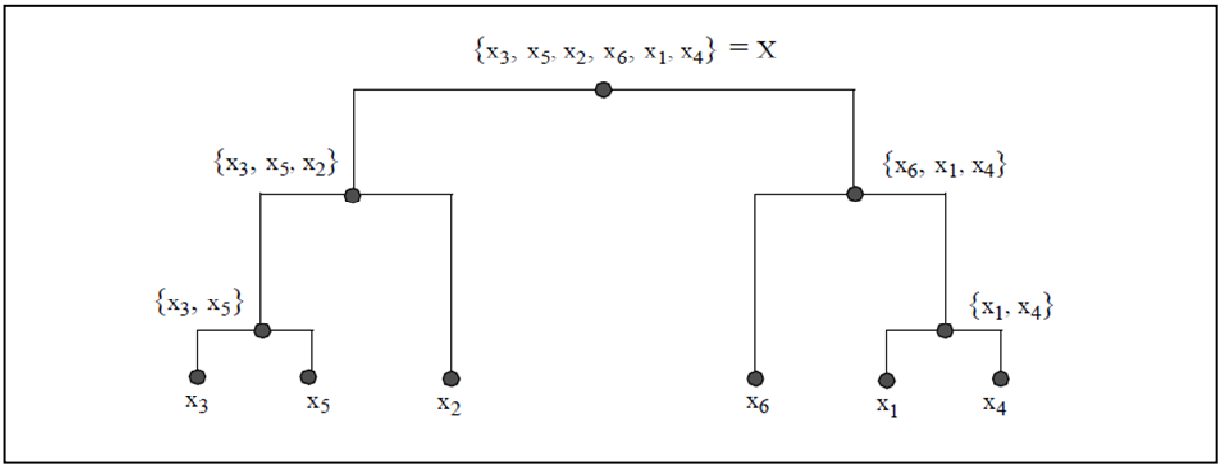


Figure 3. Dendrogramme des partitions emboîtées d'un ensemble de 6 objets.

La définition d'une mesure de distance entre clusters dans la classification hiérarchique influence largement la qualité de la classification. Il existe trois grands types de distances [Halkidi and *al.* 2004] :

- Saut minimal (Single-link) : la distance entre deux clusters est le minimum des distances entre toutes les paires d'objets appartenant à des clusters différents.
- Saut maximal (Complete-link) : la distance entre deux clusters est le maximum des distances entre toutes les paires d'objets appartenant à des clusters différents.
- Saut moyen (Average-link) : la distance entre deux clusters est la moyenne des distances entre toutes les paires d'objets appartenant à des clusters différents.

1.2.1.2. Méthodes par partitionnement

Les méthodes de classification automatique par partitionnement suivent plusieurs itérations pour créer des partitions optimales. Contrairement aux méthodes hiérarchiques, la qualité de partitionnement est évaluée et optimisée après chaque itération. Plusieurs approches sont utilisées dans la littérature pour créer des partitions. Nous pouvons catégoriser les méthodes par partitionnement selon les approches utilisées pour implémenter ces méthodes. Nous distinguons les méthodes probabilistes, les méthodes k-moyennes, les méthodes k-medoids, les méthodes basées sur la densité et les méthodes basées sur les grilles.

Les méthodes probabilistes se basent sur l'hypothèse que les données sont générées à partir d'un mélange de lois de distribution. Généralement, les méthodes probabilistes utilisent la méthode EM (expectation maximisation) pour optimiser itérativement une fonction objective spécifique pour chacune de ces méthodes. Nous pouvons citer à titre d'exemples les méthodes : « MCLUST » [Fraley and Raftery 1999], « SNOB » [Wallace and Dowe 1994], « AUTOCLASS » [Cheeseman and Stutz 1996].

Les méthodes k-moyenne [MacQueen 1967] sont les méthodes les plus utilisées en pratique dans plusieurs domaines étant donné la facilité et la rapidité de leur application. Elles consistent à optimiser itérativement l'inertie intra cluster en supposant que le nombre de clusters est connu à l'avance. Plusieurs variantes de k-moyennes telle que la méthode « k-

prototype » [Huang 1997] ont été proposées dans la littérature pour généraliser cette méthode à d'autres distances non euclidiennes et pour traiter les variables non catégoriques.

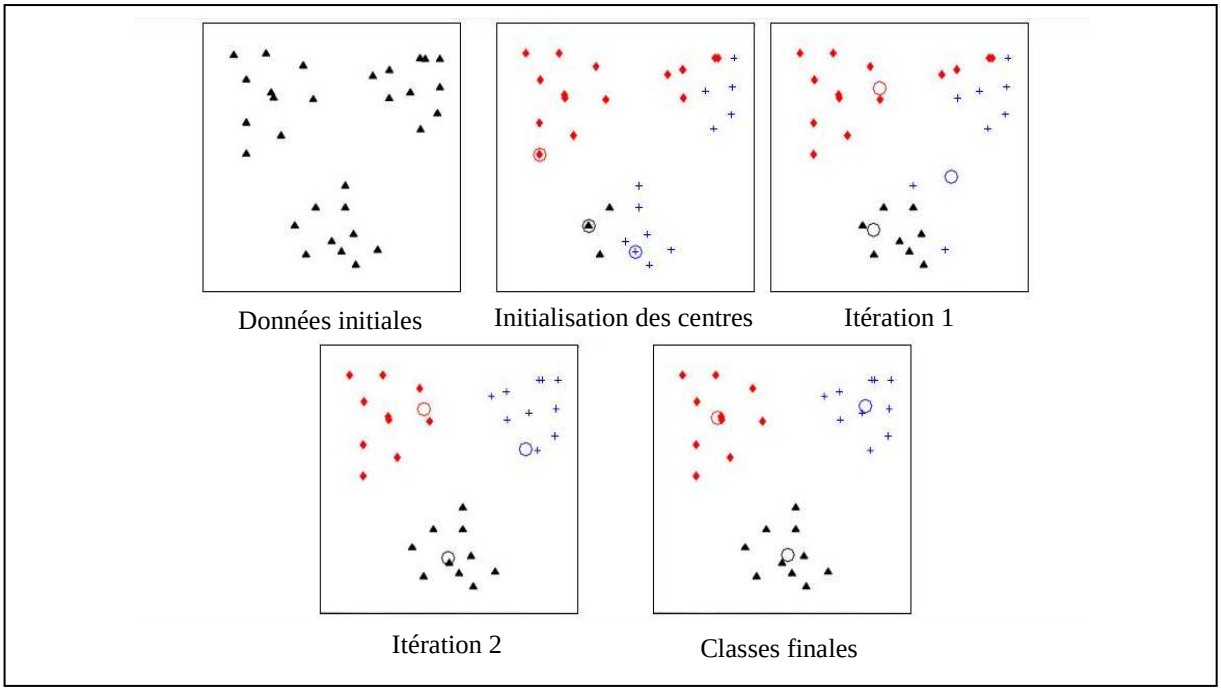


Figure 4. Mise à jour des centres des classes dans la méthode k-moyennes.

Les méthodes k-medoids se basent sur le principe que chaque cluster est représenté non pas par un centre calculé à partir des différents points appartenant à ce cluster mais par l'un de ces points. L'avantage de ces méthodes c'est qu'elles permettent de traiter n'importe quel type de variables tout en permettant aussi une tolérance aux données éloignées. Parmi les méthodes k-medoids proposées dans littérature, la méthode « PAM : Partitioning Around Medoids », la méthode « CLARA :Clustering LARge Applications » [Kaufman et Rousseeuw 1990] et la méthode « CLARANS : Clustering Large Applications based upon RANdomized Search » [Ng and Han 1994].

Les méthodes basées sur la densité sont une extension des méthodes des centres mobiles. Les différents clusters ne sont plus représentés par leur centre de gravité uniquement, mais par leurs « noyaux ». Un noyau peut être un ensemble de points, un axe principal ou un plan principal. Il faut donc disposer d'une fonction qui, à un ensemble donné, associe son noyau. Il faut ensuite réaffecter les points aux différents noyaux. Parmi les méthodes qui existent dans la littérature, la méthode « DBSCAN » [Ester and *al.* 1996].

Les méthodes basées sur les grilles consistent à partitionner l'espace en plusieurs cellules, ensuite à identifier les cellules connectées les plus denses pour former les clusters. Les paramètres des ces méthodes consistent en la taille de la grille et la densité minimale déterminante si une cellule est considérée comme dense ou non. La problématique de ce type de classification se manifeste dans le choix de ces deux paramètres. Le choix de cellules de trop petite taille amène à une estimation bruitée de la densité (sur-partitionnement), et le choix de cellules de trop grande taille amène au problème de sous-partitionnement. Parmi les

méthodes de classification basées sur les grilles, la méthode « STING :STatistical INformation Grid-based method » [Wang and *al.* 1997], la méthode « WaveCluster » [Sheikholeslami and *al.* 1998].

1.2.2. Méthodes de classification floue

Le principe des méthodes de classification floue consiste à créer des groupes non disjoints où un objet peut appartenir à un ou plusieurs groupes selon des degrés d'appartenance. L'appartenance d'un objet à un ou plusieurs groupes est définie par le degré de flou que l'algorithme de classification a attribué à un objet donné concernant son appartenance à un des groupes formés. Donc, pour offrir plus de précision au décideur concernant les appartenances d'un objet à un groupe, les méthodes de classification floue construisent des degrés d'appartenance qui vont servir à la bonne interprétation des résultats de classification. Le tableau 2. montre une répartition probabiliste de six objets selon une méthode de classification floue sur trois groupes A, B et C. Nous remarquons dans cette distribution qu'un objet peut appartenir à un groupe avec une valeur d'appartenance comprise dans l'intervalle [0..1] de telle sorte que la somme des degrés d'appartenance à tous les groupes soit égale à 1. Par exemple, pour l'objet n°6, il a une forte probabilité d'appartenir au groupe B ou bien au groupe C. La seule information dont sûre est que cet objet a de faibles chances d'appartenir au groupe A. La figure 7. illustre l'appartenance floue des objets à différents groupes en considérant les degrés faibles (<0.1) comme nul.

	Classification floue		
Objets	A	B	C
1	0.02	**0.92**	0.06
2	**0.54**	0.01	**0.45**
3	**0.35**	**0.62**	0.03
4	**0.92**	0.03	0.05
5	0.08	**0.88**	0.04
6	0.02	**0.42**	**0.56**

Tableau 2. La distribution des données entre les 3 partitions A, B et C par une méthode de classification floue.

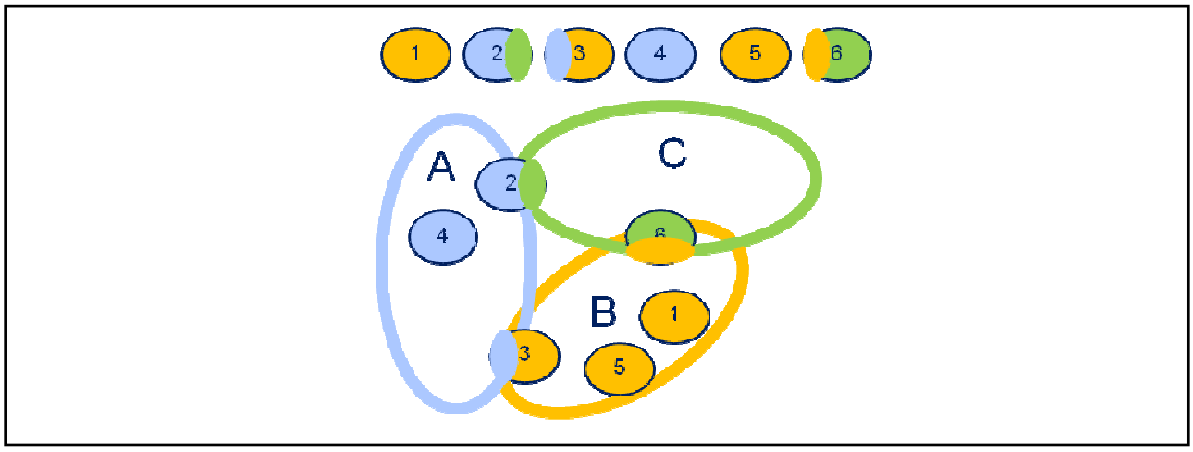


Figure 5. Exemple de groupes formés par les méthodes de classification floue

Parmi les méthodes de classification floue qui existent dans la littérature, nous distinguons la méthode « c-moyenne floue » [Bezdek 1981] qui se base sur la minimisation itérative d'un critère de type erreur quadratique. Il existe toute une famille de fonctions d'erreurs associées à cette méthode qui se distinguent par des valeurs différentes prises par un paramètre *m* appelé indice de floue. D'autres variantes de la méthode « c-moyenne floue » ont été proposées afin d'en accroître les performances. Ces variantes sont souvent dédiées à une application particulière ou bien généralisent l'algorithme de base au cas de distance non euclidienne ou encore utilisent une autre théorie pour modéliser le flou [Khodja 1997]. Parmi ces variantes, la méthode « c-moyenne possibiliste » [Krishnapuram and Keller 1993]. Elle est basée sur la théorie de possibilité pour modéliser les différents degrés d'appartenance d'un objet aux différents clusters. Les performances en présence du bruit sont meilleures par rapport aux résultats obtenus par la méthode « c-moyenne floue ».

1.2.3. Méthodes de classification recouvrante

La classification recouvrante peut être considérée comme un compromis entre la classification floue et la classification stricte. L'objectif des méthodes de classification stricte est de créer des groupes joints afin de permettre à un objet d'appartenir à plusieurs groupes sans établir des degrés d'appartenance. Ce type de schéma correspond à une organisation naturelle des données dans plusieurs applications. Par exemple, en recherche d'information, un même document peut porter sur plusieurs thématiques, en réseaux sociaux un individu peut appartenir à plusieurs communautés, etc. Le tableau 3. montre un exemple de distribution obtenue par les méthodes de classification recouvrante. Chaque ligne dans le tableau contient un ou plusieurs '1' indiquant l'appartenance de cet objet à une ou plusieurs classes. La figure 6. illustre l'appartenance multiple du même objet à plusieurs recouvrements. Par exemple l'objet n°6 appartient à la classe C et aussi il appartient à la classe B.

	Classification recouvrante		
Objets	A	B	C
1	0	**1**	0
2	**1**	0	**1**
3	**1**	**1**	0
4	**1**	0	0
5	0	**1**	0
6	0	**1**	**1**

Tableau 3. La distribution des données entre les 3 recouvrements A, B et C par une méthode de classification recouvrante.

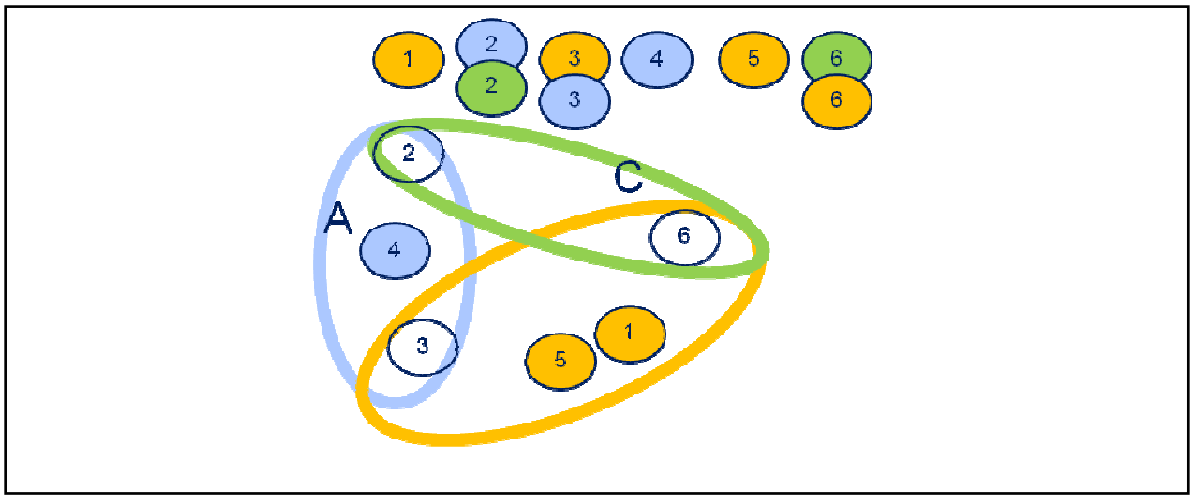


Figure 6. Exemple de recouvrements formés par les méthodes de classification recouvrante.

Plusieurs méthodes ont été proposées pour produire des schémas recouvrants. Nous pouvons catégoriser les méthodes de classification recouvrante selon les approches qui ont été utilisées afin d'implémenter ces méthodes. Les premières méthodes proposées étaient stratifiées. L'avantage principal de ces méthodes est de proposer une interprétation visuelle des classes et de leur organisation. Parmi ces méthodes, nous distinguons la méthode des « pyramides » [Diday 1984] et la méthode des « k-hiérarchies faibles » [Bertrand and Janowitz 2003]. D'autres méthodes sont basées sur l'extension des méthodes de classification floue pour produire des recouvrements. Ces méthodes se basent sur l'hypothèse qu'un bon schéma de classification recouvrante peut être une extension d'un bon schéma de classification floue. Une autre famille de méthodes recouvrantes est basée sur l'approche de réallocation dynamique. Nous présentons dans le deuxième chapitre une étude détaillée de ce type de méthodes de classification recouvrante.

1.3. Les Problématiques dans la classification automatique

Eu égard à l'importance du processus de classification dans plusieurs domaines, la communauté travaillant sur la classification automatique ne cesse de proposer des méthodes et des techniques traitant plusieurs problématiques et plusieurs thématiques et ne cesse d'étendre ce processus à d'autres domaines applicatifs. Chaque problématique citée ci-dessous peut être considérée comme un axe de recherche dans la classification et ayant fait l'objet de plusieurs travaux de recherches.

1.3.1. Notion de similarité

L'objectif de la classification est de regrouper les objets dans plusieurs clusters de telle sorte que les objets similaires appartiennent aux mêmes clusters et les objets dissimilaires appartiennent à des clusters différents. Pour regrouper des objets on a besoin de définir une notion de similarité. Une fois définie, elle est utilisée comme une règle pour regrouper les objets. Plusieurs distances sont proposées dans la littérature. La distance la plus utilisée pour calculer des distances entre des données multidimensionnelles est la distance euclidienne. Elle est définie par :

$$dist\,(x, y) = \sum_d \left(x_d - y_d\right)^{1/2}$$

Elle est généralement calculée sur des données non standardisées mais normalisées (sur la même échelle). D'autre mesures de distances sont utilisées telles que :

Manhattan distance : $dist\,(x, y) = \sum_d \left|x_d - y_d\right|$

Tchebychev distance : $dist\,(x, y) = Max\left|x_d - y_d\right|$

Le choix d'une distance appropriée est considéré parmi les principales problématiques dans la classification automatique.

1.3.2. Détermination du nombre de classes

Un autre problème, dans la classification automatique, concerne la détermination du nombre de clusters. Malheureusement, cette information est inconnue dans la plupart des cas. Plusieurs travaux ont traité ce problème dans la littérature [Dunn 1973, Johnson 1967, Figueiredo and Jain 2002, Hansen and Yu 2001]. Certaines méthodes de classification automatique sont non paramétriques et ne nécessitent pas le nombre de clusters comme paramètre. Le nombre de clusters dans ces méthodes est construit itérativement au cours de la phase d'apprentissage de l'algorithme. Le nombre de cluster est introduit comme une variable dans la fonction objective à optimiser.

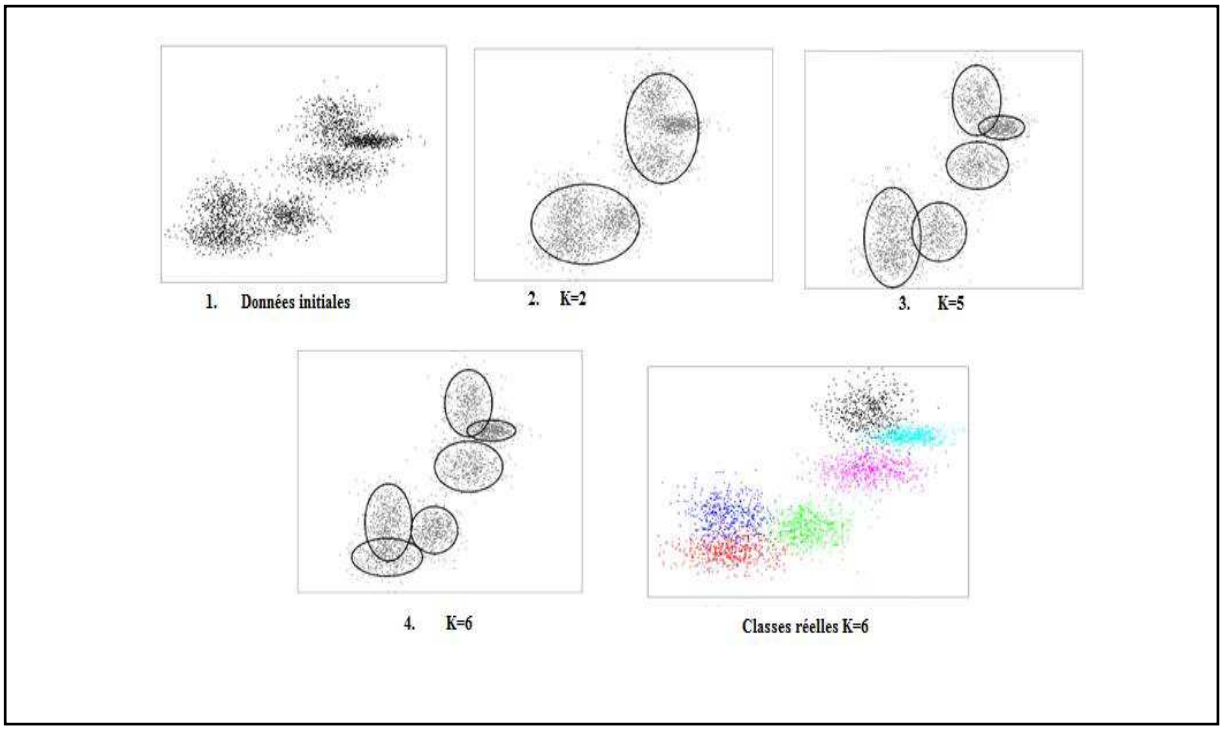


Figure 7. Estimation du nombre de clusters par densité.

D'autres méthodes donnent une estimation du nombre de clusters [Girolami 2002]. Par exemple dans la classification spectrale, le nombre de clusters est déterminé à partir de la matrice de noyaux. Le nombre des valeurs propres, les plus significatives, détermine le nombre de clusters. D'autres méthodes sont basées sur le principe de densité pour déterminer les régions les plus denses qui vont former les différentes classes. La figure 7. illustre l'estimation du nombre de clusters en utilisant l'estimation de densité.

1.3.3. Identification des classes de formes complexes et de densités variées

Il est très important que l'algorithme de clustering proposé permette d'identifier différentes formes de classes. En général, dans les applications réelles, les classes n'ont pas de formes sphériques, la séparation entre ces données doit tenir compte des séparations complexes : les classes peuvent avoir des formes ellipsoïdales, des formes concentriques (contenus les unes dans les autres).

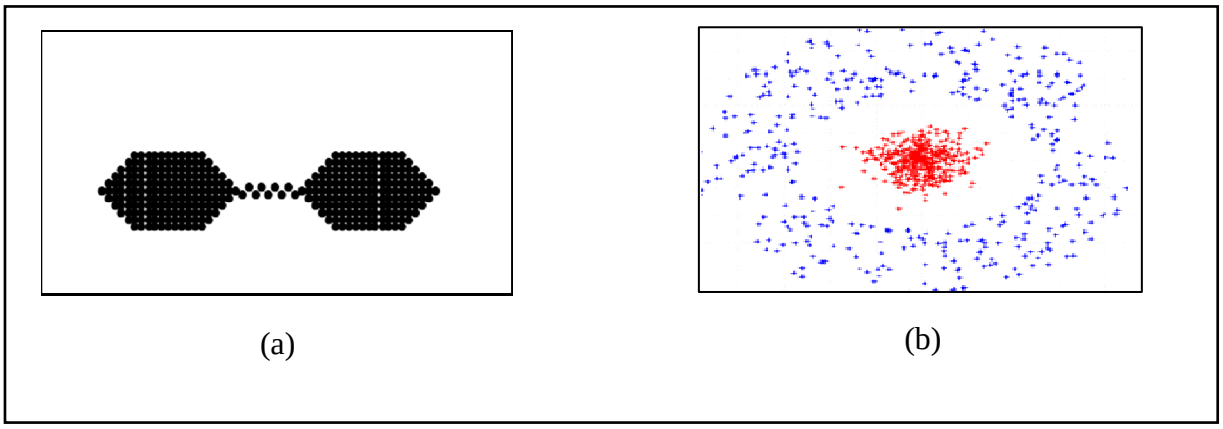


Figure 8. Clusters de formes complexes : (a) clusters sous formes de chaine (b) clusters de formes concentriques.

Les méthodes de classification automatique basées sur la densité permettent de traiter la problématique des classes ayant des séparations complexes. Mais, l'inconvénient de ces méthodes est qu'elles ne permettent pas d'identifier les classes de densités variées.

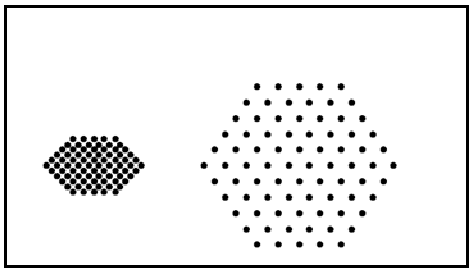


Figure 9. Clusters de densités variées.

Les données peuvent avoir à la fois des formes complexes et des densités variées comme dans la figure 8. Le critère de la capacité à identifier des clusters ayant n'importe quelle forme et n'importe quelle densité peut être un bon critère de comparaison entre les méthodes de classification automatique.

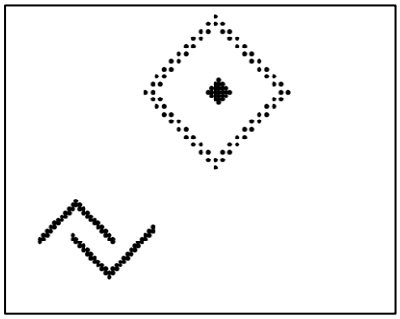


Figure 10. Clusters de formes concentriques et de densités variées.

1.3.4. Traitement des données éloignées

Les données éloignées sont des données qui ne suivent pas la distribution générale des données. Certains types de méthodes de classification permettent de traiter ce problème comme les méthodes basées sur les grilles et les méthodes basées sur la densité. Les valeurs éloignées ne sont pas toujours éliminées au cours de processus d’apprentissage car elles peuvent être utiles dans certains types d’application. Par exemple dans la sécurité des systèmes d’information, une valeur éloignée peut indiquer la présence d’une attaque ou d’une anomalie dans le système.

1.3.5. Evaluation et validation des résultats de la classification

L’évaluation des résultats est une problématique importante dans l’apprentissage non supervisé. La pertinence des partitions formées au cours de la phase d’apprentissage est un critère subjectif par nature vu qu’il existe généralement plusieurs groupements pertinents sur les mêmes données. Parmi les méthodes d’évaluation l’utilisation d’un ensemble de données où le nombre de partitions est connu. Ce type d’évaluation est appelé évaluation interne[Xu and Wunsch 2005]. Une autre méthode de validation est d’utiliser un ensemble étiqueté et de vérifier si les groupes construits permettent de trouver les classes initiales. Le problème de ce type d’évaluation est que les groupes pertinents ne correspondent pas obligatoirement aux classes initiales.

Un autre type d’évaluation appelé évaluation externe utilise le critère d’inerties intra cluster ou un critère de séparation entre clusters. Le problème dans cette méthode d’évaluation est que la séparation devient un critère non pertinent si le chevauchement entre cluster est important,

1.4. Les méthodes de noyaux

L’idée des méthodes de noyaux est de transformer les données à partir de l’espace original en un espace de dimension plus grande par une transformation non linéaire. C’est dans ce nouvel espace qu’il est possible de trouver un hyperplan qui sépare les classes. Cet espace est muni d'un produit scalaire, appelé fonction de noyaux. Un classifieur linéaire dans cet espace se comporte comme un classifieur non linéaire dans l’espace original [Girolami 2002]. L'astuce du noyau (kernel trick) a été utilisé dans les méthodes des Machines à vecteurs de support. L'idée a depuis été appliquée à plusieurs types d'algorithmes en apprentissage automatique et en statistiques. La figure 14. illustre l’idée des méthodes de noyaux en projetant les données avec une fonction polynomiale de degré trois.

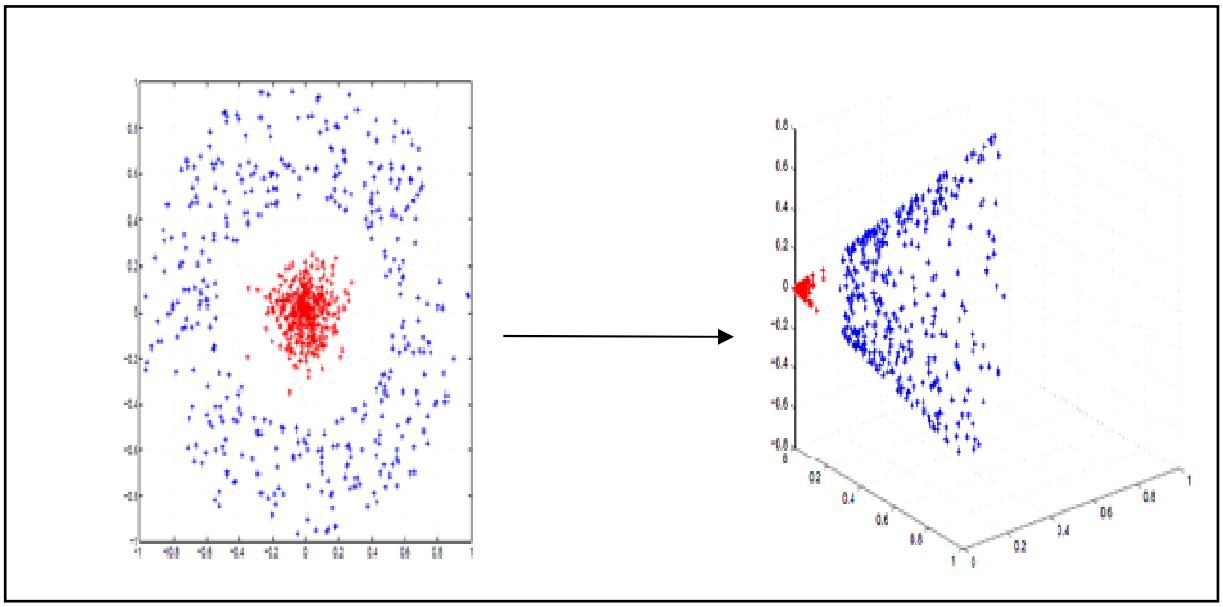


Figure 11. Clusters non linéairement séparables projetés sur une troisième dimension où ils deviennent séparables [Schölkopf and Smola 2002]

L'astuce du noyau s'utilise dans un algorithme qui ne dépend que du produit scalaire entre deux vecteurs d'entrée x et y. Après passage à un espace de redescription par une transformation ϕ, l'algorithme n'est plus dépendant que du produit scalaire: $\phi(x)*\phi(y)$. Le problème de ce produit scalaire est qu'il est effectué dans un espace de grande dimension, ce qui conduit à des calculs impraticables [Schölkopf 2002]. L'idée est donc de remplacer ce calcul par une fonction noyau telle que: $K(x,y)=\phi(x)*\phi(y)$.

Le théorème de « Mercer » [Schölkopf and Smola 2002]montre qu'un noyau K défini positif sur un espace X peut être écrit sous la forme suivante :

$$K(x,y) = \sum_{i=1}^{N} \lambda_i u_i(x) u_i(y) \qquad [1.1]$$

où λ_i et u_i sont respectivement les valeurs propres et les vecteurs propres. Dans ce cas, la fonction

$$\phi : \phi(x) = \begin{pmatrix} \sqrt{\lambda_1}\phi_1(x) \\ \sqrt{\lambda_2}\phi_2(x) \\ \vdots \\ \sqrt{\lambda_n}\phi_n(x) \end{pmatrix} \quad \text{satisfait l'équation :} \quad K(x,y)=\phi(x)*\phi(y)$$

Un noyau défini positif sur l'ensemble X est une fonction $K : X \times X \rightarrow \Re$ symétrique qui satisfait :

$$\forall n \in N \ , \ 1 \leq i \leq n \ , \ x_j \ et \ x_i \in X^n, \ c_i \ et \ c_i \in \Re^n \qquad \sum_{i,j}^{n} K(x_i, x_j) \ c_i c_j \geq 0 \qquad [1.2]$$

A partir de la définition précédente, nous pouvons déterminer que si K est un noyau défini positif sur un espace X , il existe un espace de Hilbert H muni du produit scalaire $\langle .,. \rangle_{\mathrm{H}}$ et une application ϕ définie par :

$$\begin{aligned} &\phi : \ X \rightarrow \mathrm{H} \\ &\forall (x_i, x_j) \in X^2 \qquad K(x_i, x_j) = \langle \phi(x_i) \ , \ \phi(x_j) \rangle_{\mathrm{H}} \end{aligned} \qquad [1.3]$$

L'astuce du noyau consiste donc à remplacer un produit scalaire dans un espace de grande dimension par une fonction noyau, qui est plus facile à calculer. De cette manière, un classifieur linéaire peut facilement être transformé en un classifieur non linéaire. Un autre avantage des fonctions noyaux est qu'il n'est pas nécessaire d'expliciter la transformation ϕ. Celle-ci, peut même transformer l'espace d'entrée en un espace de redescription infini, comme le noyau gaussien:

$$K(x, y) = \exp \left\{ \frac{- \| x_i - y_i \|^2}{\sigma^2} \right\} \qquad [1.4]$$

D'autres fonctions de noyaux sont utilisées dans la littérature telles que :

Noyau polynômial : $K(x, y) = \{ \langle x, y \rangle + 1 \}^d$ [1.5]

Noyau linéaire : $K(x, y) = \langle x, y \rangle$ [1.6]

1.4.1. La méthode « Kernel k-means »

La méthode « kernel k-means » [Camastra and Verri 2005] est une méthode qui étend l'algorithme « SVC : support vector clustering » [Ben-Hur and *al.* 2000] pour produire des partitions. L'inconvénient de l'algorithme « SVC » est qu'il ne permet de détecter qu'un seul groupe. « Kernel k-means » résoud ce problème en généralisant la méthode « SVC » pour produire plusieurs partitions.

Étant donné un ensemble de données X , cet ensemble est projeté dans un nouvel espace noté F en utilisant des fonctions de transformation non-linéaires ϕ. A la différence de la méthode « SVC », « Kernel k-means » considère k[1], dont la valeur est à priori fixe, des centres dans l'espace de redescription $(a_i \in F \quad i = 1, ..., k)$ et considère le jeu de données $A = (a_i, ..., a_k)$ l'espace de « Livre de code » dans F (en anglais Codebook) puisque les centres dans l'espace de redescription jouent le même rôle des vecteurs de code (en anglais Codevectors) dans l'espace d'origine.

[1] « k » en minuscule présente le nombre de classes et « K » en majuscule présente la fonction de noyau

Par analogie avec les « code-vectors » dans l'espace d'origine, « Kernel k-means » définit pour chaque centre a_c son propre ensemble de « Voronoi set » dans l'espace de redescription. Le Jeu de « Voronoi set » FV_c dans l'espace F du centre a_c est défini par l'ensemble de tous les vecteurs $x_i \in X$ tel que a_c est le vecteur le plus proche pour leurs images $\phi(x_i)$ dans l'espace F :

$$FV_c = \left\{ x_i \in X \;\; \middle| c = \arg\min_j \| \phi(x_i) - a_j \| \right\} \qquad [1.7]$$

La méthode « kernel K-Means » utilise la même stratégie que k-moyennes, c'est-à-dire elle recalcule à chaque itération tous les centres a_c dans l'espace F, jusqu'à ce qu'aucun centre ne change.

Un autre avantage de la méthode « Kernel K-Means » par rapport à la méthode k-moyennes consiste à traiter les données éloignées. « Kernel K-Means » est plus robuste que k-moyennes puisque elle définit une distance maximale p entre l'objet x_i et le centre du cluster dans l'espace F. Les $FV_c(p)$ sont définis par l'équation ci-dessous :

$$FV_c(p) = \left\{ x_i \in FV_c \quad et \quad \| \phi(x_i) - a_c \| < p \right\} \qquad [1.8]$$

Avec $FV_c(p)$ est un ensemble de « Voronoi set » dans l'espace de redescription du centre a_c ne contenant pas des données éloignées. Il s'agit des représentations de points dont la distance du centre est plus petite que le paramètre p . La description de la méthode « kernel k-means » est décrite en pseudo-code dans l'annexe 3.

1.4.2. SKK: Soft Kernel K-means

La méthode « SKK : Soft Kernel K-means » [Daoqiang and Zhang 2002] commence par la construction d'une matrice de noyaux de taille *NxN*(*N* est le nombre des individus à classifier). Les valeurs de la matrice contiennent le produit scalaire entre deux vecteurs quelconques dans l'espace de représentation de données. L'algorithme « SKK » interprète cette valeur comme une probabilité que ces deux vecteurs soient dans le même cluster. Ainsi, elle est interprétée comme une mesure de similarité.

La méthode « SKK » étend la méthode « kernel k-means » pour travailler avec les modèles de mélange de lois. En utilisant l'algorithme « EM :Expectation Maximisation », la fonction à maximiser itérativement est la suivante :

$$\sum_{i=1}^{N}\sum_{j=1}^{k} h_{x_i}^{C_j} \log\left(P\left(C_j\right)P\left(\phi(x_i)/\theta_j\right)\right) \qquad [1.9]$$

Où $\phi(x_i)$ est la représentation de x_i dans un espace de forte dimensionnalité, C_j représente le J ème cluster**,** θ_j représente le paramètre du cluster *j*, $P\left(C_j\right)$ est la probabilité a priori du cluster *j* et $h_{x_i}^{C_j}$ est la probabilité a posteriori $P\left(z_i = c_j / \theta, x_i\right)$ que le point x_i soit généré à partir du cluster C_j.

L'algorithme « EM » nécessite à chaque itération de calculer la distance euclidienne entre un objet et le centre d'une classe. Cette distance doit être calculée dans l'espace de description des données.

$$dist(x_i, U_j) = \left\| x_i - U_j \right\|^2 \qquad [1.10]$$

Où U_j représente le centre du cluster j dans le nouvel espace de redescription. Dans l'espace de redescription, le centre du cluster est calculé comme suit :

$$U_j = \frac{\sum_{i=1}^{N} h_{x_i}^{C_j} \phi(x_i)}{\sum_{i=1}^{N} h_{x_i}^{C_j}} \qquad [1.11]$$

La distance entre le point et le centre peut être calculée comme suit :

$$dist(\phi(x_i), U_j) = \phi(x_i).\phi(x_i) - \frac{\sum_{l=1}^{N} h_{x_l}^{C_j} \phi(x_i).\phi(x_l)}{\sum_{l=1}^{N} h_{x_l}^{C_j}} + \frac{\sum_{p=1}^{N} \sum_{l=1}^{N} h_{x}^{C_j} h_{x_l}^{C_j} \phi(x_p).\phi(x_l)}{\left(\sum_{l=1}^{N} h_{x_l}^{C_j}\right)^2} \qquad [1.12]$$

La distance entre l'objet x_i et le centre u_j dans l'espace de redescription est représentée seulement par des produits cartésiens entre objets. De cette façon, il possible d'introduire une fonction de noyaux pour claculer ce produit cartesésien. La formule [1.12] devient :

$$dist(\phi(x_i), U_j) = K(x_i, x_i) - \frac{\sum_{l=1}^{N} h_{x_l}^{C_j} K(x_i, x_l)}{\sum_{l=1}^{N} h_{x_l}^{C_j}} + \frac{\sum_{p=1}^{N} \sum_{l=1}^{N} h_{x}^{C_j} h_{x_l}^{C_j} K(x_p, x_l)}{\left(\sum_{l=1}^{N} h_{x_l}^{C_j}\right)^2} \qquad [1.13]$$

L'algorithme principale de « SKK » est décrit en pseudo code dans l'annexe 4.

Conclusion

Nous avons décrit dans ce chapitre le cadre général de notre travail basé essentiellement sur le schéma de classification recouvrante. Nous avons défini en premier lieu la notion de classification automatique ainsi que les différents types de schémas de classification automatique à savoir le schéma de classification stricte, le schéma de classification flou et le schéma de classification recouvrante. Le choix d’un type de schéma de classification est proportionnel à la nature du problème et la nature de l’application. Nous avons énuméré en deuxième section quelques problématiques associées au domaine de la classification automatique. Enfin, nous avons présenté la technique des méthodes de noyaux qui permet de projeter explicitement les données dans un espace plus riche où il est plus facile de trouver des séparations entre les données.

Dans le chapitre suivant, nous allons nous limiter au schéma de classification recouvrante. Nous présentons les principales méthodes caractérisant ce type de schéma de classification.

.

Chapitre 2
Méthodes de classification recouvrante

Introduction

La classification recouvrante a pour objectif de déterminer des recouvrements optimaux où un objet peut appartenir à plusieurs classes. Nous nous basons dans ce travail sur la définition de Bertrand qui définit la classification recouvrante comme une collection F de parties non vides de E (E étant l'ensemble des objets à classifier). L'union des parties de F est égale à E telle qu'aucune partie X dans F ne contient une autre partie X' dans F. Pour une telle structure, chaque élément de E appartient à une ou plusieurs parties dans F. Les parties de F sont appelées "classes" ou encore "recouvrements". La dernière condition de cette définition, exclut les méthodes de classification recouvrante stratifiées comme la méthode des « pyramides » et la méthode des « k- hiérarchies faibles ».

En se basant sur cette définition, nous présentons dans ce chapitre un état de l'art des méthodes de classification recouvrante proposées dans la littérature. La première section présente les méthodes recouvrantes basées sur la réallocation dynamique autour des centres mobiles alors que les méthodes recouvrantes basées sur les modèles de mélange de lois seront présentées dans la deuxième section. Enfin, la troisième section présente les méthodes recouvrantes basées sur la théorie des graphes.

2.1. Méthodes recouvrantes basées sur la réallocation dynamique autour des centres mobiles

Le principe de ces méthodes se base sur l'affectation multiple des objets selon leur proximité des prototypes des clusters. Ces prototypes sont mis à jour après chaque étape d'affectation. La différence principale entre les méthodes de réallocation dynamique réside dans la manière de calcul du centre. Nous présentons dans cette section deux méthodes recouvrantes de classification automatique à savoir la méthode « OKM » et la méthode «OKMED»

2.1.1. La méthode « OKM : Overlapping K-Means »

La méthode « OKM », proposée par Cleuziou [Cleuziou 2007], étend la méthode k-moyennes pour chercher des recouvrements optimaux plutôt que des partitions optimales. Etant donné un ensemble d'individus à classifier $X = \{x_i\}_{i=1}^{n}$ où $x_i \in R^d$ représente un vecteur de taille d et n est le nombre des individus. Le but de « OKM » est de trouver les *k* recouvrements $\{\pi_c\}_{c=1}^{k}$ de telle sorte que la fonction objective suivante soit optimisée :

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} \left\| x_i - \varphi(x_i) \right\|^2 \qquad [2.1]$$

A partir de $\{\pi_c\}_{c=1}^{k}$ un ensemble de recouvrement, chaque vecteur x_i appartient à un ou plusieurs clusters de telle sorte que l'union des clusters est égale à l'ensemble de données :

$$U_{c=1}^{k} \pi_c = X \qquad [2.2]$$

La notation $\varphi(x_i)$ désigne l'image de x_i définie par la combinaison des centres des clusters aux quels x_i appartient :

$$\varphi(x_i) = \sum_{A_i} m_c \Big/ \left| A_i \right| \qquad [2.3]$$

Où A_i est l'ensemble des affectations aux différents clusters de l'objet x_i, c'est-à-dire les clusters auquel x_i appartient et m_c est le centre du cluster *c*.

Le critère *J* de la fonction objective généralise le critère des moindres carrés utilisés dans la méthode k-moyennes. Pour minimiser ce critère deux grandes étapes sont exécutées itérativement tant que le critère *J* n'est pas minimisé : La première étape consiste à calculer les centres des clusters alors que la deuxième étape consiste à affecter chaque objet à une ou plusieurs classes selon une fonction d'affectation. La convergence de la méthode est caractérisée par plusieurs critères comme dans la méthode de k-moyennes. Les critères sont : le nombre d'itérations maximales et le seuil minimal d'amélioration de la fonction objective entre deux itérations. L'algorithme principale de la méthode « OKM » est décrit en pseudo code dans l'annexe 1.

En ce qui concerne l'étape des calculs des centres des recouvrements, la méthode « OKM » fait appel à la fonction « *PROTOTYPE* » [annexe1] qui permet de mettre à jour les centres des clusters après chaque étape d'affectation. Cette fonction tient compte des objets qui sont affectés à plusieurs clusters. Elle garantit la minimisation du critère objectif et par la suite la convergence de l'algorithme en définissant un critère général pour calculer le centre d'inertie défini par :

$$m_h^* = \frac{1}{\sum_{x_i \in \pi_h} \alpha_i} \sum_{x_i \in \pi_h} \alpha_i . m_h^i$$

[2.4]

avec m_h^i égale à : $m_h^i = |A_i| * \quad x_i - \sum_{m_c \in A_i / \{m_h\}} m_c$

où m_h désigne le centre de la classe h et α_i est égale à $\frac{1}{A_i}$.

Pour l'étape de l'affectation multiple, la méthode « OKM » utilise la fonction « *ASSIGN* » [annexe1] qui permet de garantir que les objets sont affectés à un ou plusieurs clusters. L'affectation des objets se base sur une heuristique pour minimiser l'espace des affectations possibles. L'heuristique consiste à trier les clusters de plus proche au plus loin puis d'affecter chaque fois les objets selon l'ordre défini. L'affectation est conservée si et seulement si cette affectation permet de minimiser la distance entre l'objet et son image. Si l'affectation n'est pas conservée, la fonction retourne l'affectation précédente et ne continue pas les autres affectations.

2.1.2. La méthode « OKMED: Overlapping K-Medoid »

En se basant sur le même principe que la méthode « OKM» qui étend la méthode k-moyennes, la méthode « OKMED » étend la méthode « PAM : partionning around medoid » [Kaufman and Rousseeuw 1990] pour produire des recouvrements. La différence principale par rapport à la méthode « OKM » est que le centre d'un cluster ne peut appartenir qu'aux objets qui appartiennent à ce cluster. La méthode « OKMED » [Cleuziou 2009] optimise itérativement la distance entre l'objet et son image. La fonction objective à optimiser est la même utilisée dans « OKM ». Elle est définie par la formule [2.1].

La notion d'image utilisée dans « OKMED » est différente de la notion d'image définie dans « OKM ». Elle n'utilise plus des centroids mais elle utilise des medoids. L'image d'un objet x_i dans « OKMED » est définie par l'objet x_j appartenant à X qui minimise la somme des distances avec tous les medoids auxquels x_i appartient :

$$\varphi(x_i) = \arg\min_{x_{j \in X}} \sum_{x_{mc} \in A_i} d\left(x_j, m_c\right) \qquad [2.5]$$

Le critère J de la fonction objective est minimisé itérativement selon deux étapes : La première étape consiste à calculer les medoids des clusters alors que la deuxième étape consiste à affecter chaque objet à une ou plusieurs classes selon une fonction d'affectation. La convergence de la méthode est caractérisée par le nombre d'itérations maximales et le seuil minimal d'amélioration de la fonction objective entre deux itérations. L'algorithme de la méthode « OKMED » qui permet de minimiser itérativement le critère J est décrit dans l'annexe 2.

De la même façon que dans l'algorithme « OKM », l'affectation multiple des objets est effectuée par la fonction « *ASSIGN* ». Son adaptation pour « OKMED » consiste pour chaque x_i, de considérer les medoids avec un ordre spécifique (du plus proche au plus éloigné de x_i)

et d'affecter x_i successivement à la classe correspondante tant que l'inertie $d(x_i, \phi(x_i))$ diminue.

Concernant la mise à jour des prototypes des recouvrements, elle est effectuée par la fonction « *MEDOID* » [annexe2]. Elle joue le même rôle que la fonction « *PROTOTYPE* » dans « OKM » qui consiste en la recherche d'un medoid approprié plutôt que de trouver le medoid optimal pour deux raisons principales : premièrement, parce qu'il est préférable de limiter les évaluations des medoids qui sont très coûteux dans le contexte de classification recouvrante puisqu'ils exigent pour chaque donnée appartenant à la classe , le calcul de son image utilisant le nouveau medoid et deuxièmement, pour éviter le plus possible de choisir les données qui appartiennent à plusieurs autres classes comme medoid pour une classe. Une information appartenant seulement au groupe considéré sera préférée si elle permet d'améliorer le critère objectif [Cleuziou 2009].

Chacune des deux fonctions « *ASSIGN* » et « *MEDOID* » garantissent à chaque itération la minimisation du critère objectif *J* ce qui permet d'améliorer la vitesse de convergence de cet méthode vers la solution optimale sachant que cette solution fait référence à un optimum local du critère objectif qui dépend de l'initialisation des medoids.

2.1.3. Synthèse des méthodes recouvrantes basées sur la réallocation dynamique autour des centres mobiles

Les méthodes recouvrantes basées sur la réallocation dynamique autour des centres mobiles étendent les méthodes classiques de classification stricte pour produire des recouvrements optimaux. Par exemple la méthode « OKM » étend la méthode k-moyennes et la méthode « OKMED » étend la méthode « PAM ». Ces méthodes recouvrantes proposées héritent aussi les avantages et les inconvénients des méthodes principales de classification stricte.

Les limites de la méthode « OKM » consistent tout d'abord dans la détermination du nombre de recouvrements qui est fixé a priori. Cette limite n'est pas posée dans les méthodes non paramétriques où la détermination du nombre de classes est introduite dans le processus de classification tel que la méthode « IOMM » [Heller and Ghahramani 2007]. Malheureusement, l'information du nombre de classes n'est pas facile à prédire dans la pratique. L'une des solutions est de faire appel à un expert qui va estimer ce nombre. L'estimation de l'expert est basée sur ses propres connaissances et dépend des connaissances acquises par cet expert et ne dépend pas de la nature des données traitées. Dans la classification stricte et la classification floue, plusieurs méthodes sont proposées pour déterminer le nombre de partitions ; par contre, pour la classification recouvrante nous n'avons pas trouvé dans la littérature des travaux qui estiment le nombre de recouvrements. Nous proposons dans le troisième chapitre, une extension des méthodes spectrales d'estimation du nombre de partitions pour déterminer le nombre de recouvrements en utilisant la matrice de noyaux.

La deuxième limite dans « OKM » est que les recouvrements finaux obtenus dépendent de l'initialisation aléatoire des centres de recouvrements. La solution optimale retournée par « OKM » est une solution optimale locale et ne converge pas vers un optimum global. La détermination d'une bonne initialisation des recouvrements permet d'améliorer les résultats de classification. Cleuziou [Cleuziou 2007] montre qu'une initialisation des recouvrements en

utilisant les centres optimaux après l'exécution de k-moyennes permet d'améliorer les performances de « OKM ».

Une autre limite de la méthode « OKM » consiste en la création des grandes zones de recouvrements entre clusters ce qui influence négativement sur la précision de cette méthode. En effet la méthode « OKM » utilise la distance euclidienne pour grouper les objets similaires dans le même recouvrement. La même méthode « OKM » est utilisée avec la mesure de distance « I-divergence ». Cette mesure donne des résultats meilleurs que les résultats obtenus avec la distance euclidienne. Le choix d'une métrique pour calculer les distances ou bien la similarité entre les objets influence directement la qualité de la classification. Le point commun entre ces deux mesures de distances est que les distances entre les objets sont calculées dans un espace de petite dimension. Si la structure des données n'est pas sphérique, par exemple ellipsoïdales, ou bien la structure est concentrique, la distance euclidienne ainsi que la distance « I-divergence » ne donnent pas de bon résultat.

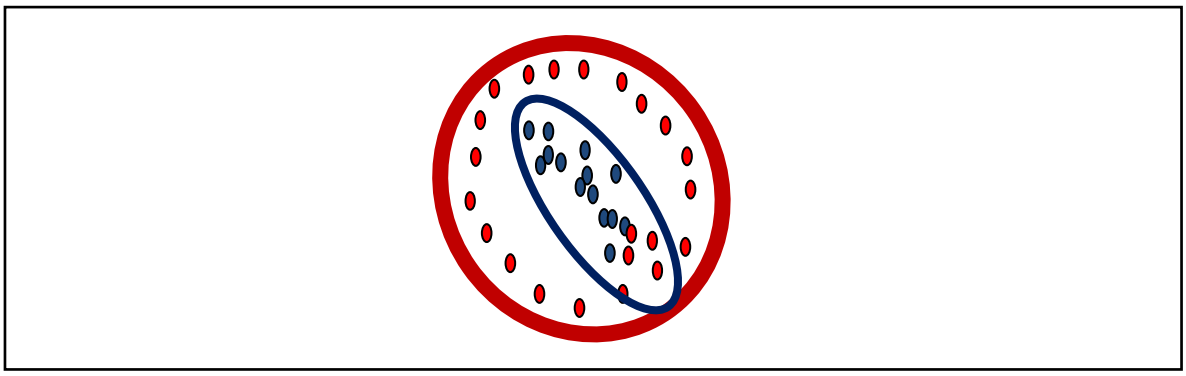


Figure 12. Problématiques de recouvrements concentriques dans la méthode « OKM ».

Une des méthodes de résolution de ce problème dans la classification automatique stricte et la classification automatique floue est de redéfinir les données dans un autre espace de dimension plus grande. La distance entre les objets est calculée ensuite dans ce nouvel espace en utilisant les fonctions de noyaux. L'avantage majeur de ces fonctions est qu'il n'est pas nécessaire de calculer les nouveaux attributs des objets, mais nous possédons seulement leurs données relatives (produit cartésien de leur représentation dans un espace de grande dimension). Nous proposons dans le troisième chapitre une solution utilisant les distances basées sur les fonctions de noyaux pour améliorer les résultats de classification de la méthode « OKM ».

Une autre limite dans « OKM » est la faible capacité de cette méthode à traiter les données éloignées. En fait, les données éloignées affectent le calcul du prototype de chaque classe ce qui perturbe le processus d'affectation multiple de chaque objet à plusieurs classes.

En ce qui concerne la méthode « OKMED », elle possède les mêmes inconvénients que la méthode « OKM » sauf la capacité à traiter les données éloignées. Étant donné que les prototypes des recouvrements dans « OKMED » sont choisis parmi les objets eux-mêmes, les données éloignées n'ont par conséquent aucun effet sur la détermination du prototype de recouvrements et sur l'affectation multiple des objets aux différents clusters.

Malgré les différents inconvénients décelés au niveau des méthodes recouvrantes basées sur la réallocation dynamique autour des centres mobiles, ces méthodes ont l'avantage de représenter un modèle simple, intuitif et facilement applicable sur des problèmes réels. Aussi, l'atout principal de ces méthodes est qu'elles sont déterministes et convergent au moins vers une solution optimale locale.

2.2. Méthodes basées sur les modèles de mélange de lois

Les méthodes recouvrantes basées sur les modèles de mélange de lois étendent les méthodes de classification stricte pour produire des recouvrements optimaux. Les méthodes probabilistes ont prouvé une bonne qualité de classification dans plusieurs domaines d'application de la classification automatique. Nous présentons dans cette section deux méthodes recouvrantes probabilistes à savoir la méthode « MOC » et la méthode « IOMM ».

2.2.1. La méthode « MOC: Model based Overlapping Clustering »

La méthode « MOC » peut être considérée comme la première méthode en classification recouvrantre qui cherche directement des recouvrements optimaux et non des partitions optimales. Elle est implémentée par Banerjee [Banerjee and *al.* 2005b]. Elle généralise les modèles de mélange de lois et les modèles probabilistes pour trouver les recouvrements optimaux. La version de base de « MOC » utilise la maximisation de la vraisemblance à partir d'un mélange de gaussienne avec des variances constantes. La méthode est ensuite généralisée pour utiliser n'importe quelle distribution exponentielle et par la suite l'ouverture de cette méthode à utiliser d'autres distances telle que la « I-divergence ».

La méthode « MOC » est inspirée de la bioinformatique. Elle peut être vue comme une instanciation du modèle « PRM :probabilistic relationel model » permettant de modéliser les relations entre des gènes, des processus et des valeurs d'expressions de ces gènes mesurées sur des puces ADN. Cette instanciation repose sur l'hypothèse que le niveau d'expression d'un gène X_{ij} observé dans une certaine condition expérimentale dépend des processus A_j auxquels le gène participe et de leur niveau d'influence M_i dans cette même condition expérimentale. Le modèle « PRM » ne met aucune restriction ou contrainte quant aux processus auxquels un gène peut participer. La figure 13. illustre l'instanciation du modèle PRM qui montre que l'expression X_{ij} du gêne X_i dans la condition j dépend du niveau d'influence du processus A_{hj} dans la condition j et de la participation M_{ih} du gêne i dans le processus A_h.

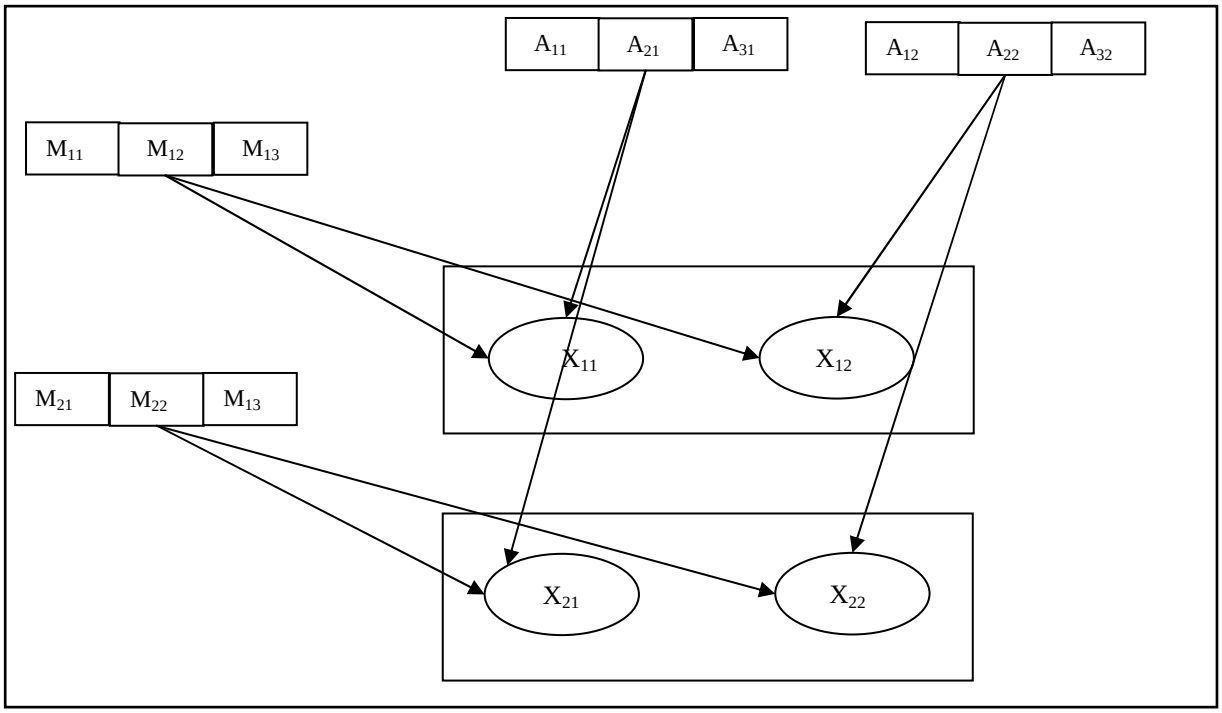


Figure 13. Instanciation du modèle « PRM » avec 2 gènes (données), 2 conditions (dimensions), 3 processus (classes).

.

Le modèle « PRM » utilise trois matrices : la matrice des observations du gène *X* (gênes, expérimentations), matrice d'appartenance d'un gêne à un processus *M* (gênes, processus) et la matrice des activités *A* (processus, conditions) de chaque processus dans chaque condition d'expérimentation. L'hypothèse de base de ce modèle est que l'expression X_{ij} du gêne *i* dans l'expérimentation *j* a une distribution gaussienne avec une variance constante. La moyenne des distributions est égale à la somme des niveaux d'influences A_{hj} du processus *h* auquel le gêne *i* participe. Mathématiquement cette expression peut être écrite comme suit :

$$P(X_{ij} / M_i, A) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(X_{ij} - M_i A_j)^2}{2\sigma^2}\right) \qquad [2.6]$$

Le modèle « PRM » suppose aussi que les paramètres *M* et *A* sont indépendants et par la suite :

$$P(M, A) = P(M)\, P(A) \qquad [2.7]$$

Et que les observations X_{ij} sont indépendantes sachant M_i et A_j.

Le problème de classification sera alors résolu par la recherche des paramètres *M* et *A* maximisant la vraisemblance du modèle. La distribution jointe de *X*,*M* et *A* à optimiser est :

$$P(X, M, A) = P(M, A) \quad P(X / M, A) = P(M)P(A) \quad P(X / M, A) \\ = \prod_{i,h} P(M_{ih}) \quad \prod_{h,j} P(A_{hj}) \quad \prod_{i,j} P(X_{ij} / M_i, A_j) \qquad [2.8]$$

En supposant que les processus A_{hj} sont uniformément distribués, la maximisation du log de vraisemblance est comme suit :

$$\max_{M,A} \log P(X, M, A) \equiv \max_{M,A} \left[\sum_{i,h} \log P(M_{ih}) - \frac{1}{2\sigma^2} \sum_{i,j} \left(X_{ij} - M_i A_j \right)^2 \right] \\ \equiv \min_{M,A} \left[\frac{1}{2\sigma^2} \left\| X - MA \right\|^2 - \log P(M) \right] \qquad [2.9]$$

Donc, pour déterminer la valeur des variables M et A le modèle « PRM » utilise l'approche « EM : expectation maximisation ». L'étape « expectation » permet de chercher la bonne estimation d'appartenance d'un gêne à un processus et l'étape « maximisation » permet de calculer la probabilité d'appartenance d'une gêne à un processus et les niveaux d'influence du processus dans chaque condition [Banerjee and *al.* 2005b].

La méthode « MOC » généralise ce modèle pour produire des recouvrements. Pour adapter le modèle de mélange de lois à la classification recouvrante, cette méthode suppose que chaque observation X_i soit obtenue indépendamment avec une densité de probabilité :

$$P(X_i / \Theta) = \sum_{h=1}^{K} \alpha_h P_h (X_i / \theta_h) \qquad [2.10]$$

Où $\Theta = \{\theta_h\}_{h=1}^{k}$, k est le nombre de composants de mélange (classes), et α_h est le coefficient de mélange qui est égale à $P(M_h)$.

Chaque ligne de la matrice X correspond à un vecteur booléen M_i. La généralisation du modèle « PRM » à la classification recouvrante implique l'existence de plusieurs '1' dans le vecteur booléen M_i qui indique l'appartenance du gène i (objet) à plusieurs processus h (classes). Pour maximiser le critère [2.9], la méthode « MOC » itère entre les 3 paramètres à savoir, la matrice α , la matrice d'appartenance M et la matrice d'activité A pour trouver l'estimation optimale [Banerjee and *al.* 2005b].

En ce qui concerne la mise à jour de α , elle peut être calculée au moment de l'estimation de la matrice M. supposant que π_h est la probabilité a priori de n'importe quel objet pour qu'il appartient au cluster h, la valeur α_{ih} d'un objet i est définie par :

$$\alpha_{ih} = \pi_{hM_{ih}} (1 - \pi_h)^{1-M_{ih}} \qquad [2.11]$$

La valeur de π_h qui maximise la vraisemblance du modèle est égale à:

$$\pi_h = \frac{1}{n} \sum_{i=1}^{n} \{M_{ih} = 1\} \qquad [2.12]$$

Donc la matrice α est calculée à partir des équations [2.11] et [2.12].

Pour l'étape de mise à jour de *M*, elle vise à minimiser la distance entre la matrice *X* et les 2 matrices *M* et *A* :

$$\sum_{i,j} d_{\phi}\left(X_{ij},\left(MA_{ij}\right)\right) \qquad [2.13]$$

Où d_{ϕ} est la divergence de « Bregman » [Banerjee and *al.* 2005a] associée à la densité exponentielle choisie, en particulier la distance euclidienne pour des densités Gaussiennes et la « I-divergence » pour des densités de Poisson. La méthode « MOC » utilise l'algorithme « dynamicM » pour résoudre partiellement le problème de la recherche des composantes binaires optimales d'un vecteur M_i .

Enfin pour la mise à jour de la matrice *A*, elle est facilement réalisée pour la distance euclidienne où il s'agit d'une minimisation de type moindres carrés :

$$A = M^{\oplus} X \qquad [2.14]$$

Où $M^{\oplus}$ désigne la pseudo-inverse de la matrice *M*. alors que pour la distance de « I-divergence », Banerjee s'appuie sur des techniques de factorisation de matrices non-négatives pour aboutir à la règle de mise à jour suivante :

$$A_{hj}^{new} = A_{hj} \frac{\sum_i M_{ih} X_{ij} / (MA)_{ij}}{\sum_i M_{ih}} \qquad [2.15]$$

Les trois étapes de mises à jour de la matrice α **,** la matrice d'appartenance *M* et la matrice d'activité *A* permettent d'améliorer, à chaque itération, la vraisemblance du modèle probabiliste. L'initialisation de ces matrices est effectuée au moyen d'un algorithme de partitionnement tel que k-moyennes.

2.2.2. IOMM: infinite overlapping mixture model

La méthode « IOMM » [Heller and Ghahramani 2007] peut être considérée comme une autre généralisation des modèles de mélange de lois pour la classification recouvrante. Cette méthode utilise le processus « IBP :Idian Buffet Process » pour se dégager de la contrainte du nombre de classes fixé a priori : la détermination de nombre de classe et la détermination des affectations de chaque objet à un ou plusieurs clusters sont réalisées selon ce processus.

Pour adapter les modèles de mélange de lois à la classification recouvrante, Heller a modifié la représentation classique de l'hypothèse qui considère que chaque observation X_i soit obtenue indépendamment avec une densité de probabilité en une nouvelle représentation différente de la présentation de Banerjee qui est utilisée dans la méthode « MOC » (Formule [10]). Heller a défini la densité de probabilité pour qu'un objet soit généré d'un cluster par la formule :

$$P(X_i|\theta) = \sum_{z_i} P(z_i) \prod_{j=1}^{K} P_j\left(X_i \middle| \theta_j\right)^{z_{ij}} \qquad [2.16]$$

Où θ est le paramètre du cluster k et $z_i = [z_{i1},...,z_{ik}]$ est un vecteur binaire indiquant l'appartenance d'un objet à un cluster. La valeur $z_{ik} = 1$ indique que l'objet i appartient au cluster k.

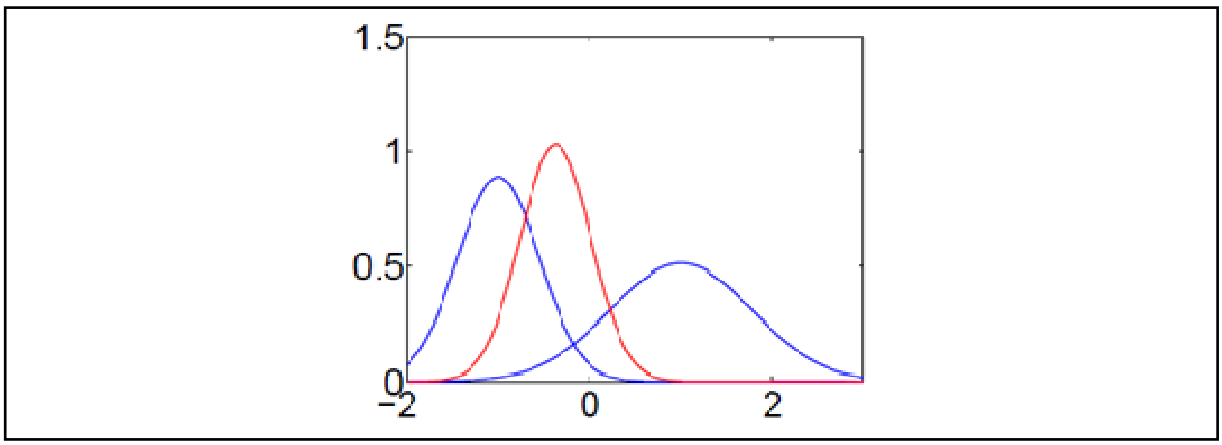


Figure 14. Le produit de deux distributions gaussiennes (en bleu) de paramètre N_1 (-1, 0.2) et N_2 (1,0.6) est une distribution gaussienne (en rouge).

Pour que cette représentation soit bien adaptée à la problématique de classification recouvrante, Heller a enlevé la restriction sur la variable z_{ik} établie par le modèle de mélange de lois dans la classification stricte qui suppose que $\sum_{j=1}^{K} z_{ij} = 1$. Une autre contrainte nécessite d'être résolue pour rendre le problème non paramétrique est celle du nombre de classes. Heller a utilisé la loi beta-binomiale dans le processus « IBP :indian buffet process ». La détermination du nombre de classes est donc proportionnelle à la structure des données. En enlevant la restriction du nombre de classes auxquelles un objet peut appartenir la formule [2.16] devient :

$$P(x_i|z_i,\theta) = \frac{1}{c} \prod_{k} P_k\left(x_i|\theta_k\right)^{z_{ik}} \qquad [2.17]$$

où θ est le paramètre du cluster k, $z_i = [z_{i1},..., z_{ik}]$ est un vecteur binaire indiquant l'appartenance d'un objet à un ou plusieurs clusters et c est une variable de normalisation pour que la probabilité reste dans l'intervalle [0 1]. La figure 11 illustre que le produit de la multiplication de deux distributions gaussiennes est aussi une distribution gaussienne ce qui explique la multiplication des distributions dans la formule [2.17]. Heller suppose aussi que la variable z_{ik} suit une distribution de « Bernoulli » de la forme :

$$z_{ik} \mid \pi_k \approx Bernoulli \ (\pi_k) \qquad [2.18]$$

Où π_k est une variable indiquant la probabilité d'appartenance au cluster *k*. La somme des probabilités $\sum_k \pi_k$ n'est pas égale à 1 vue qu'un objet peut appartenir à plusieurs clusters. Donc la variable π_k est modélisée par une distribution « Beta » de la forme :

$$\pi_k \mid \alpha \approx Beta \ (\frac{\alpha}{k}, 1) \qquad [2.19]$$

où α est une variable pour contrôler le nombre de clusters auxquels un objet peut appartenir. Sachant que le nombre de clusters *k* n'est pas fixé a priori dans la méthode « IOMM », Heller suppose que le nombre de clusters appartient à l'intervalle [1 $+\infty$]. Plus le nombre de clusters est grand, plus la probabilité π_k qu'un objet appartient à un cluster *k* diminue. Heller a défini l'affectation multiple d'un objet à plusieurs clusters, où le nombre de clusters est inconnu, par le processus « IBP » [Griffiths and Ghahramani 2006]. Ce processus consiste à définir une distribution de poissons pour modéliser l'affectation d'un objet à un ensemble infini de clusters.

Pour déterminer les valeurs des différentes variables définies dans le modèle infini de mélange de lois où le nombre de clusters est infini, la méthode « IOMM » utilise l'algorithme « MCMC : Markov Chain Monte Carlo ». Cet algorithme est utilisé comme un algorithme d'inférence dans un modèle « baysien ».

2.2.3. Synthèse des méthodes recouvrantes basées sur les modèles de mélange de lois

En se basant sur le même principe que les méthodes recouvrantes basées sur la réallocation dynamique autour des centres mobiles, les méthodes recouvrantes basées sur les modèles de mélange étendent les méthodes de classification stricte pour produire des recouvrements optimaux. La méthode « MOC » étend la méthode « CEM : Classification EM » pour permettre à un objet d'appartenir à plusieurs classes. L'inconvénient de cette méthode est tout d'abord le nombre de clusters qui est fixé a priori ; Puis la faiblesse de l'hypothèse de base qui n'est pas adaptée à la classification recouvrante. L'hypothèse suppose que chaque objet x_i soit généré à partir d'une seule classe (densité de probabilité d'une seule classe). La notion de chevauchement entre clusters est introduite seulement dans l'étape de maximisation du log de vraisemblance du modèle proposé dans « MOC ». Une étude comparative [Cleuziou 2007] entre la méthode « MOC » et la méthode « OKM » montre des ressemblances entre ces deux méthodes. La différence réside dans le calcul de la définition de l'image qui est une somme dans « MOC » et une moyenne dans « OKM ». La définition de l'image comme une somme est inspirée de la bioinformatique puisque « MOC » combine les processus auxquels une gêne participe. Selon Cleuziou, cette somme n'est pas adéquate sur des bases réelles. Il explique les résultats consistant sur des bases bioinformatiques trouvés par Banerjee par la concentration naturelle de ces données autour de 0. Il montre expérimentalement sur d'autres bases de données que l'utilisation d'une somme dans le calcul de l'image n'est pas juste. Un autre inconvénient dans la méthode « MOC » est la détermination de la densité de probabilité initiale des données et l'initialisation des différentes matrices définies dans cette méthode.

En ce qui concerne la méthode « IOOM » elle résout le problème de la faiblesse de l'hypothèse de base dans « MOC » en se basant sur les modèles multiplicatifs de mélange de lois. Chaque observation x_i est générée à partir de la multiplication de plusieurs classes. Aussi cette méthode se dégage de la contrainte du nombre de classes fixé a priori. Ce nombre est déterminé itérativement dans l'étape d'inférence. L'inconvénient de la méthode « IOOM » est la difficulté de l'initialisation des différentes variables et des différentes distributions de probabilité.

2.3. Méthodes basées sur la théorie des graphes

Les méthodes basées sur cette approche commencent par la construction d'un graphe de similarité. Ensuite à partir de ce graphe, elles cherchent le nombre de cliques maximales. La recherche des cliques maximales dans un graphe de similarité est un problème NP complet.

Parmi ces méthodes, la méthode « POBOC : Pole based overlapping clustering » [Cleuziou and *al.* 2004] qui définit un graphe de similarité contenant des nœuds et des arcs. Un nœud est un objet à classifier et deux nœuds ne sont reliés par un arc que si leur similarité est supérieure à la similarité maximale entre la similarité moyenne du premier nœud avec tous les autres objets et la similarité moyenne du deuxième nœud avec tous les autres objets.

D'autres nouvelles méthodes sont récemment proposées telles que « OCP :Overlapping Cluster Planarity » [Didimo and *al.* 2007] et « Graph-Based Data Clustering with Overlaps » [Fellows and *al.* 2009].

Conclusion

Nous avons présenté dans ce chapitre une étude détaillée des méthodes recouvrantes basées sur la réallocation dynamique autour des centres mobiles et des méthodes recouvrantes basées sur les modèles de mélange de lois. Nous avons aussi brièvement présenté les méthodes basées sur la théorie des graphes pour produire des recouvrements. Nous avons déterminé pour chaque approche les différents avantages et les différentes limites. Dans le chapitre suivant, nous traitons la problématique du nombre de recouvrements et la problématique des classes de forme concentrique en utilisant les fonctions de noyaux.

Chapitre 3
Estimation du nombre de recouvrements et identification des classes de formes complexes

Introduction

Plusieurs limites existent dans les méthodes recouvrantes décrites dans le deuxième chapitre. Nous allons traiter dans ce chapitre, la problématique du nombre de classes et la problématique des classes de formes concentriques. Nous proposons une solution pour traiter ces deux problématiques en se basant sur les méthodes de noyaux. Ces méthodes permettent de projeter les données d'un espace non riche (euclidien) à un espace hilbertien de forte dimensionnalité ou infiniment dimensionnel. En s'inspirant des méthodes « kernel k-means » et « soft kernel k-means » qui étendent la méthode k-moyennes et la méthode « c-means » pour chercher des clusters dans un espace de grande dimension, nous visons à chercher des recouvrements dans cet espace. Nous proposons tout d'abord une estimation du nombre de classes en utilisant la matrice de noyaux. Ensuite nous adaptons la méthode « OKM » pour produire des recouvrements dans un espace de dimension plus grande.

Ce chapitre comprend une première section présentant la solution proposée pour résoudre la problématique du nombre de classes. La deuxième section est consacrée à l'adaptation de « OKM » pour utiliser une distance basée sur les méthodes de noyaux. Enfin, la troisième section présente la nouvelle méthode de classification recouvrante proposée, basée sur les méthodes de noyaux.

3.1. Estimation du nombre de classes

La méthode d'estimation du nombre de classes à partir de la matrice de noyaux est utilisée principalement dans la classification spectrale. Plusieurs travaux estiment le nombre de classes par extraction des valeurs propres de la matrice de noyaux [Girolami 2002]. Le nombre de valeurs propres les plus significatifs déterminent le nombre de clusters. Ces travaux se basent sur le principe suivant : puisque une valeur dans la matrice de noyaux présente le produit scalaire entre deux objets dans l'espace de redescription, cette valeur peut être interprétée comme une mesure de similarité entre ces objets si le noyau utilisé est de

dimensionnalité infinie tel que le noyau RBF. Si les données contiennent des groupes de densité non variés, la matrice va contenir des structures en blocks indiquant la présence de ces groupes dans les données initiales. En déterminant le nombre de valeurs les plus significatives, nous pouvons déterminer le nombre de groupes.

Objets	Attribut 1	Attribut 2	Attribut 3
1	1	2	0
2	1	1	1
3	2	3	0
4	2	2	1
5	3	0	0
6	3	2	1

Objets	1	2	3	4	5	6
1	1	0.6065	0.6065	0.6065	0.1353	0.2865
2	0.6065	1	0.2231	0.6065	0.2231	0.2865
3	0.6065	0.2231	1	0.6065	0.0821	0.4724
4	0.6065	0.6065	0.6065	1	0.2231	0.7788
5	0.1353	0.2231	0.0821	0.2231	1	0.2865
6	0.2865	0.2865	0.4724	0.7788	0.2865	1

Figure 15. Exemple de matrice de noyaux *K(6,6)* construite à partir d'un ensemble de données (6 objets) de dimension 3 (3 attributs) avec un noyau RBF de paramètre $\sigma = 2$.

En utilisant des notations matricielles, nous pouvons représenter ce principe comme suit : à partir de la quantité $1_N^T K\ 1_N$ où 1_N est le vecteur dimensionnel de taille 1xN contenant chacune la valeur 1/N et 1_N^T est le transposé du vecteur précédemment décrit. la décomposition de la matrice *K* en vecteurs propres donnent : $K = UAU^T$. Les colonnes de la matrice U contiennent les vecteurs propres élémentaires u_i et la matrice diagonale A contient les valeurs propres correspondantes notées λ_i.

$$1_N^T\ K\ \ 1_N = 1_N^T \left\{ \sum_{i=1}^{N} \lambda_i u_i u_i^T \right\} 1_N = \sum_{i=1}^{N} \lambda_i [1_N^T u_i]^2 \qquad [3.1]$$

Donc si la matrice *K* de taille *NxN* contient k groupes, il y aura k valeurs propres λ_i qui sont dominants.

Il est à noter que le nombre de valeurs propres significatives est subjectif. C'est pourquoi les travaux [Girolami 2002] qui présentent cette méthode utilisent le terme estimation. Dans les algorithmes de classification automatique qui nécessitent le nombre de clusters a priori, il est important au moins de connaitre un intervalle de ce nombre. Le fait de donner une valeur minimum et une valeur maximum peut améliorer la qualité de la classification. C'est dans ce

cadre que nous utilisons des méthodes de noyaux dans la classification recouvrante pour estimer le nombre de classes.

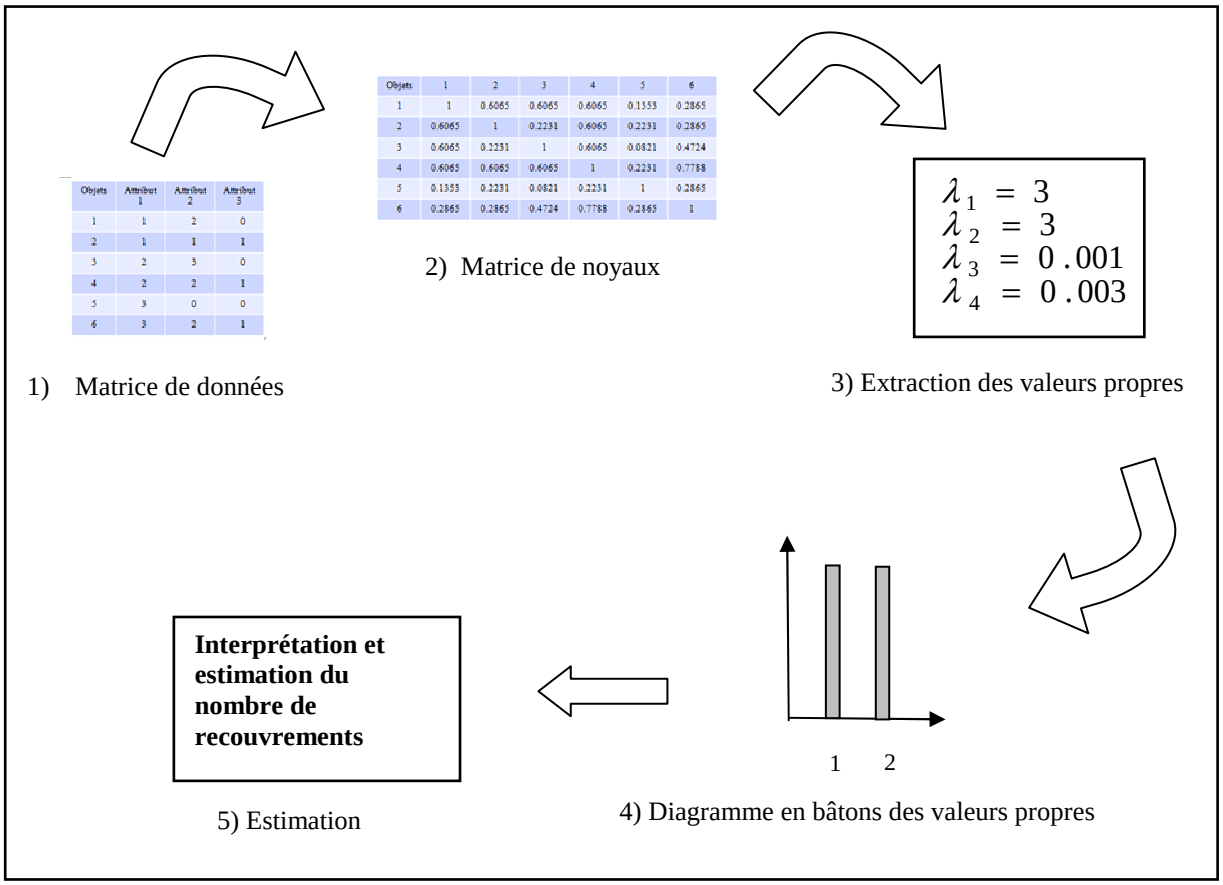


Figure 16. Méthode de détermination du nombre de recouvrements.

En général, dans les données réelles le chevauchement entre clusters n'est pas important sinon il est plus convivial de créer un nouveau cluster pour ces objets. C'est pourquoi le nombre de clusters en général est proche du nombre de partitions. La matrice de noyaux peut être utile dans la classification recouvrante pour construire des hypothèses sur la structure des données. Nous expliquons la méthode d'estimation du nombre de recouvrements à partir de la matrice de noyaux sur ces exemples illustratifs de différentes possibilités de distributions de données, et nous précisons les limites de cette méthode d'estimation.

3.1.1. Premier cas : classes bien séparées et non recouvrantes

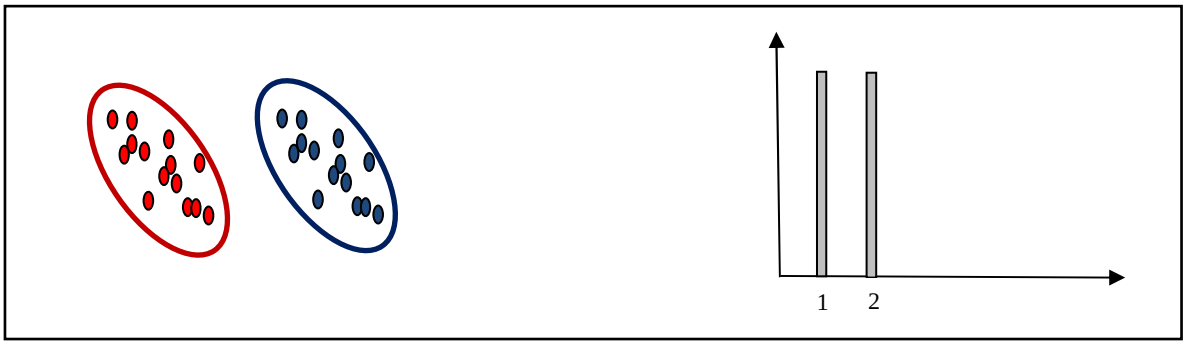


Figure 17. Classes bien séparées et non recouvrantes.

Dans ce cas, étant donnée que les clusters sont bien séparés et de densité équivalente, le diagramme en bâtons des valeurs propres indique bien la présence de deux clusters dans cet ensemble de données. Nous remarquons qu'il n'y a pas de valeurs propres non significatives ce qui peut indiquer l'absence de chevauchement entre les clusters. Donc, ce diagramme en bâtons des valeurs propres de la matrice de noyaux nous indique le nombre de classes et nous indique aussi que les données ne sont pas recouvrantes.

3.1.2. Deuxième cas : classes qui se chevauchent et de densité non variée

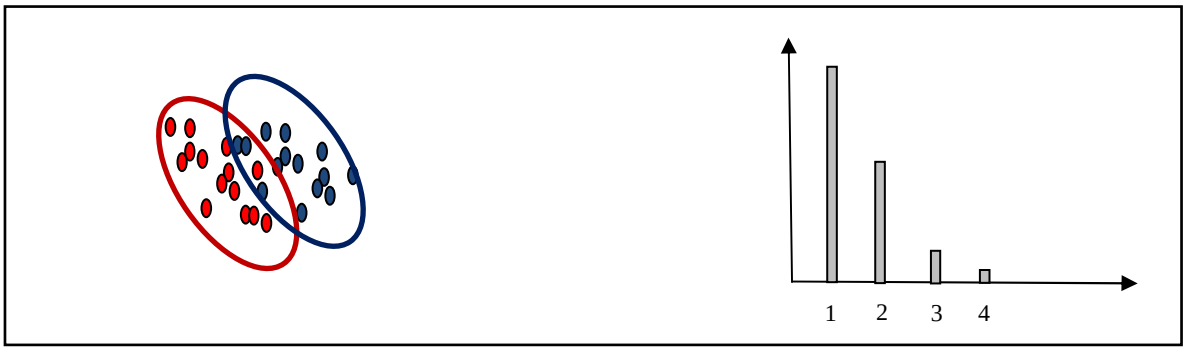


Figure 18. Classes recouvrantes de densité non variée.

Dans ce cas, nous remarquons la présence d'au moins deux valeurs propres significatives avec une autre valeur moins signifiante indiquant la probabilité d'existence d'un troisième cluster. Ici, nous pouvons poser deux hypothèses : la première hypothèse est que le jeu de données n'est pas recouvrant et par conséquent, il est préférable d'ajouter un troisième cluster qui n'est pas facilement séparable des autres clusters. La deuxième hypothèse est que les classes se chevauchent, et par conséquent il est préférable de ne pas ajouter un troisième cluster pour mettre en avantage la possibilité qu'un objet appartienne à plusieurs clusters.

3.1.3. Troisième cas : clusters qui se chevauchent et qui sont concentriques

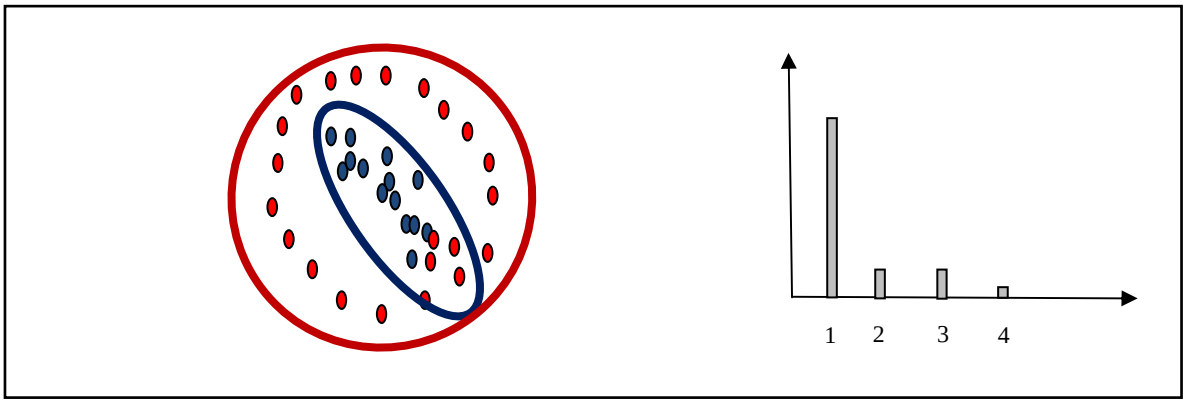


Figure 19. Classes recouvrantes concentriques.

Dans ce cas, les classes sont concentriques. Nous remarquons la présence d'une seule région dense qui va faire apparaitre une seule valeur propre importante. Nous remarquons aussi qu'il y a une probabilité d'existence d'autres clusters. Les clusters concentriques restent toujours parmi les importantes problématiques dans la classification automatique.

3.1.4. Quatrième cas : clusters de densité variée

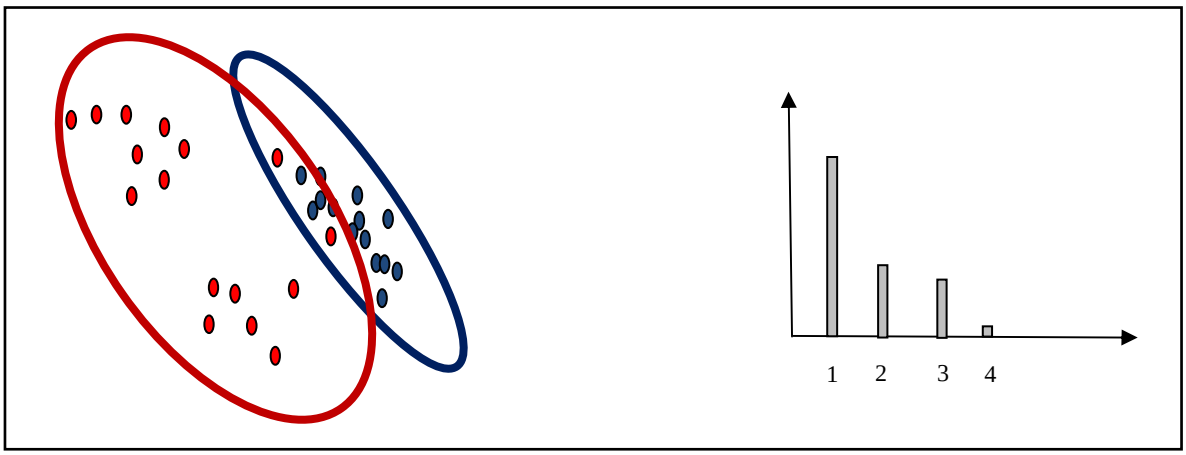


Figure 20. Classes recouvrantes de densité variée.

Si les clusters sont recouvrants et de densités variées, la méthode d'estimation proposée peut être inadéquate étant donné que les densités variées dans le même cluster indiquent une probabilité d'existence d'un autre cluster comme dans le cluster rouge et aussi elle peut indiquer la présence de chevauchement entre les clusters.

3.2. Utilisation des Méthodes de noyaux comme une mesure de distance

Selon le théorème de « Mercer », les méthodes de noyaux peuvent définir directement des mesures de similarité entre les objets. Le produit scalaire entre deux représentations d'objets dans un espace de grande dimension se comporte comme une mesure de similarité : Plus les objets sont similaires plus ce produit est grand. D'autres auteurs, adaptent cette propriété de similarité pour définir une mesure de distance dans l'espace de grande dimension [Satish and Sekhar 2004]. La distance entre deux objets est minimale si les objets se ressemblent. La figure 21. montre géométriquement la notion de distance entre deux représentations d'objets dans un espace de grande dimension.

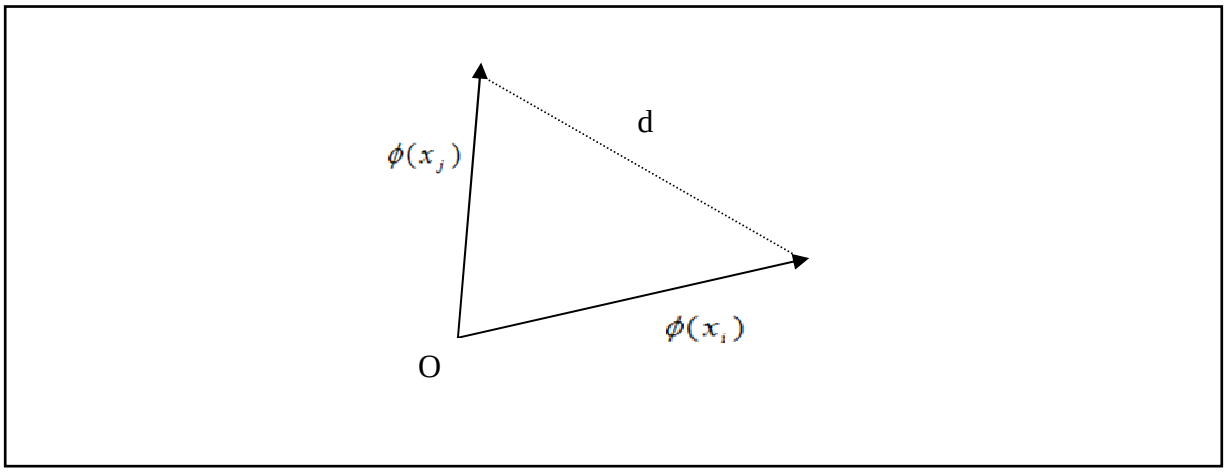


Figure 21. Distance entre deux représentations d'objets dans l'espace de redescription.

La distance dans l'espace de représentation entre deux objets est définie par :

$$d\left(\phi(x_i),\phi(x_j)\right)=\left\|\phi(x_i)-\phi(x_j)\right\|^2 \quad =\left(\phi(x_i)-\phi(x_j)\right)\ \left(\phi(x_i)-\phi(x_j)\right)$$

$$=\langle\phi(x_i)\ \ \phi(x_i)\rangle+\langle\phi(x_j)\ \ \phi(x_j)\rangle-2\langle\phi(x_i)\ \ \phi(x_j)\rangle \qquad [3.2]$$

$$=K(x_i,x_i)+k(x_j,x_j)-2k(x_i,x_j)$$

Si le noyau utilisé est un noyau RBF alors la distance devient :

$$d\left(\phi(x_i),\phi(x_j)\right)=K(x_i,x_i)+k(x_j,x_j)-2k(x_i,x_j)=2-2\exp\left\{-\frac{\left\|x_i-x_j\right\|^2}{\sigma^2}\right\} \qquad [3.3]$$

En se basant alors sur cette définition de distance entre les objets dans un espace fortement dimensionnel, nous transformons la méthode « OKM » pour chercher des recouvrements après la transformation des données de l'espace d'origine à un autre espace de dimension plus grande. Dans « OKM » la fonction objective à maximiser est comme suit :

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} \left\| x_i - image(x_i) \right\|^2 \qquad [3.4]$$

Où $X = \{x_i\}_{i=1}^{n}$ est l'ensemble des données avec $x_i \in R^P$ et $\{\pi_c\}_{c=1}^{k}$ est l'ensemble des recouvrements. Cette fonction minimise la distance entre l'objet x_i et l'image de l'objet. Nous supposons maintenant que les données sont transformées dans un espace de plus grande dimension. L'objectif est donc de minimiser la distance entre l'objet et son image dans cet espace. La formule [3.4] devient:

$$\begin{aligned} J(\{\pi_c\}_{c=1}^{k}) &= \sum_{x_i \in X} \left\| \phi(x_i) - \phi(image(x_i)) \right\|^2 \\ &= \sum_{x_i \in X} [\phi(x_i) * \phi(x_i)] + [\phi(image(x_i)) * \phi(image(x_i))] - 2[\phi(x_i) * \phi(image(x_i))] \end{aligned} \qquad [3.5]$$

Nous remplaçons chaque produit scalaire par la fonction du noyau la formule [3.5] devient:

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} [K(x_i, x_i)] + [K(image(x_i), image(x_i))] - 2[K((x_i), image(x_i))] \qquad [3.6]$$

Si le noyau utilisé est un noyau RBF la formule [3.6] sera minimisée à :

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} \left(2 - 2 \exp\left\{ \frac{-\left\| x_i - image(x_i) \right\|^2}{\sigma^2} \right\}\right) \qquad [3.7]$$

La distance à minimiser itérativement est une distance calculée dans un espace fortement dimensionnel. La projection des données dans cet espace permet de traiter le problème des recouvrements ayant des formes concentriques et ayant des formes ellipsoïdales.

La méthode « OKM » maximise la fonction objective décrite précédemment en itérant deux grande étapes : la première affecte les objets à un ou plusieurs clusters en utilisant la fonction « ASSIGN » [annexe1] et la deuxième étape met à jour les centres en utilisant la fonction « *PROTOTYPE*» [annexe1]. La fonction « *PROTOTYPE*» ne nécessite pas le calcul de distance entre objets donc elle est utilisée comme dans la méthode « OKM ». Par contre la fonction « *ASSIGN* » doit être modifiée pour calculer la distance entre objets et la distance entre un objet et un centre de cluster dans l'espace de redescription. Nous avons définit une autre fonction « *K_ASSIGN* » qui permet d'utiliser des distances de noyaux. Le pseudo de cette fonction est comme suit :

$K_ASSIGN\ (x_i, \{m_c^{(0)}\}_{c=1}^{K}, A_i^{old})$

Entrée :

x_i : *vecteur dans* R^d, $\{m_1,\ldots m_k\}$ *l'ensemble des centres des k clusters et* A_i^{old} *l'ensemble des affectations de l'objet i au différent cluster.*

Sortie :

$A_i^{old} \subset \{m_1,\ldots, m_k\}$ *un sous ensemble des affectations de l'objet* x_i

Etapes :

1) Soit $A_i = \{m^*\}$ *tel que* $m^* = \arg\min_{\{m_c\}_{c=1}^{K}} \|\phi(x_i) - \phi(m_c)\|^2$ *et calculer image* (x_i) *avec les affectations* A_i

2) Chercher le cluster le plus proche suivant

$m' = \arg\min_{\{m_c\}_{c=1}^{K} / A_i} \|\phi(x_i) - \phi(m_c)\|^2$ *et calculer image'* (x_i) *avec les affectations* $A_i \cup \{m'\}$

3) Si $\|\phi(x_i) - \phi(image'(x_i))\|^2 < \|\phi(x_i) - \phi(image(x_i))\|^2$ *alors*

$A_i \longleftarrow \{m'\}$ *et* $image(x_i) = image'(x_i)$ *et retour à l'étape 2*

Sinon calculer $image(x_i)$ *avec* A_i^{old}

Si $\|\phi(x_i) - \phi(image(x_i))\|^2 \leq \|\phi(x_i) - \phi(image^{old}(x_i))\|^2$

retourner A_i

Sinon retourner A_i^{old} .

Figure 22. Algorithme de la fonction « *K*_ASSIGN » utilisant les distances basées sur les méthodes de noyaux

Concernant les critères de convergence, nous avons conservé les mêmes critères utilisés dans « OKM » qui sont le nombre d'itérations maximales et le seuil minimal d'amélioration de la fonction objective entre deux itérations.

3.3. Autocritique de la solution proposée

Nous avons proposé d'utiliser les méthodes de noyaux comme une mesure de distance pour chercher des recouvrements dans un espace fortement dimensionnel. Cette solution nous a permis de nous libérer de la contrainte des clusters ayant des formes acomplexes et aussi la perspective d'appliquer la classification sur des données structurées non vectorielles.

Le point critique dans la méthode proposée précédemment est que l'image (centre de gravité des recouvrements auxquels un objet appartient) des objets et les centres des clusters sont calculés dans l'espace d'origine alors que seulement la distance entre ces objets est calculée dans l'espace de redescription. La formule [3.5] montre que l'image d'un objet est calculée dans l'espace d'origine et ensuite elle est projetée dans l'espace de redescription. C'est seulement la distance entre l'objet et l'image qui est calculée dans le nouvel espace. Aussi pour la fonction d'affectation « *k_ASSIGN* », les centres les plus proches sont calculés dans l'espace d'origine.

La formule du choix du cluster le plus proche de chaque objet est calculée dans l'espace de redescription comme suit:

$$m^* = \arg\min_{\{m_c\}_{c=1}^{K}} \left\| \phi(x_i) - \phi(m_c) \right\|^2 \qquad [3.8]$$

Donc le centre m_c du cluster est calculé dans l'espace d'origine et ensuite il est projeté dans l'espace de redescription, pour mesurer sa distance avec la représentation de l'objet.

La solution qui parait la plus adéquate est de proposer un algorithme qui cherche des recouvrements dans l'espace de redescription : le calcul d'images des objets, le calcul des centres des recouvrements et le calcul des différentes distances s'effectuent dans un espace de forte dimensionnalité. Le problème est que le principe de l'astuce de noyau ne peut être appliqué que dans un algorithme qui ne dépend que du produit scalaire entre deux vecteurs d'entrée.

3.4. Solution proposée « KOKM : Kernel Overlapping K-Means »

Nous proposons dans cette section une méthode désignée « KOKM » qui combine les avantages de la méthode « OKM » pour produire des recouvrements optimaux et les avantages de la méthode « kernel k-means » qui cherche des classes dans un espace fortement dimensionnel. « KOKM » maximise itérativement la distance entre l'objet et sa représentation dans l'espace de redescription.

3.4.1. Critère objectif de « KOKM »

Etant donnée l'ensemble des vecteurs $X = \{x_i\}_{i=1}^{N}$ avec $x_i \in R^d$, N est le nombre des objets, *K* est le nombre de recouvrements et $\phi(x_i)$ est la représentation de l'objet x_i par une transformation non linéaire $\phi : x_i \mapsto \phi(x_i) \in F$ dans un espace de dimension plus grande. Le but de « KOKM » est de trouver les *k* recouvrements $\{\pi_c\}_{c=1}^{k}$ de telle sorte qu'on minimise, dans l'espace de redescription *F*, la distance entre la représentation de l'objet et l'image de la représentation de cet objet.

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} \left\| \phi(x_i) - image\ (\phi(x_i)) \right\|^2 \qquad [3.9]$$

Chaque objet x_i appartient à un ou plusieurs clusters de telle sorte que l'union des clusters est égale à l'ensemble de données : $U_{c=1}^{k}\pi_c = X$

L'image est définie par le centre de gravité des représentations des centres des clusters auquel un objet appartient :

$$image\ (\phi(x_i)) = \frac{\sum_{k=1}^{K} P_{ik} * m_k}{\sum_{k=1}^{K} P_{ik}} \qquad [3.10]$$

Où $P_{ik} \in \{0,1\}$ est une variable binaire indiquant l'appartenance de l'objet au cluster k, et m_k est le centre du cluster k dans l'espace de redescription *F*. le centre est défini par le centre de gravité des objets qui appartiennent au cluster *k* pondérer par le poids de chaque objet selon le nombre d'affectation de cet objet. La section 3.4.3. Présente en détail la méthode de calcul du centre dans l'espace de redescription.

La fonction objective J dans la formule [3.9] devient donc :

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} \left\| \phi(x_i) - \frac{\sum_{k=1}^{K} P_{ik} * m_k}{\sum_{k=1}^{K} P_{ik}} \right\|^2 \qquad [3.11]$$

Le critère J de la fonction objective définie précédemment généralise le critère des moindres carrés utilisés dans l'algorithme « kernel k-means ». Si l'objet appartient à un seul cluster nous minimisons la distance entre la représentation de l'objet et le centre de cluster.

3.4.2. Algorithme de classification de « KOKM »

Pour maximiser la fonction objective *J*, la méthode « KOKM » affecte à chaque itération les représentations des objets dans l'espace *F* à un ou plusieurs clusters puis elle calcule de nouveau le critère *J* de la fonction objective. Si ce critère s'améliore d'une itération à une autre, les objets sont réaffectés aux clusters les plus proches jusqu'à l'optimisation de ce critère. Les conditions d'arrêt de « KOKM » sont le nombre d'itération maximale et l'amélioration minimale dans la fonction objective d'une itération à une autre.

L'algorithme général de « KOKM » peut être décrit en pseudo-code comme suit :

KOKM $(X, k, t_{\max}, \varepsilon)$

Entrée :
-X :un ensemble de vecteurs dans l'espace R^d :les individus à classifier
-t_{max} : nombre maximum d'itérations
- ε : seuil d'amélioration minimale pour continuer une autre itération
k : nombre de clusters

Sorties :
$\{\pi_c\}_{c=1}^{k}$ *:les recouvrements finaux retournés par l'algorithme*

Etapes :

1) *Initialiser aléatoirement les clusters par des centres arbitraires* $\{m_c^{(0)}\}_{c=1}^{K}$
2) *Pour chaque objet x_i dans X calculer les affectations de la représentation $\phi(x_i)$ aux différents clusters* $A_i^{(0)} = AFFECT\ (\phi(x_i), \{m_c^{(0)}\}_{c=1}^{K})$ *et dériver le recouvrement initial tel que* $\{\pi_c\}_{c=1}^{k}$ *et la valeur initiale de la fonction objective* $J(\{\pi_c^{(0)}\})$.
3) *t=0 // t est le numéro de l'itération*
4) *Pour chaque objet x_i dans X calculer les affectations* $A_i^{(t+1)} = AFFECT\ (\phi(x_i), \{m_c^{(t+1)}\}_{c=1}^{K}, A_i^{(t)})$ *et dériver le nouveau recouvrement* $\{\pi_c^{t+1}\}_{c=1}^{K}$

   Si *(t_{max} > t) ou ($J(\{\pi_c^{(t)}\}) - J(\{\pi_c^{(t+1)}\}) > \varepsilon$), mettre t=t+1et retourner à l'étape 4*

   Sinon stop et retourné la distribution actuelle $\{\pi_c^{t+1}\}_{c=1}^{K}$

Figure 23. Algorithme principal de la méthode « KOKM ».

3.4.3. Le calcul des centres des recouvrements dans « KOKM »

Contrairement à la méthode proposée dans la section 3.2., le centre du cluster est calculé dans l'espace de redescription *F*. Nous avons défini le centre m_k d'un recouvrement *k* comme suit :

$$m_k = \frac{\sum_{j=1}^{N} P_{jk} * \phi(x_j) * (1/A_j^2)}{\sum_{j=1}^{N} P_{jk} * (1/A_j^2)} \qquad [3.12]$$

Où $P_{jk} \in \{0,1\}$ est une variable binaire indiquant l'appartenance de l'objet j au cluster k *et* $1/A_j^{\,2}$ est une pondération du poids de l'objet j dans le calcul du centre du cluster. Le poids est pondéré par le nombre d'affectation A_j de cet objet aux différents clusters. Le poids est égal à 1 si l'objet appartient à un seul cluster. Graphiquement, le centre m_k présente le centre de gravité pondéré des représentations des objets qui appartiennent au cluster *k*. De cette façon, le centre est donc calculé dans l'espace de représentations *F*.

Pour simplifier la formule [3.12], nous considérons C_k le poids associé aux objets appartenant au cluster *k*. C_k est défini par : $C_k = \sum_{j=1}^{N} P_{jk} * (1/A_j^{\,2})$. En remplaçant cette expression dans la formule [3.12] nous obtenons :

$$m_k = \frac{\sum_{j=1}^{N} P_{jk} * \phi(x_j) * (1/A_j^{\,2})}{C_k} \qquad [3.13]$$

Nous remarquons dans cette formule que nous ne pouvons pas calculer directement le centre du cluster dans l'espace *F* car on ne peut pas calculer directement la représentation $\phi(x_j)$ de l'objet x_j. Les méthodes de noyaux permettent seulement le calcul d'un produit scalaire entre les représentations des objets. Comme solution, nous adoptons la même approche utilisée dans l'algorithme « kernel-kmeans » qui consiste à introduire le calcul du centre dans le calcul de la fonction objective. Donc, la fonction objective est transformée comme suit :

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} \left\| \phi(x_i) - \frac{\sum_{k=1}^{K} P_{ik} * m_k}{\sum_{k=1}^{K} P_{ik}} \right\|^2 \qquad [3.14]$$

$$J(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in X} \left\| \phi(x_i) - \frac{\sum_{k=1}^{K} P_{ik} * \frac{\sum_{j=1}^{N} P_{jk} * \phi(x_j) * (1/A_j^{\,2})}{C_k}}{\sum_{k=1}^{K} P_{ik}} \right\|^2$$

Nous présentons dans la section 2.4 le développement de cette formule pour calculer le critère *J* de la fonction objective.

3.4.4. L'affectation des objets à un ou plusieurs clusters

A chaque itération, nous devons décider de l'affectation d'un objet à un ou plusieurs clusters. Chaque objet à 2^k affectations possibles. Pour minimiser la complexité de l'algorithme, nous avons utilisé la même heuristique utilisée dans la méthode « OKM ». Elle consiste à affecter itérativement l'objet au cluster le plus proche tant que la distance entre l'objet et son image diminue. Dans l'espace de redescription *F*, cette distance est définie par :

$$\left\|\phi(x_i) - image\ (\phi(x_i))\right\|^2 = \left\|\phi(x_i) - \frac{\sum_{k=1}^{K} P_{ik} * m_k}{\sum_{k=1}^{K} P_{ik}}\right\|^2 \qquad [3.15]$$

Pour faciliter les écritures nous supposons que L_i présente le nombre d'affectation de chaque objet aux différents clusters défini par $L_i = \sum_{k=1}^{K} P_{ik}$ .En remplaçant cette expression dans la formule [3.15], elle devient :

$$\left\|\phi(x_i) - image\,(\phi(x_i))\right\|^2 = \left\|\phi(x_i) - \frac{\sum_{k=1}^{K} P_{ik} * m_k}{L_i}\right\|^2 \qquad [3.16]$$

En remplaçant aussi le centre m_k par sa formule nous obtenons :

$$\left\|\phi(x_i) - image\,(\phi(x_i))\right\|^2 = \left\|\phi(x_i) - \tfrac{1}{L_i}\sum_{k=1}^{K}[P_{ik} * \tfrac{1}{C_k}\sum_{j=1}^{N}(P_{jk} * (1/A_j{}^2) * \phi(x_j))]\right\|^2 \qquad [3.17]$$

$$\left\|\phi(x_i) - image\ (\phi(x_i))\right\|^2 = [\phi(x_i) * \phi(x_i) - \tfrac{2}{L_i}\sum_{k=1}^{K}\sum_{j=1}^{N}[P_{ik} * \tfrac{1}{C_k} * P_{jk} * (1/A_j{}^2) * \phi(x_i) * \phi(x_j)] +$$
$$\tfrac{1}{L_i{}^2}\sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{t=1}^{K}\sum_{g=1}^{N}[P_{ik} * \tfrac{1}{C_k} * P_{jk} * P_{it} * \tfrac{1}{C_t} * P_{gt} * (1/A_j{}^2) * (1/A_g{}^2) * \phi(x_j) * \phi(x_g)]$$

En remplaçant les produits scalaires obtenus dans la formule [3.17] par la fonction *K* du noyau nous obtenons :

$$\left\|\phi(x_i)-image\ (\phi(x_i))\right\|^2 = [\mathrm{K}(x_i,x_i)-\tfrac{2}{L_i}\sum_{k=1}^{k}\sum_{j=1}^{N}[P_{ik}*\tfrac{1}{C_k}*P_{jk}*(1/A_j^{\,2})*\mathrm{K}(x_i,x_j)]+$$
$$\tfrac{1}{L_i^{\,2}}\sum_{k=1}^{k}\sum_{j=1}^{N}\sum_{t=1}^{k}\sum_{g=1}^{N}[P_{ik}*\tfrac{1}{C_k}*P_{jk}*P_{it}*\tfrac{1}{C_t}*P_{gt}*(1/A_j^{\,2})*(1/A_g^{\,2})*\mathrm{K}(x_j,x_g)] \qquad [3.18]$$

Donc, à partir de cette formule, nous pouvons calculer la distance entre la représentation de l'objet et l'image de la représentation de cet objet en garantissant que le calcul du centre et le calcul de l'image sont réalisés dans l'espace de redescription *F*.

La réalisation de l'heuristique définie dans cette section nécessite aussi à chaque itération de déterminer le cluster le plus proche de l'objet. Le cluster le plus proche d'un objet x_i dans l'espace de redescription *F* est définie par :

$$m_c^{\,*} = \underset{\{m_c\}_{c=1}^{K}}{\arg\min}\ \left\|\phi(x_i)-m_c\right\|^2 \qquad [3.19]$$

En remplaçant le centre m_c par sa formule nous obtenons :

$$m_c^{\,*} = \underset{\{m_c\}_{c=1}^{K}}{\arg\min}\left\|\phi(x_i)-\frac{\sum_{j=1}^{N}P_{jk}*\phi(x_j)*(1/A_j^{\,2})}{C_k}\right\|^2 \qquad [3.20]$$

Pour que nous puissions calculer cette formule en utilisant les méthodes de noyaux nous la développons comme suit :

$$m_c^{\,*} = \underset{\{m_c\}_{c=1}^{K}}{\arg\min}\left\{\begin{array}{l}\phi(x_i)*\phi(x_i)-\tfrac{2}{C_k}\sum_{j=1}^{N}P_{jk}*(1/A_j^{\,3})*\phi(x_i)*\phi(x_j)+\\ \frac{1}{C_k^{\,2}}\sum_{j=1}^{N}\sum_{l=1}^{L}P_{jk}*P_{lk}*(1/A_j^{\,3})*(1/A_l^{\,3})*\phi(x_j)*\phi(x_l)\end{array}\right\} \qquad [3.21]$$

En remplaçant les produits cartésiens des représentations des objets par la fonction du noyau nous obtenons :

$$m_c^{\,*} = \underset{\{m_c\}_{c=1}^{K}}{\arg\min}\left\{\begin{array}{l}K(x_i,x_i)-\tfrac{2}{C_k}\sum_{j=1}^{N}P_{jk}*(1/A_j^{\,3})*K(x_i,x_j)+\\ \frac{1}{C_k^{\,2}}\sum_{j=1}^{N}\sum_{l=1}^{L}P_{jk}*P_{lk}*(1/A_j^{\,3})*(1/A_l^{\,3})*K(x_j,x_l)\end{array}\right\} \qquad [3.22]$$

Nous pouvons résumer l'heuristique utilisée dans l'algorithme « KOKM » pour affecter les objets à un ou plusieurs clusters par le pseudo code de la procédure « *AFFECT*» suivante :

$AFFECT \quad (x_i, \{m_c^{(0)}\}_{c=1}^{K}, A_i^{old})$

<u>*Entrée*</u> *:*

x_i *: vecteur dans* R^d, $\{m_1,\ldots m_k\}$ *l'ensemble des centres des K clusters et* A_i^{old} *l'ensemble des affectations de l'objet i au différent cluster.*

<u>*Sortie*</u> *:*

$A_i \subset \{m_1,\ldots, m_k\}$ un sous ensemble des affectations de l'objet x_i

<u>*Etapes :*</u>

1) *Soit* $A_i = \{m^*\}$ *tel que* $m^* = \arg\min_{\{m_c\}_{c=1}^{K}} \|\phi(x_i) - m_c\|^2$ *, calculer la distance entre la représentation de l'objet et son image* $\|\phi(x_i) - image\ (\phi(x_i))\|^2$ *avec les affectations* A_i

2) *Chercher le cluster le plus proche suivant*
$m' = \arg\min_{\{m_c\}_{c=1}^{K} / A_i} \|\phi(x_i) - m_c\|^2$ et *calculer la distance entre la représentation de l'objet et son image* $\|\phi(x_i) - image'(\phi(x_i))\|^2$ *avec les affectations* $A_i \cup \{m'\}$

3) *Si* $\|\phi(x_i) - image'(\phi(x_i))\|^2 < \|\phi(x_i) - image'(\phi(x_i))\|^2$ *alors* $A_i \longleftarrow \{m'\}$ *et retour à l'étape 2*

Sinon calculer $\|\phi(x_i) - image^{old}(\phi(x_i))\|^2$ *avec* A_i^{old}

Si $\|\phi(x_i) - image\ (\phi(x_i))\|^2 \leq \|\phi(x_i) - image^{old}(\phi(x_i))\|^2$

retourner A_i

Sinon retourner A_i^{old} .

Figure 24. Algorithme de la fonction « *AFFECT* ».

3.4.5. Exemple illustratif d'exécution de « KOKM »

Nous présentons dans cette section un exemple d'exécution de l'algorithme « KOKM » sur un ensemble de données comportant deux dimensions. Nous présentons seulement l'étape d'initialisation pour expliquer l'affectation multiple dans « KOKM » à un ou plusieurs clusters. Graphiquement, l'ensemble de données est schématisé sur la figure 25.

N° objets	Attribut 1	Attribut 2
1	1	1
2	1	2
3	2	1
4	2	2
5	2	3
6	3	1
7	3	2
8	3	3
9	4	3
10	4	5
11	5	3
12	5	4
13	5	5
14	6	3
15	6	4
16	6	5
17	7	3
18	7	4
19	7	5

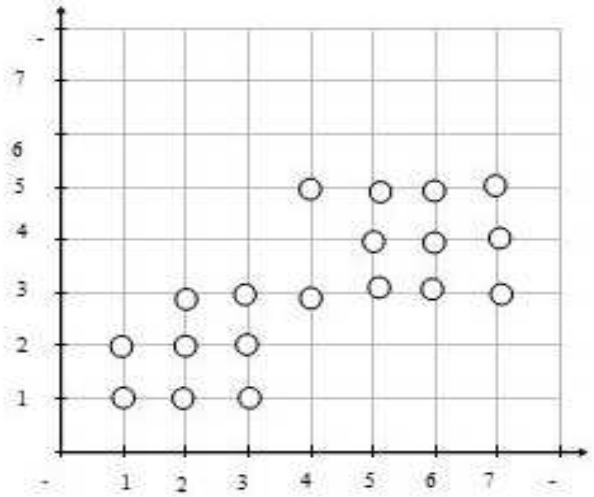


Figure 25. Distribution des objets test sur les deux dimensions définies.

Supposant que le nombre de recouvrements est deux recouvrements : La première étape consiste à initialiser aléatoirement les centres des recouvrements. Une bonne initialisation

influence positivement la qualité de la classification. Nous pouvons par exemple initialiser les centres des clusters par des simulations de la méthode k-moyennes. Nous supposons aussi que le noyau utilisé est le noyau RBF de paramètre $\sigma = 2$. Le noyau RBF consiste donc en cette formule :

$$K(x_i, x_j) = \exp\left\{ \frac{-\sum_{l=1}^{d} (x_{il} - x_{jl})^2}{2^2} \right\} \qquad [3.23]$$

Avec d est la dimension des données qui est égale à 2 dans notre exemple.

Nous supposons dans cet exemple que l'objet n°3 est le centre du premier cluster et l'objet n°16 est le centre du deuxième cluster. Nous affectons maintenant les autres objets itérativement selon l'heuristique « *AFFECT* ». Par exemple l'affectation de l'objet n° 1 consiste tout d'abord à calculer le centre du cluster le plus proche de l'objet n°1 :

$$m^* = \arg\min_{\{m_c\}_{c=1}^{2}} \left\| \phi(x_1) - m_c \right\|^2$$

La distance entre l'objet x_1 et le centre du premier cluster dans l'espace de redescription F est :

$$d(x_1,m_1) = K(x_1,x_1) - \frac{2}{C_1}\sum_{j=1}^{19} P_{j1} * (1/A_j^3) * K(x_1,x_j) + \frac{1}{C_1^2}\sum_{j=1}^{19}\sum_{l=1}^{19} P_{j1} * P_{l1} * (1/A_j^3) * (1/A_l^3) * K(x_j,x_l)$$

$$d(x_1,m_1) = 1 - \frac{2}{1}\left(1*(1/1^3)*\exp(-0.25)\right) + \frac{1}{1^2}\left(1*1*(1/1^3)*\exp(-0)\right) = 2 - 2*(0.77888) = 0.442$$

La distance entre l'objet x_1 et le centre du deuxième cluster est :

$$d(x_1,m_2) = K(x_1,x_1) - \frac{2}{C_2}\sum_{j=1}^{19} P_{j2} * (1/A_j^3) * K(x_1,x_j) + \frac{1}{C_1^2}\sum_{j=1}^{19}\sum_{l=1}^{19} P_{j2} * P_{l2} * (1/A_j^3) * (1/A_l^3) * K(x_j,x_l)$$

$$\begin{aligned} d(x_1,m_2) &= 1 - \frac{2}{1}\left(1*(1/1^3)*\exp(-10.25)\right) + \frac{1}{1^2}\left(1*1*(1/1^3)*\exp(-0)\right) \\ &= 2 - 2*(0.00003535) = 1.999 \end{aligned}$$

Nous constatons que le cluster n°1 est le cluster le plus proche de x_1. Nous affectons donc l'objet x_1 au premier cluster et nous calculons la distance entre sa représentation et son image dans l'espace de redescription.

$$\left\|\phi(x_1) - image\ \ (\phi(x_1))\right\|^2 = [\mathrm{K}(x_1, x_1) - \tfrac{2}{L_i}\sum_{k=1}^{K}\sum_{j=1}^{N}[P_{1k} * \tfrac{1}{C_k} * P_{jk} * (1/A_j^{\ 3}) * \mathrm{K}(x_1, x_j)] +$$
$$\tfrac{1}{L_i^{\ 2}}\sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{t=1}^{K}\sum_{g=1}^{N}[P_{1k} * \tfrac{1}{C_k} * P_{jk} * P_{it} * \tfrac{1}{C_t} * P_{gt} * (1/A_j^{\ 3}) * (1/A_g^{\ 3}) * \mathrm{K}(x_j, x_g)]$$

$$\left\|\phi(x_i) - image\ \ (\phi(x_i))\right\|^2 = [1 - \frac{2}{1}\left((1*(1/1)*(1/1^3)*\exp(-0.25)\right) + \frac{1}{1^2}\left(1*1*(1/1^3)*\exp(-0)\right)]$$
$$\left\|\phi(x_i) - image\ \ (\phi(x_i))\right\|^2 = 0.442$$

Nous affectons ensuite l'objet x_1 au deuxième cluster et nous calculons de nouveau sa distance par rapport à son image avec les nouvelles affectations.

$$\left\|\phi(x_i) - image\ \ (\phi(x_i))\right\|^2 = [1 - \frac{2}{2}\left\{\left((1*(1/1)*(1/1^3)*\exp(-0.25)\right) + \left(1*(1/1^3)*\exp(-10.25)\right)\right\}$$
$$+ \frac{1}{2^2}\left(1*1*(1/1^3)*\exp(-8)\right) + \left(1*1*(1/1^3)*\exp(-0)\right) + \left(1*1*(1/1^3)*\exp(-0)\right)] = 0.7212$$

Nous remarquons donc quand nous affectons l'objet x_1 au deuxième cluster la distance augmente. Donc nous nous arrêtons et nous retournons l'affectation précédente. L'objet x_1 donc appartient à un seul cluster qui est le premier cluster. Nous terminons ensuite les affectations de chaque objet de la même façon que l'objet x_1 et nous dérivons la valeur J de la fonction objective.

Conclusion

Nous avons proposé dans ce chapitre une estimation du nombre de recouvrements pour traiter le problème du nombre de classes et nous avons aussi proposé une reformulation de la méthode « OKM » pour chercher des recouvrements dans un espace de dimension plus grande que l'espace d'origine. Les deux solutions proposées se basent sur les méthodes de noyaux qui sont de plus en plus utilisées dans la classification automatique. Nous avons aussi proposé une méthode de classification recouvrante désignée sous l'appellation « KOKM » qui combine les avantages de la méthode « OKM » pour produire des recouvrements et les avantages de la méthode « kernel-kmeans » pour chercher des classes de formes complexes. Dans le chapitre suivant, nous validons par des expérimentations ces solutions proposées.

Chapitre 4
Evaluation empirique des solutions proposées

Introduction

Nous avons proposé dans le troisième chapitre un cadre théorique de l'utilisation des méthodes de noyaux comme une métrique de distance dans la méthode « OKM ». Ces méthodes de noyaux peuvent être utilisées aussi pour estimer le nombre de recouvrements. Aussi,nous avons proposé aussi une nouvelle méthode de classification recouvrante dans un espace de forte dimensionnalité en utilisant aussi ces méthodes de noyaux. Nous présentons dans ce chapitre, des expérimentations sur des datasets connus dans le domaine de classification pour valider empiriquement les méthodes proposées.

Ce chapitre comporte quatre sections : la première présente la méthode et les mesures de validation utilisées pour comparer les méthodes proposées avec d'autres méthodes dans la littérature. La deuxième section présente les premières expérimentations sur le dataset « Iris ». Ensuite, la troisième section décrit les expérimentations sur le dataset « Ionosphère » qui contient des clusters de formes concentriques. La quatrième section présente les expérimentations sur le dataset « Eachmovie » qui contient des clusters qui se chevauchent.

4.1. Méthode et Mesures de validation utilisées

Nous présentons dans cette section les mesures de distance utilisées dans la méthode « OKM » pour les comparer avec la métrique des fonctions de noyaux que nous avons proposés. Nous présentons aussi la méthode et les mesures d'évaluation que nous avons utilisées pour prouver la pertinence des recouvrements générés par « KOKM » par rapport à « OKM ».

4.1.1. Distances utilisées dans « OKM »

Nous avons proposé dans le troisième chapitre un cadre théorique de l'utilisation des fonctions de noyaux comme des mesures de distance dans l'algorithme « OKM ». Pour évaluer cette mesure, nous comparons les résultats obtenus par l'algorithme « OKM » en utilisant deux autres distances à savoir la distance euclidienne et la distance de « I-divergence ».

La distance Euclidienne entre 2 vecteurs x et y de dimension p dans un espace cartésien $\Re^p$ est définie par :

$$d(x,y) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2} \qquad [4.1]$$

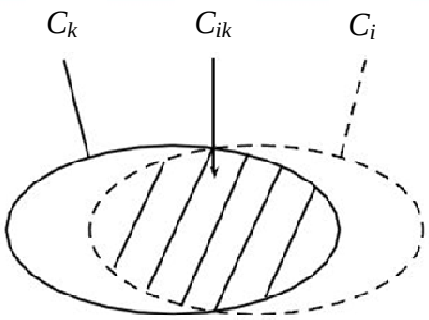
La distance de « I-divergence : Information divergence » connue aussi sous le nom de « divergence de Kullback-Leibler » ou « divergence K-L » ou encore « Entropie relative » est définie par [Banerjee and *al.* 2005a] :

$$d(x,y) = \sum_{i=1}^{p} x_i * \frac{\log(x_i)/\log(y_i)}{\log(2)} - x_i + y_i \qquad [4.2]$$

Bien qu'elle porte le nom de distance, la mesure d'entropie relative de « Kullback-Liebler » n'est pas une métrique. Elle n'est pas symétrique et ne satisfait pas l'inégalité du triangle.

4.1.2. Méthode de validation

Plusieurs méthodes et mesures sont proposées dans la littérature pour valider les algorithmes de classification automatique. Nous avons choisi d'utiliser la même méthode et les mêmes mesures utilisées dans la méthode « OKM » proposé par Cleuziou et dans la méthode « MOC » proposé par Banerjée. Ces deux auteurs valident leurs méthodes sur une base où les classes sont connues. Ils mesurent ensuite la précision « precision », le rappel « recall », et la « F-measure » sur les groupes créés.



$$precision(i,k) = \frac{N_{ik}}{N_k}$$

$$rappel(i,k) = \frac{N_{ik}}{N_{Ci}}$$

Figure 26. Illustration de la mesure du rappel et de la mesure de précision.

Cette méthode d'évaluation a été introduite par Rijsbergen [Rijsbergen 1979] où il propose une mesure équilibrée entre la précision et le rappel qui est la mesure « F-measure ». Le rappel d'une classe *i* dans un groupe *k* est mesuré par le nombre d'objets de la classe *i* qui sont dans le groupe *k* divisés par le nombre d'objets de la classe *i*. La précision est définie par le nombre d'objets de la classe *i* qui sont dans le groupe *k* divisés par le nombre d'objets du groupe *k*. La « F-measure » est une moyenne définie entre la précision et le rappel.

Supposant que N_{Ci} est le nombre d'objets dans la classe i, N_k est le nombre d'objet dans le groupe *k* et N_{ik} le nombre d'objets de la classe *i* dans le groupe *k*. La mesure «« F-measure » est définie par :

$$F = \sum_{i=1}^{nb_class} N_{Ci} \quad \max_{k=1}^{nb_Group} \; F(i,k) \qquad [4.3]$$

L'expression $F(i,k)$ est la mesure « F-measure » élémentaire des objets de la classe *i* qui sont contenus dans le groupe *k*. $F(i,k)$ est définie par :

$$F(i,k) = \frac{(1+\alpha) * rappel(i,k) * precision(i,k)}{\alpha * rappel(i,k) + precision(i,k)} \qquad [4.4]$$

Le coefficient est utile pour contrôler l'importance du rappel ou bien la précision. En général, le rappel et la précision ont les mêmes importances :

$$F(i,k) = \frac{2 * rappel(i,k) * precision(i,k)}{rappel(i,k) + precision(i,k)} \qquad [4.5]$$

Le rappel d'une classe *i* dans un groupe *k* est :

$$rappel(i,k) = \frac{N_{ik}}{N_{Ci}} \qquad [4.6]$$

Le rappel global est la somme des rappels de chaque classe divisée par le nombre de classes. Le rappel d'une classe *i* est défini par le rappel maximal de cette classe dans tous les groupes obtenus.

$$rappel = \frac{\sum_{i=1}^{nb_class} \max_{k=1}^{nb_group} \frac{N_{ik}}{N_{Ci}}}{nb_class} \qquad [4.7]$$

La précision d'une classe *i* dans un groupe *k* est :

$$precision(i,k) = \frac{N_{ik}}{N_k} \qquad [4.8]$$

La Précision globale est la somme des précisions de chaque classe divisée par le nombre de classes. La précision d'une classe *i* est définie par la précision maximale de cette classe dans tous les groupes obtenus.

$$precision = \frac{\sum_{i=1}^{nb_class} \max_{k=1}^{nb_group} \frac{N_{ik}}{N_k}}{nb_class} \qquad [4.9]$$

4.1.3. Environnement de travail

Nous avons choisi d'implémenter l'algorithme « KOKM » avec le langage C vue la portabilité du programme sur d'autres machines, la capacité du langage à fournir au programmeur des commandes pour optimiser la gestion de la mémoire et la facilité à programmer un module de chargement de n'importe quel format de fichiers de base de données. L'environnement matériel et logiciel se présente comme suit :

- Machine de capacité 4 GB de mémoire vive ayant un processor « Intel Core 2 Duo (2.1 GHz) ».
- Système d'exploitation « windows vista »
- Un éditeur libre « codeBlocks » pour créer et exécuter les programmes en C.
- Le logiciel libre « Weka » pour créer les graphiques de différentes distributions des données.
- Le logiciel « Matlab » pour créer une matrice de noyaux utilisée uniquement pour l'estimation du nombre de recouvrements.

4.2. Expérimentations sur le dataset « Iris »

La base de données « Iris » [web 2010[1]] contient une description des trois espèces de fleurs « Iris- setosa », « Iris-virginica » et « Iris-versicolor ». Cette base a été utilisée par Ronald Aylmer Fisher en 1936 qui a collecté des données sur ces fleurs pour quantifier leur variation géographique. Elle est la base la plus utilisée pour évaluer les algorithmes de classification. L'ensemble des données se compose de 50 observations sur chaque espèce. Chaque observation est décrite en quatre caractéristiques à savoir la longueur et la largeur des sépales et des pétales mesurée en cm. La spécificité de la base est qu'il existe une espèce (« Iris- setosa ») facilement séparable des deux autres alors que « Iris-virginica » et « Iris-versicolor » sont difficiles à séparer. (Figure 27.). Pour résoudre le problème de séparation entre la classe « Iris-virginica » et la classe « Iris-versicolor », nous avons projeté les données dans un espace infiniment dimensionnel en utilisant un noyau RBF de paramètre $\sigma = 2$.

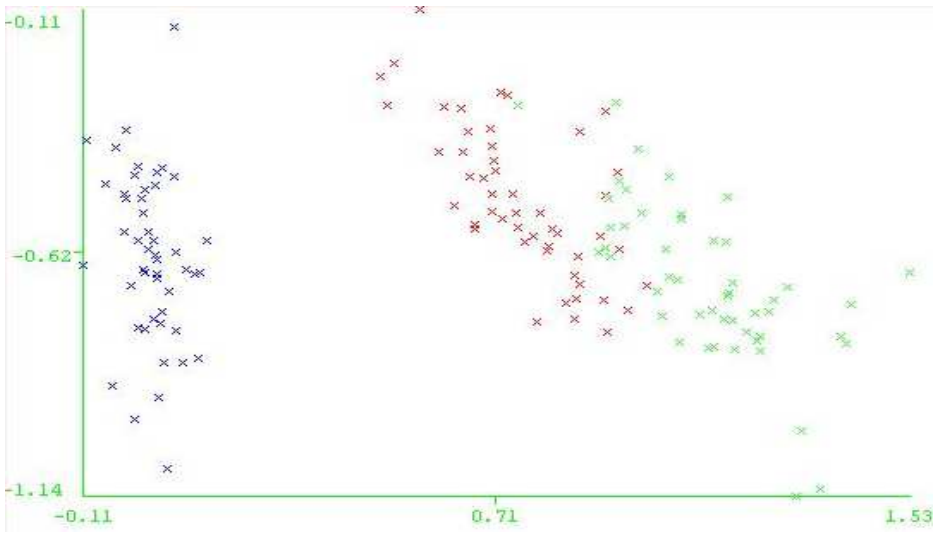


Figure 27. Représentation du dataset « Iris » sur les 2 premiers axes en utilisant la méthode PCA.

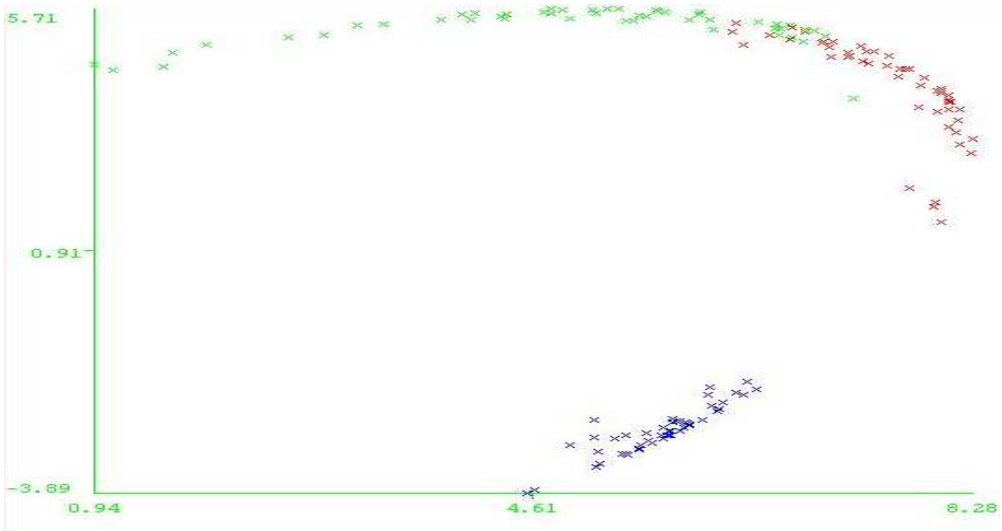


Figure 28. Représentation du dataset « Iris » sur les 2 premiers axes avec un noyau RBF $\sigma = 2$ en utilisant la méthode PCA.

Nous remarquons qu'en projetant les données de la base « Iris » dans un espace infiniment dimensionnel, la classe « Iris-setosa » est devenue plus éloignée des deux autres classes. Alors que pour « Iris-virginica » et « Iris-versicolor » elles restent très proche l'une de l'autre mais plus éloignées que leur représentation dans l'espace d'origine.

4.2.1. Détermination du nombre de classes du dataset « Iris »

La figure 29. présente les valeurs propres les plus significatives dans la matrice de noyaux construite à partir du dataset « Iris » en utilisant un noyau « RBF » avec un paramètre sigma σ = 2. La figure montre la présence de deux ou trois groupes dans les données. Le troisième groupe parait moins important que les autres. La difficulté de séparation entre « Iri-virginica » et « Iris-versicolor » nous donne une probabilité d'existence d'un seul groupe.

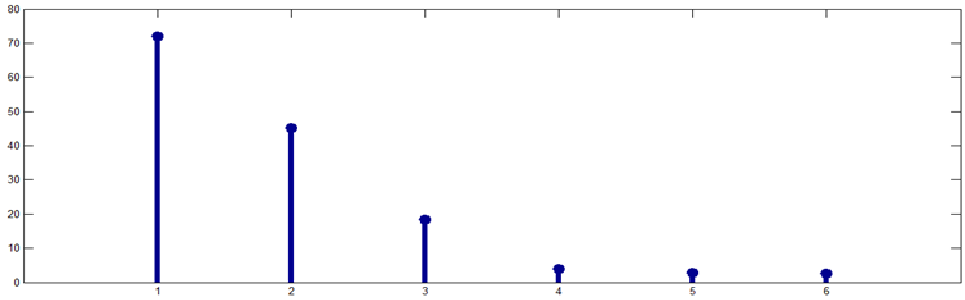


Figure 29 : diagramme en bâton des valeurs propres les plus significatives dans le dataset « Iris ».

Donc en suivant une stratégie d'augmentation du nombre de groupes pour minimiser le chevauchement entre clusters, le choix optimal serait trois clusters.

4.2.2. Comparaison entre « OKM » et « KOKM » sur le dataset « Iris »

Nous avons exécuté la méthode « OKM » avec différentes mesures de distances à savoir distance euclidienne, distance « I-divergence », distance basée sur un noyau RBF de paramètre σ=2 et distance basée sur un noyau polynômial de paramètre d=0.25. Aussi nous avons exécuté la méthode « KOKM » en utilisant un noyau RBF de paramètre σ=2 et en utilisant un noyau polynômial de paramètre d=0.25. Nous avons exécuté dix fois chaque méthode avec les mêmes initialisations des centres des clusters. Le tableau 4. montre les différents résultats obtenus.

	Minimum			Maximum			Moyenne		
	Precision	Rappel	F-measure	Precision	Rappel	F-measure	Precision	Rappel	F-measure
OKM avec distance euclidienne	0.603	0.853	0.706	0.765	0.993	0.854	0.707	0.900	0.815
OKM avec "I-Divergence"	0.514	0.733	0.617	0.848	**1**	0.906	0.759	**0.993**	0.834
OKM avec noyau RBF (σ=2)	0.585	0.906	0.711	0.782	0.993	0.901	0.771	0.960	0.830
OKM avec noyau Polynomial (d=0.25)	0.771	**0.936**	**0.865**	0.838	**1**	0.911	0.808	**0.993**	0.892
KOKM avec noyau RBF (σ=2)	**0.836**	0.87	0.854	**0.935**	0.98	**0.96**	**0.872**	0.925	**0.895**
KOKM avec noyau polynomial (d=0.25)	0.785	0.86	0.82	0.885	**1**	0.939	0.854	0.9	0.88

Tableau 4. Comparaison entre la méthode « OKM » et la méthode « KOKM » avec même importance de la précision et du rappel sur le dataset « Iris »

Nous remarquons tout d'abord que l'utilisation d'une distance basée sur un noyau polynômiale permet d'améliorer les résultats de classification. Cette amélioration est due à l'amélioration remarquable de la mesure de précision. Cette mesure de précision s'améliore encore dans la méthode « KOKM » vu l'exactitude des méthodes de noyaux en projetant les données dans un espace de grande dimension.

	Minimum			Maximum			Moyenne		
	Precision	Reappel	F-measure	Precision	Rappel	F-measure	Precision	Rappel	F-measure
OKM avec distance euclidienne	0.603	0.853	0.668	0.765	0.993	0.828	0.707	0.900	0.761
OKM avec "I-Divergence"	0.514	0.733	0.571	0.848	**1**	0.893	0.759	**0.993**	0.824
OKM avec noyau RBF (σ=150)	0.585	0.906	0.663	0.782	0.993	0.842	0.771	0.960	0.825
OKM avec noyau Polynomial (d=0.25)	0.771	**0.936**	0.819	0.838	**1**	0.886	0.808	**0.993**	0.862
KOKM avec noyau RBF l(σ=2)	**0.836**	0.87	**0.847**	**0.935**	0.98	**0.950**	**0.872**	0.925	**0.889**
KOKM avec noyau polynomial (d=0.25)	0.785	0.86	0.809	0.885	**1**	0.920	0.854	0.9	0.869

Tableau 5. Comparaison entre la méthode « OKM » et la méthode « KOKM » en donnant une double importance de la précision par rapport au rappel sur le dataset « Iris »

L'amélioration de la méthode « KOKM » par rapport à la méthode « OKM » en utilisant un noyau polynômiale n'est pas importante en terme de « F- measure » vu que l'importance de mesure de précision et de la mesure de rappel sont les mêmes. Le tableau 5 montre bien l'amélioration importante de la méthode « KOKM » si nous donnons une importance double pour la précision par rapport au rappel.

4.3. Expérimentations sur le dataset « Ionosphère »

La base de données « Ionosphère » [web 2010[3]] était construite par un système de radar à Goose Bay au Labrador. Ce système de radar contient 16 antennes avec une puissance totale transmise de 6.4 kilowatts. Ce système analyse les électrons dans l'ionosphère. Certains électrons témoignent un certain type de structure dans l'ionosphère. Ces électrons définissent la première classe dans le jeu de données qui est la classe « good». D'autres électrons ne témoignent aucune structure dans l'ionosphère. Ils définissent la deuxième classe dans le jeu de données qui est la classe « bad ». Les électrons sont transmis via un signal émis par les antennes. Ce signal est décrit en 34 attributs qui vont constituer la dimension de la base de données « Ionosphère ». Le nombre de signaux total dans la base est 351 signaux. La caractéristique de ce jeu de données est que les deux classes ont une forme concentrique ce qui va être difficile à les séparer par certains algorithmes de classification comme indique la Figure 30.

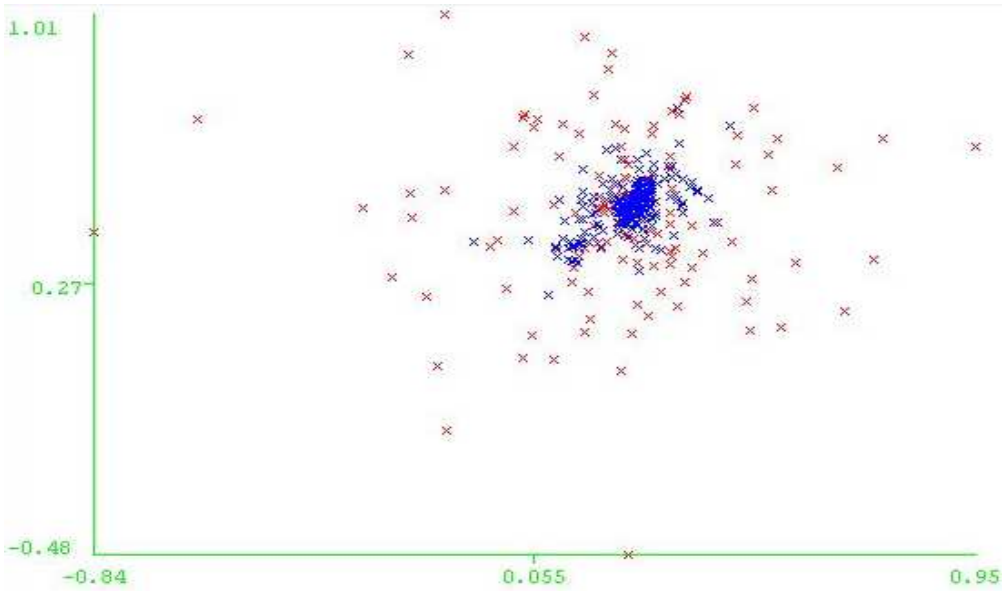


Figure 30. Représentation de la base « Ionosphère » sur les deux premiers axes en utilisant la méthode PCA.

La projection de ces données dans un autre espace de dimension plus grande permet d'enrichir les connaissances sur ces deux classes et permet de créer une séparation entre eux.

La Figure 31. montre qu'en projetant ces données dans un espace infiniment dimensionnel en utilisant un noyau RBF de paramètre $\sigma = 20$, nous pouvons faciliter la tâche de séparation des bons et des mauvais signaux. Les clusters ont perdu leur forme concentrique dans le nouvel espace de représentation.

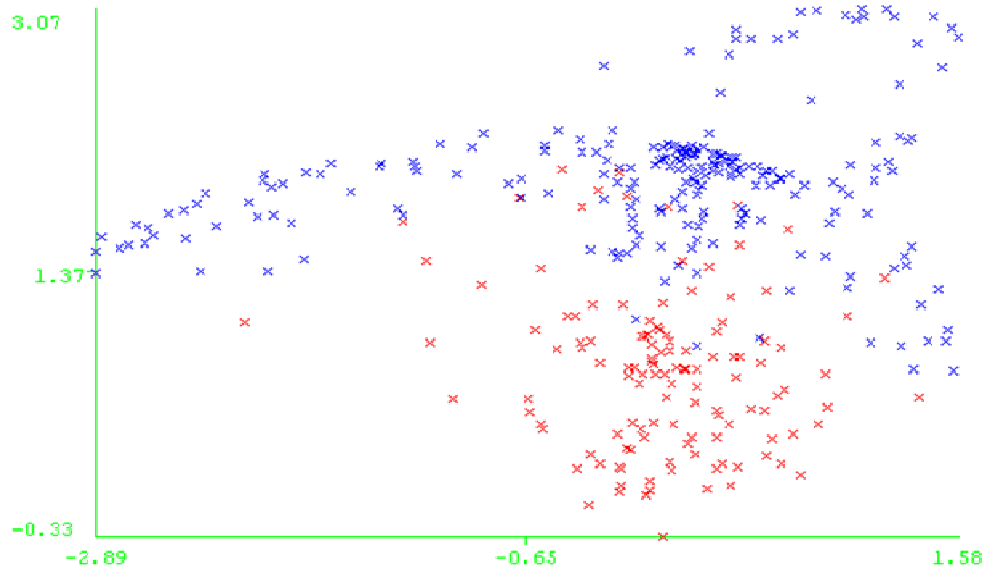


Figure 31. Représentation du dataset Ionosphère sur les deux premiers axes avec un noyau RBF $\sigma = 20$ en utilisant la méthode PCA.

4.3.1. Détermination du nombre de classes du dataset « Ionosphère »

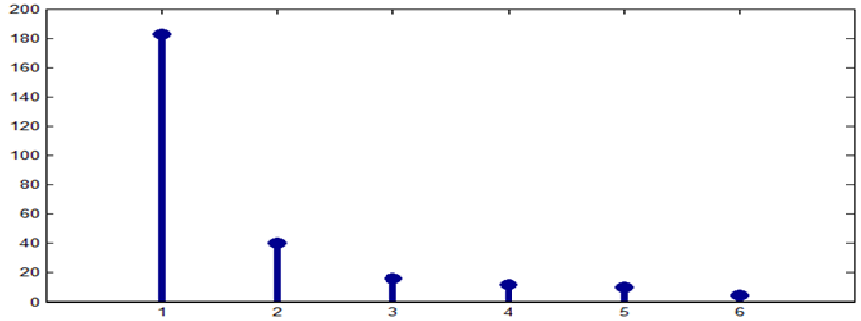


Figure 32. Diagramme en bâton des valeurs propres les plus significatives dans le dataset « Ionosphère » en utilisant un noyau RBF de paramètre $\sigma = 20$

La Figure 32. montre deux valeurs propres significatives dans le dataset « Ionosphère ». La première valeur parait plus importante vu l'existence d'une zone très dense par rapport aux autres zones. Le nombre optimal de classes est donc 2 classes.

4.3.2. Comparaison entre « OKM » et « KOKM » sur le dataset « Ionosphère »

Nous avons adapté la même stratégie utilisée pour évaluer la méthode « OKM » avec différentes mesures de distance et la méthode « KOKM » sur le jeu de données « IRIS ». Nous avons effectué dix exécutions de chaque méthode avec les mêmes initialisations dans chaque exécution. Le tableau 6. montre les différents résultats obtenus en supposant que la mesure de précision et la mesure du rappel ont les mêmes importances.

	Minimum			**Maximum**			**Moyenne**		
	Precision	Reappel	F-measure	Precision	Rappel	F-measure	Precision	Rappel	F-measure
Okm Euclidean distance	0.330	0.58	0.421	0.637	0.752	0.689	0.532	0.689	0.597
OKM avec noyau RBF (σ=0.1)	0.330	0.58	0.421	0.683	0.780	0.728	0.56	0.737	0.636
OKM avec noyau Polynomial (d=2)	0.325	**0.605**	0.423	0.677	0.763	0.718	0.58	0.721	0.644
KOKM avec noyau RBF l(σ=2)	**0.35**	0.6	**0.445**	**0.775**	**0.783**	**0.765**	0.662	0.745	0.701
KOKM avec noyau polynomial (d=4)	**0.35**	0.6	**0.445**	0.677	0.763	0.718	0.6	0.73	0.658

Tableau 6. Comparaison entre la méthode « OKM » et la méthode « KOKM » avec même importance de la précision et du rappel sur le dataset « Ionosphère ».

Nous remarquons dans ces résultats que l'amélioration de « F-measure » est influencée par l'amélioration du rappel ainsi que l'amélioration de la précision contrairement à la base « Iris ». L'amélioration du rappel justifie la problématique que la méthode « OKM » ne permet pas de trouver les classes ayant des formes concentriques. Pour la précision, nous remarquons qu'il n'y a pas une bonne amélioration par rapport à l'amélioration des résultats dans la base « Iris » parce que le nombre de classes est inferieur au nombre de classes des données « Iris » ce qui diminue naturellement la probabilité d'erreur de classification.

4.4. Expérimentations sur le dataset « Eachmovie »

La base de données « EachMovie » [web 2010[2]] contient des évaluations des internautes pour certains films et certaines vidéos en lignes. Les internautes donnent des notes de un à cinq, avec cinq indiquant que l'internaute a bien aimé la vidéo. La base contient en totale 74.424 utilisateurs. Le nombre moyen et le nombre médian des utilisateurs qui votent au moins une vidéo sont 1732 et 379 respectivement. Les informations correspondantes au genre du film ou de la vidéo sont extraites à partir de « IMDB : Internet Movie Database ». Si chaque genre de vidéo est considéré comme une classe contenant plusieurs vidéos, alors cette base de données contient naturellement des classes recouvrantes. Un film peut appartenir à plusieurs genres. Par exemple le film « Aliens » appartient à trois genres de films: le genre « Action », le genre «horror » et le genre « science-fiction ».

A partir de cet ensemble de données, nous avons construit un sous ensemble de jeu de données contenant 75 films dispersés sur trois classes recouvrantes : Action = 21 films, Comédie = 26 films, Crime = 17 films et 11 films appartenant simultanément au genre action et crime. Le processus de classification dans ce sous ensemble consiste à déterminer le genre du film en se basant sur l'âge, le sexe et la note d'évaluation des internautes. La figure 33. montre la distribution des 3 genres de film sur les deux premiers axes en utilisant la méthode PCA. Les films en cercle sont des films recouvrants qui appartiennent au genre Action (en bleu) et au genre Crime(en vert).

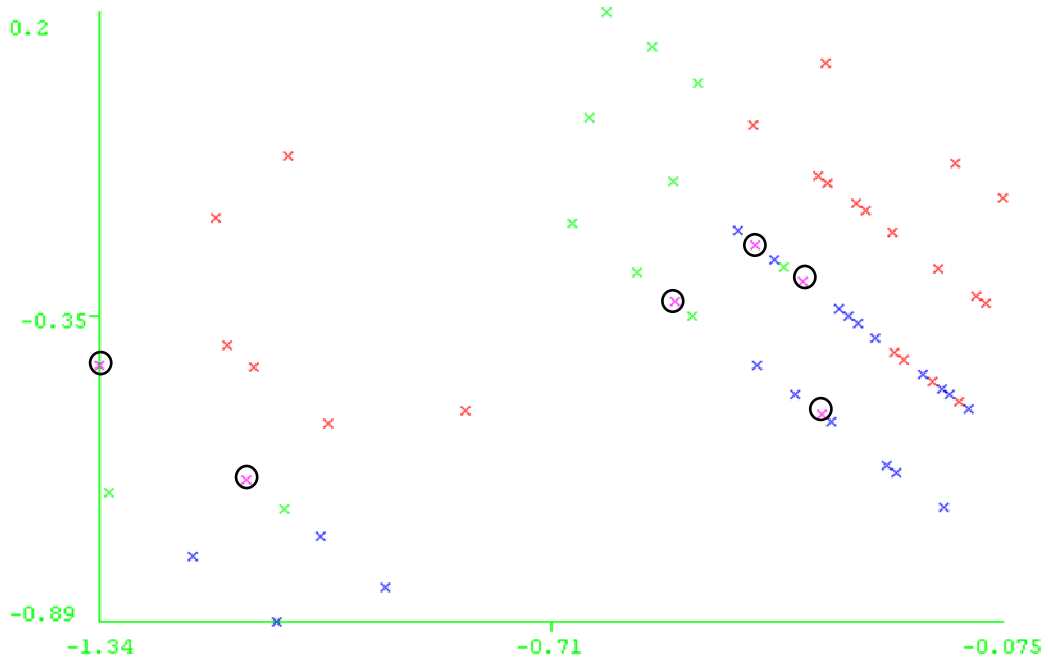


Figure 33. Représentation d'un sous ensemble du dataset « Eachmovie » contenant 3 genres de films sur les deux premiers axes en utilisant la méthode PCA.

En projetant ces données dans un espace infiniment dimensionnel en utilisant un noyau RBF de paramètre σ = 2, nous remarquons une amélioration de la représentation des films recouvrants dans la figure 34. puisque ces films (en cercles) se trouvent dans l'espace de redescription entre les films de type Action (en bleu) et les films de type Crime(en vert). Malgré l'amélioration au niveau des films recouvrants, les 3 genres de films restent difficiles à séparer puisque la note de l'utilisateur, son sexe et son âge ne permet pas de construire une bonne connaissance sur le genre de la vidéo.

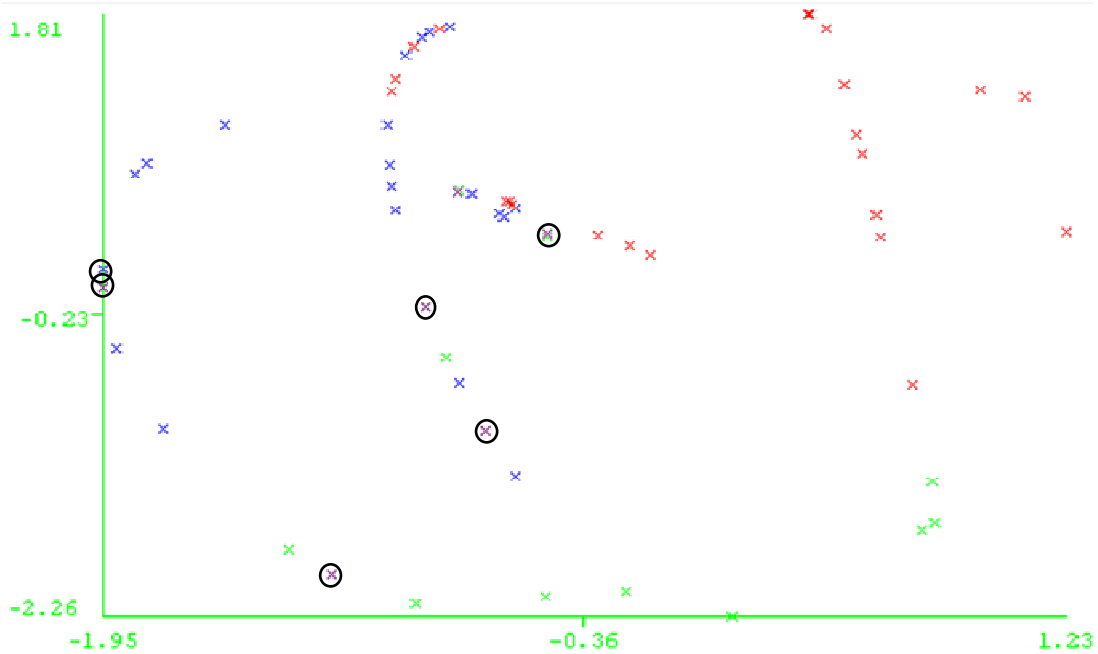


Figure 34. Représentation d'un sous ensemble de la base « Eachmovie » contenant 3 genres de films avec un noyau RBF $\sigma = 20$ sur les deux premiers axes en utilisant la méthode PCA.

4.4.1. Détermination du nombre de classes dans le dataset « Eachmovie »

La figure 35. montre qu'il existe entre deux et trois valeurs propres significatives dans le sous ensemble construit à partir de la base « Eachmovie ». En suivant une stratégie d'augmentation du nombre de classes pour minimiser le chevauchement entre les clusters le choix optimal serait trois clusters.

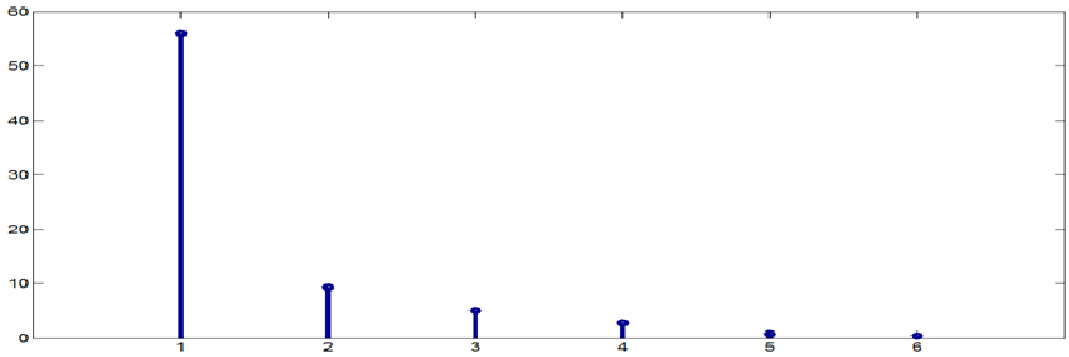


Figure 35. Diagramme en bâton des valeurs propres les plus significatives d'un sous ensemble du dataset « Eachmovie » contenant 3 genres de films en utilisant un noyau RBF de paramètre $\sigma = 20$.

4.4.2. Comparaison entre « OKM » et « KOKM » sur le dataset « Eachmovie »

Nous avons adapté la même stratégie utilisée pour évaluer la méthode « OKM » avec différentes mesures de distance et la méthode « KOKM » sur le jeu de données « Iris ». Nous avons effectué dix exécutions de chaque méthode avec les mêmes initialisations dans chaque exécution. Le tableau 7. montre les différents résultats obtenus en supposant que la mesure de précision et la mesure du rappel ont les mêmes importances.

	Minimum			**Maximum**			**Moyenne**		
	Precision	Reappel	F-measure	Precision	Rappel	F-measure	Precision	Rappel	F-measure
OKM avec distance euclidienne	0.488	0.641	0.530	0.654	**0.919**	0.715	0.557	0.788	0.616
OKM avec "I-Divergence"	0.530	0.552	0.540	0.632	0.814	0.711	0.582	0.687	0.630
OKM avec noyau RBF (σ=20)	0.494	0.641	0.540	0.673	**0.919**	0.740	0.582	0.827	0.687
OKM avec noyau Polynomial (d=2)	0.550	**0.693**	**0.653**	0.7	0.909	0.766	0.628	**0.851**	**0.721**
KOKM avec noyau RBF (σ=2)	0.591	0.491	0.585	0.747	0.823	**0.769**	**0.693**	0.637	0.668
KOKM avec noyau polynomial (d=2)	**0.663**	0.590	0.612	**0.768**	0.685	0.724	0.700	0.615	0.665

Tableau 7. Comparaison entre la méthode « OKM » et la méthode « KOKM » avec même importance de la précision et du rappel sur le dataset « Eachmovie »

Nous remarquons que la méthode « OKM » utilisée avec des distances basées sur un noyau polynomial donne la valeur « F-measure » la plus élevée. L'utilisation de cette distance à permis d'améliorer simultanément la mesure de précision et la mesure de rappel par rapport à la distance euclidienne. Nous remarquons aussi que le taux de précision de la méthode « KOKM » est supérieur au taux de précision de la méthode « OKM » avec toutes les distances utilisées. Le tableau 8. Montre les résultats obtenus si nous donnons une importance double à la mesure de précision par rapport à la mesure du rappel. Nous remarquons que la méthode « OKM » utilisée avec des distances basées sur un noyau polynomial reste la méthode la plus performante, mais avec une diminution de l'écart de performance par rapport aux premiers résultats obtenus dans le tableau 7.

	Minimum			Maximum			Moyenne		
	Precision	Reappel	F-measure	Precision	Rappel	F-measure	Precision	Rappel	F-measure
OKM avec distance euclidienne	0.488	**0.641**	0.530	0.654	**0.919**	0.715	0.557	0.788	0.616
OKM avec "I-Divergence"	0.530	0.552	0.537	0.632	0.814	0.682	0.582	0.687	0.613
OKM avec noyau RBF (σ=20)	0.494	0.641	0.534	0.673	**0.919**	0.738	0.582	**0.827**	0.645
OKM avec noyau Polynomial (d=2)	0.55	**0.693**	0.590	0.7	0.909	0.758	0.628	0.851	0**.688**
KOKM avec noyau RBF (σ=2)	0.591	0.491	0.585	0.747	0.823	**0.769**	0.693	0.637	**0.673**
KOKM avec noyau polynomial (d=2)	**0.663**	0.590	**0.612**	**0.768**	0.685	0.724	**0.700**	0.615	0.654

Tableau 8. Comparaison entre la méthode « OKM » et la méthode « KOKM » en donnant une double importance de la précision par rapport au rappel sur le dataset « Eachmovie ».

Conclusion

Les différentes expérimentations réalisées sur les différentes bases de données montrent empiriquement la performance des solutions proposées. L'utilisation des distances basées sur les méthodes de noyaux permet d'améliorer la précision de la méthode « OKM » et permet d'améliorer le rappel si les classes sont de formes complexes. La méthode « KOKM » permet aussi d'améliorer davantage les résultats de classification. La précision de la méthode « KOKM » est toujours supérieur à toute les précisions obtenues par La méthode « OKM » avec toutes les distances utilisées. En ce qui concerne l'estimation du nombre de recouvrements, elle est utile pour déterminer des intervalles et pour construire des hypothèses sur la structure des données.

Conclusion générale

Plusieurs méthodes de classification automatique ont été proposées dans la littérature. Le choix d'une méthode de classification dépend de la nature de l'application et du type de schéma de classification désiré. Nous nous sommes intéressés dans le cadre de ce mémoire au schéma de classification recouvrante qui désigne le groupement des objets dans des classes pouvant s'intersecter. Ce schéma de classification ne constitue pas un nouveau domaine dans la classification automatique, mais c'est un domaine relancé par des besoins applicatifs. Plusieurs problèmes réels de classification sont naturellement recouvrants et présentent des intersections naturelles entre les clusters.

Nous avons proposé dans ce mémoire une étude détaillée des principales méthodes de classification recouvrante. Les méthodes de classification étudiées cherchent directement des recouvrements optimaux et ne se basent pas sur l'hypothèse d'extension des schémas de classification floue pour produire des schémas recouvrants. Nous avons concentré notre travail sur les méthodes de réallocation dynamique, principalement sur la méthode « OKM : Overlapping K_means». Pour résoudre le problème du nombre de recouvrements qui est fixé a priori dans cette méthode, nous avons proposé une méthode d'estimation de ce nombre en utilisant la technique des méthodes de noyaux. Cette estimation est valable pour toutes les méthodes recouvrantes qui considèrent le nombre de recouvrements comme un paramètre. Aussi, en utilisant cette même technique de projection explicite des données, nous avons proposé une reformulation de « OKM » pour utiliser des distances basées sur les méthodes de noyaux afin de résoudre la problématique des classes de formes complexes et afin d'améliorer la précision de « OKM ». Les distances entre les objets sont calculées dans un espace de dimension plus grand ce qui permet de mieux trouver les bonnes séparations entre les données.

Le point critique dans l'utilisation des méthodes de noyaux comme mesure de distance dans la méthode « OKM » est que seules les distances entre les objets sont calculées dans l'espace de redescription. L'étape d'affectation multiple d'un objet à un ou plusieurs clusters et l'étape de mise à jour des centres des clusters sont réalisées dans l'espace d'origine et non pas dans l'espace de projection. Comme solution, nous avons proposé la méthode « KOKM : Kernel Overlapping K-Means» qui permet de combiner les avantages de la méthode « kernel

k-means » [Camastra and Verri 2005] pour produire des classes dans l'espace de redescription, et les avantages de la méthode « OKM » pour produire des classes recouvrantes.

Les différentes expérimentations réalisées sur différents data sets à savoir « Iris », « Ionosphère » et « Eachmovie » ont montré l'efficacité d'utilisation de la technique de noyaux dans la classification recouvrante. La méthode « KOKM » permet d'améliorer la mesure de précision dans tous les ensembles de données testés. Cette méthode permet également d'améliorer la qualité de classification (amélioration de la mesure du rappel et amélioration de la mesure de précision) si les groupes traités ont des formes complexes telles le cas de la base « Ionosphère » qui contient des classes de formes concentriques.

Nous visons dans des futurs travaux d'étendre l'application de la méthode « KOKM » sur des données structurées non vectorielles telles que les données textuelles, les données sous forme d'arbres et les données sous forme d'histogrammes. Plusieurs travaux proposent des métriques basées sur les méthodes de noyaux adaptées à des domaines variées de représentation de données. Par exemple, des mesures de noyaux ont été définies pour la représentation des images [Harchaoui and Bach 2007] . D'autres mesures de noyaux sont adaptées à la classification textuelle. Ces mesures permettent en effet, d'introduire des connaissances sémantiques dans la formule de noyau ce qui permet d'améliorer la qualité de classification des données textuelles.

Une autre perspective concerne la détermination des recouvrements optimaux dans un espace fortement dimensionnel et ce en étendant la méthode « OKMED ». L'avantage de cette méthode est que le centre d'un recouvrement ne peut appartenir qu'aux objets contenus dans ce recouvrement ce qui pourrait améliorer les résultats de classification même en présence des données éloignées.

Références Bibliographiques

[Banerjee and *al.* 2005a] Banerjee A., Krumpelman C., Basu S., Mooney R., Ghosh J., "Clustering with Bregman divergences". Journal of Machine Learning Research, vol 7, pp 1705–1749, 2005.

[Banerjee and *al.* 2005b] Banerjee A., Krumpelman C., Basu S., Mooney R., Ghosh J., "Model based overlapping clustering". International Conference on Knowledge Discovery and Data Mining, Chicago, USA, 2005.

[Ben-Hur and *al.* 2000] Ben-Hur A. and Horn D., Siegelmann H.T., Vapnik V., "A support vector clustering method", International Conference on Pattern Recognition,vol 2 , pp 724-727, 2000.

[Berkhin 2002] Berkhin, P., "Survey Of Clustering Data Mining Techniques". Technical report, Accrue Software, San Jose, CA, 2002.

[Bertrand and Janowitz 2003] Bertrand P., Janowitz M. F., "The k-weak hierarchical representations: An extension of the indexed closed weak hierarchies". Discrete Applied Mathematics, vol 127, pp 199–220, 2003.

[Bezdek 1981] Bezdek J., "Pattern Recognition with Fuzzy Objective Function Algorithms". Plenum Press, New York, 1981.

[Boley 1998] Boley D.L., "Principal direction divisive partitioning". Data Mining and Knowledge Discovery, vol 2, pp 325-344, 1998.

[Camastra and Verri 2005] Camastra F., Verri A., "A Novel Kernel Method for Clustering". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 27, pp 801-804, 2005.

[Cheeseman and Stutz 1996] Cheeseman P., Stutz J., "Bayesian Classification (AutoClass): Theory and Results". Advances in Knowledge Discovery and Data Mining, pp 153-180, 1996

[Cleuziou and *al.* 2004] Cleuziou G., Martin L., Vrain C., "PoBOC : an Overlapping Clustering Algorithm.Application to Rule-Based Classication and Textual Data". Proceedings of the 16th European Conf, on Artificial Intelligence, Valencia, Spain, pp 440-444, 2004.

[Cleuziou 2007] Cleuziou G., "Les okm : une extention des k-moyennes pour la recherche de classes recouvrantes ". Revue des Nouvelles Technologies de l'Information, pp 691–702, 2007.

[Cleuziou 2009] Cleuziou G., "OKMED et WOKM : deux variantes de OKM pour la classification recouvrante". in 9èmes Journées Francophones d'Extraction et de Gestion des Connaissances, Revue des Nouvelles Technologies de l'Information, Vol.1, pp 31-42, 2009.

[Cornuéjols and *al.* 2002] Cornuéjols A. and Miclet L., Kodratoff Y., "Apprentissage Artificiel : Concepts et algorithmes". Eyrolles, 2002.

[Daoqiang and Zhang 2002] Daoqiang G., Zhang C., "Fuzzy Clustering Using Kernel Method". Proceedings of International Conference on Control and Automation, Xiiamen, China, 2002.

[Defays 1977] Defays D., "An efficient algorithm for a complete link method". The Computer Journal, vol 20, pp 364-366, 1977.

[Didimo and *al.* 2007] Didimo W. and Giordano F., Liotta G., "Overlapping Cluster Planarity". In Proceedings of Asia Pacific Symposium on Visualisation, pp 73-80, 2007.

[Diday 1984] Diday E., "Orders and overlapping clusters by pyramids". Technical report, INRIA num.730, France, 1987.

[Fisher 1987] Fisher D., "Knowledge acquisition via incremental conceptual clustering". Journal Machine Learning, vol 2, pp 139-172, 1987.

[Dunn 1973] Dunn J.C., "A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-Separated Clusters". J. Cybernetics, vol 3, pp 32-57, 1973.

[Ester and *al.* 1996] Ester M., Kriegel H-P. and Sander J. Xu, X., "A density-based algorithm for discovering clusters in large spatial databases with noise". In Proceedings of the 2nd SIGKDD pp 226-231, Portland, Oregon, 1996.

[Fayyad and *al.* 1996] Fayyad M.U., and Piatesky-Shapiro G., Smuth P., Uthurusamy R., "Advances in Knowledge Discovery and Data Mining". AAAI Press, 1996.

[Fellows and *al.* 2009] Fellows M. and Guo J., Komusiewicz C., Niedermeier R., Uhlmann J. ,"Graph-Based Data Clustering with Overlaps". Computing and Combinatorics, pp 516-526,USA, 2009.

[Figueiredo and Jain 2002] Figueiredo M. and Jain A.K., "Unsupervised learning of finite mixture models". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 24, pp 381–396, 2002.

[Fraley and Raftery 1999] Fraley C. and Raftery A., "MCLUST: Software for model-based cluster and discriminant analysis". Tech Report 342, Dept. Statistics, Univ. of Washington, USA, 1999.

[Gennari and *al.*1989] Gennari, J. and Langley, P., Fisher D., "Models of incremental concept formation". Artificial Intelligence, vol 40, pp 11-61, 1989.

[Girolami 2002] Girolami M., "Mercer Kernel-Based Clustering in Feature Space". IEEE Transactions on Neural Networks, vol 13, pp 780–784, 2002.

[Griffiths and Ghahramani 2006] Griffiths T. and Ghahramani Z., "Infinite latent feature models and the indian buffet process". In Advances in Neural Information Processing Systems, volume 16, 2006.

[Guha and *al.* 1998] GUHA S. and RASTOGI R., SHIM K., "CURE: An efficient clustering algorithm for large databases". In Proceedings of the ACM SIGMOD Conference, pp 73-84, Seattle, WA, 1998.

[Halkidi and *al.* 2004] Halkidi M. and Yannis B., Brodley Carla E., Wrobel S., "On validation Clustering techniques",.Journal of Intelligent Information Systems, vol 17, pp 107–145, 2004.

[Hansen and Yu 2001] Hansen M. H. and Yu B., «Model selection and the principle of minimum description length". Journal of the American Statistical Association, vol 96, pp 746–774, 2001.

[Heller and Ghahramani 2007] Heller K. and Ghahramani Z., "A nonparametric bayesian approach to modeling overlapping clusters"3 11th International Conference on Artifical Intelligence and Statistics, San Juan, Puerto Rico,2007.

[Huang 1997] Huang Z., "A Fast Clustering Algorithm to Cluster very Large Categorical Data Sets". Data Mining and Knowledge Discovery, vol 2, pp 283-304, 1997.

[Johnson 1967] Johnson S.C., "Hierarchical Clustering Schemes". Psychometrika, vol 32, pp 241-254, 1967.

[Karypis and *al.* 1999] Karypis G. and Han E.H., Kumar V., "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling". COMPUTER, vol 32, pp 68-75, 1999.

[Kaufman and Rousseeuw 1990] Kaufman L. and Rousseeuw P., "Finding Groups in Data: An Introduction to Cluster Analysis". John Wiley and Sons, New York, NY, 1990.

[Khodja 1997] Khodja L., "Contribution à la Classification Floue non Supervisée". Thèse de l'université de Savoie, 1997.

[Krishnapuram and Keller 1993] Krishnapuram R. and Keller J., "A possibilistic Approch to Clustering". IEEE Transactions on Fuzzy Systems, vol 1, 1993.

[MacQueen 1967] MacQueen J.B., "Some Methods for Classification and Analysis of Multivariate Observations". In Proceedings of 5th Berkley Symposium on Mathematical Statistics and Probability, vol 1, pp 281–297, 1967.

[Ng and Han 1994] Ng R. and Han J., "Efficient and effective clustering methods for spatial data mining". In Proceedings of the 20th Conference on VLDB, pp 144-155, Santiago, Chile, 1994.

[Rijsbergen 1979] Van Rijsbergen C.J., "Information Retireval", Butterworths, London, 1979.

[Satish and Sekhar 2004] Satish D.S. and Sekhar C.C., "Kernel based clustering and vector quantization for speech recognition". In 14th IEEE Signal Processing Society Workshop, pp 315–324, 2004.

[Schölkopf and Smola 2002] Schölkopf B. and Smola A.J., "Learning with kernels: Support vector machines". MIT Press, 2002.

[Sibson 1973] Sibson R. "SLINK: An optimally efficient algorithm for the single link cluster method". Computer Journal, vol 16, pp 30-34, 1973.

[Sheikholeslami and *al.* 1998] Sheikholeslami G. and Chatterjee S., Zhang, A., "WaveCluster: A multiresolution clustering approach for very large spatial databases". In Proceedings of the 24th Conference on Very Large Data Bases, pp 428-439, NY,USA, 1998.

[Voorhees 1986] Voorhees E.M.,"Implementing agglomerative hierarchical clustering algorithms for use in document retrieval". Information Processing and Management", vol 22, pp 465-476, 1977.

[Wallace and Dowe 1994] Wallace C. and Dowe D., "Intrinsic classification by MML. the Snob program". In the Proceedings of the 7th Australian Joint Conference on Artificial Intelligence, pp 37-44, Armidale, Australia, 1994.

[Wang and *al.* 1997] Wang W. and Yang J., Muntz R., "Sting: a statistical information grid approach to spatial data mining". In Proceedings of the 23th Conference on Very Large Data Bases, pp 186-195, Athens, Greece, 1997.

[WEB 2010[1]] Iris dataset téléchargeable à partir de ce lien : http://archive.ics.uci.edu/ml/datasets/Iris. Consulté en Janvier 2008.

[WEB 2010[2]] EachMovie dataset téléchargeable à partir de ce lien : http://www.grouplens.org/node/76. Consulté en Janvier 2008.

[WEB 2010[3]] Ionosphère téléchargeable à partir de ce lien : http://archive.ics.uci.edu/ml/datasets/Ionosphere. Consulté en Mars 2008.

[Xu and Wunsch 2005] Xu R., Wunsch D., "survey of clustering algorithms", IEEE Transactions on Neural Networks, pp 6:645–678, 2005.

[Harchaoui and Bach 2007] Harchaoui Z., Bach F., "Image Classification with Segmentation Graph Kernels». IEEE Conference on Computer Vision and Pattern Recognition, pp 1-8, MN, USA, 2007.

Annexe 1
Algorithme de la méthode « OKM »

L'algorithme principal de la méthode « OKM » peut être décrit en pseudo code comme suit :

OKM (X, t_{max},ε, K)

<u>Entrée :</u>

-X : ensemble de vecteurs dans l'espace R^d : l'ensemble des individus à classifier

-t_{max} : nombre maximum d'itérations

- ε : seuil d'amélioration minimale pour continuer une autre itération

K : nombre de clusters

<u>Sorties :</u>

$\{\pi_c\}_{c=1}^{k}$ *: Les recouvrements finaux retournés par l'algorithme*

<u>Etapes :</u>

1) *Initialiser aléatoirement les clusters par des centres arbitraires* $\{m_c^{(0)}\}_{c=1}^{K}$
2) *Pour chaque objet x_i dans X calculer les affectations de x_i aux différents clusters* $A_i^{(0)} = ASSIGN\ (x_i, \{m_c^{(0)}\}_{c=1}^{K})$ *et dériver le recouvrement initial* $\{\pi_c\}_{c=1}^{k}$
3) *t=0 // t est le numéro de l'itération*
4) *Pour chaque cluster* π_c^t *calculer le nouveau centre* $m_c^{(t+1)} = PROTOTYPE\ (\pi_c^t)$

5) *Pour chaque objet xi dans X calculer les affectations* $A_i^{(t+1)} = ASSIGN\ (x_i, \{m_c^{(t+1)}\}_{c=1}^{K}, A_i^{(t)})$ *et dériver le nouveau recouvrement* $\{\pi_c^{t+1}\}_{c=1}^{K}$

6) *<u>Si</u> (t_{max} > t) ou ($J(\{\pi_c^{(t)}\}) - J(\{\pi_c^{(t+1)}\}) > \varepsilon$), mettre t=t+1et retourner à l'étape 4*

<u>Sinon</u> stop et retourné la distribution actuelle $\{\pi_c^{t+1}\}_{c=1}^{K}$

Figure 36. Algorithme principal de la méthode « OKM »[Cleuziou 2007].

La fonction « Prototype » utilisée dans cet algorithme permet à chaque itération de calculer le centre d'un recouvrement. Le centre calculé tient compte des objets qui sont affectés à plusieurs recouvrements. Le pseudo code de cette fonction est :

Prototype (π_h)

<u>Entrée :</u>

π_h : Cluster à calculer le centre

<u>Sortie :</u>

m_h : le nouveau centre du cluster

<u>Etapes :</u>

1) *Calculer le nouveau centre avec la formule :*

$$m_h^* = \frac{1}{\sum_{x_i \in \pi_h} \alpha_i} \sum_{x_i \in \pi_h} \alpha_i . m_h^i$$

où m_h^i est égale à : $m_h^i = |A_i| * x_i - \sum_{m_c \in A_i / \{m_h\}} m_c$

Figure 37. Description de la fonction « Prototype » utilisée dans la méthode « OKM ».

La fonction « ASSIGN » utilisée dans l'algorithme principal de « OKM » permet l'affectation multiple d'un objet à un ou plusieurs clusters. Le pseudo code de cette fonction est :

$ASSIGN\ (x_i, \{m_c^{(0)}\}_{c=1}^{K}, A_i^{old})$

Entrée :

x_i *: un vecteur dans* R^d

$\{m_1,\ldots m_k\}$ *: l'ensemble des centres des k clusters*

A_i^{old} *: L'ensemble des anciennes affectations de l'objet i aux différents clusters.*

Sortie :

$A_i^{old} \subset \{m_1,\ldots, m_k\}$ *Un sous ensemble des affectations de l'objet* x_i.

Etapes :

1) *calculer* $\varphi(x_i)$ *avec les affectations* $A_i = \{m^*\}$ *tel que* $m^* = \arg\min_{\{m_c\}_{c=1}^{K}} \|x_i - m_c\|^2$
2) *Chercher le cluster le plus proche suivant* $m' = \arg\min_{\{m_c\}_{c=1}^{K} / A_i} \|x_i - m_c\|^2$ *et calculer* $\varphi'(x_i)$ *avec les affectations* $A_i \cup \{m'\}$
3) *Si* $\|x_i - \varphi'(x_i)\|^2 < \|x_i - \varphi(x_i)\|^2$ *alors* $A_i \longleftarrow \{m'\}$ *et* $\varphi(x_i) = \varphi'(x_i)$ *et retourner à l'étape 2.*

   Sinon calculer $\varphi^{old}(x_i)$ *avec* A_i^{old}

   Si $\|x_i - \varphi(x_i)\|^2 <= \|x_i - \varphi^{old}(x_i)\|^2$ *retourner* A_i

   Sinon retourner A_i^{old} .

Figure 38. Description de la fonction « ASSIGN » utilisée dans la méthode « OKM » [Cleuziou 2007].

Annexe2
Algorithme de la méthode « OKMED »

L'algorithme principal de la méthode « OKMED » peut être décrit en pseudo code comme suit :

$OKMED\ (D, k, t_{\max}, \varepsilon)$

Entrée:

D *: Une matrice de dissimilitude de taille* $(N \times N)$ *définit sur l'ensemble de données.*

k *: Le nombre de classses.*

$t_{\max}$ *: Le nombre maximal d'itérations.*

ε *: Seuil d'amélioration minimale pour continuer une autre itération.*

Sortie :

$\{\pi_c\}_{c=1}^{k}$ *: Un ensemble de k recouvrements.*

Etapes :

1) *choisir aléatoirement* k *medoids* $\{m_c^{(0)}\}_{c=1}^{k}$ *dans* N ,
2) $t = 0$ *// numéro de l'itération*
3) *Pour chaque donnée* $x_i \in \mathrm{N}$ *calculer les affectations*

$$A_i^{(t+1)} = ASSIGN(x_i, \{m_c^{(t)}\}_{c=1}^{k})$$

Et dériver les recouvrements $\{\pi_c^{(t+1)}\}_{c=1}^{k}$ *tel que* $\pi_c^{(t+1)} = \{x_i \mid m_c^{(t)} \in A_i^{(t+1)}\}$

4) Pour chaque recouvrement $\pi_c^{(t+1)}$ *successivement, calculer le nouveau medoide*

$m_c^{(t+1)} = MEDOID\ (\pi_c^{(t+1)})$

5) Si $\{\pi_c^{(t+1)}\}$ *est différent de* $\{\pi_c^{(t)}\}$ *ou* $t < t_{\max}$ *ou* $J(\{\pi_c^{(t)}\}) - J(\{\pi_c^{(t+1)}\}) > \varepsilon$ *alors* $t = t+1$ *et revenir à l'étape 3 ;*

Sinon retourner les recouvrements finaux $\{\pi_c^{(t+1)}\}_{c=1}^{k}$.

Figure 39. Algorithme principal de la méthode « OKMED » [Cleuziou 2009].

La fonction « *MEDOID* » utilisée dans cet algorithme permet à chaque itération de calculer le médoid d'un recouvrement. Le pseudo code de cette fonction est :

$MEDOID\ (\pi_h)$

Entrée :

π_h *: Cluster à calculer le centre.*

Sortie :

m_h *: le nouveau medoid du cluster* π_h.

Etapes :

1) *Calculer l'inertie des objets contenus dans* π_h *:* $J(\pi_h) = \sum_{x_i \in \pi_h} d^2(x_i, \phi(x_i))$

2) *pour* p *de 1 à* k *faire :*

Pour chaque $x_j \in \pi_h$ *tel que* $|A_j| = p$ *faire :*

Calculer l'image $\phi(x_i)'$ *avec* $m_h = x_j$ *pour chaque* $x_i \in \pi_h$

Calculer la nouvelle inertie de π_h *:* $J'(\pi_h) = \sum_{x_i \in \pi_h} d^2(x_i, \phi'(x_i))$

Si $J'(\pi_h) < J(\pi_h)$ *retourner* $m_h = x_j$

Figure 40. Description de la fonction « *MEDOID* » utilisée dans la méthode « OKMED ».

La fonction « ASSIGN » de l'algorithme « OKMED » permet l'affectation multiple de chaque objet à un ou plusieurs clusters. Le pseudo code de cette fonction est comme suit :

$ASSIGN\ (x_i, \{m_c^{(0)}\}_{c=1}^{K}, A_i^{old})$

Entrée :

x_i *: un vecteur dans* R^d

$\{m_1,\ldots m_k\}$ *: l'ensemble des centres des k clusters*

A_i^{old} *: L'ensemble des anciennes affectations de l'objet i aux différents clusters.*

Sortie :

$A_i^{old} \subset \{m_1,\ldots, m_k\}$ *Un sous ensemble des affectations de l'objet* x_i.

Etapes :

1) *calculer* $\varphi(x_i)$ *avec les affectations* $A_i = \{m^*\}$ *tel que* $m^* = \arg\min_{\{m_c\}_{c=1}^{K}} \|x_i - m_c\|^2$
2) *Chercher le cluster le plus proche suivant* $m' = \arg\min_{\{m_c\}_{c=1}^{K} / A_i} \|x_i - m_c\|^2$ *et calculer* $\varphi'(x_i)$ *avec les affectations* $A_i \cup \{m'\}$
3) *Si* $\|x_i - \varphi'(x_i)\|^2 < \|x_i - \varphi(x_i)\|^2$ *alors* $A_i \longleftarrow \{m'\}$ *et* $\varphi(x_i) = \varphi'(x_i)$ *et retourner à l'étape 2.*

   Sinon calculer $\varphi^{old}(x_i)$ *avec* A_i^{old}

   Si $\|x_i - \varphi(x_i)\|^2 <= \|x_i - \varphi^{old}(x_i)\|^2$ *retourner* A_i

   Sinon retourner A_i^{old} .

Figure 41. Algorithme de la fonction « ASSIGN » utilisée dans « OKMED » [Cleuziou 2007].

Annexe3
Algorithme de la méthode « Kernel k-means »

L'algorithme « Kernel K-Means » peut être décrit en pseudo code comme suit :

Kernel K-Means(D, k)

Entrée :
D : l'ensemble des données à classifier
K :le nombre de classes

Sorties :
$\{FV_c(p)\}_{c=1}^{k}$ *: L'ensemble des classes formées*

Etapes :

1) *Projeter le Jeu de données* D *dans un espace* F *de dimension plus grande, par des fonctions de transformation non-linéaires* ϕ. *Initialiser les centres* a_c, $c = 1,\ldots,k$ *avec* $a_c \in F$.
2) *Calculer pour chaque centre* a_c *le* $FV_c(p)$
3) *Appliquer « SVC » à chaque* $FV_c(p)$ *et assigner à* a_c *le centre :* $a_c = SVC(FV_c(p))$
4) *Passer à l'étape 2 jusqu'à n'importe quels* a_c *change, si non retourner l'espace de redescription (codebook).*

Figure 42. Algorithme principal de la méthode « kernel k-means » [Camastra and Verri 2005].

Annexe4
Algorithme de la méthode « SKK »

L'algorithme « SKK : Soft kernel k-means » peut être décrit en pseudo code comme suit :

SKK(D, ε, k),

<u>Entrée :</u>
D : l'ensemble des données à classifier
K :le nombre de classes
ε: amélioration minimale dans la fonction objective

<u>Sorties :</u>
La distribution finale des objets sur les k classes

<u>Etapes :</u>
1) Initialiser chaque cluster par des objets choisi aléatoirement. Pour chaque objet x_i : si x_i appartient à C_j alors $h_{x_i}^{C_j} = 1$ *sinon* $h_{x_i}^{C_j} = 0$
2) Calculer les distances entre les objets et les centres des clusters avec la formule suivante :

$$dist(\phi(x_i), U_j) = K(x_i, x_i) - \frac{\sum_{l=1}^{N} h_{x_l}^{C_j} K(x_i, x_l)}{\sum_{l=1}^{N} h_{x_l}^{C_j}} + \frac{\sum_{p=1}^{N} \sum_{l=1}^{N} h_{x}^{C_j} h_{x_l}^{C_j} K(x_p, x_l)}{\left(\sum_{l=1}^{N} h_{x_l}^{C_j}\right)^2}$$

3) *Calculer la variance de chaque cluster:*

$$\sigma_j = \frac{\sum_{i=1}^{N} h_{x_i}^{C_j} \, dist \, (\phi(x_i), u_j)}{\sum_{i=1}^{N} h_{x_i}^{C_j}}$$

4) *Calculer la probabilité a priori pour chaque cluster :*

$$P(C_j) = \frac{\sum_{i=1}^{N} h_{x_i}^{C_j}}{N}$$

5) *Calculer la probabilité que chaque point* x_i *soit généré à partir du cluster* C_j.

$$h_{x_i}^{C_j} = \frac{P(x_i / \theta_j) P(C_j)}{\sum_{j=1}^{k} P(x_i / \theta_j) P(C_j)}$$

6) *si l'amélioration dans la fonction objective est supérieure à* ε *alors retourné à l'étape 2 sinon retourné la derniere distrubution des objets.*

Figure 43. Algorithme principal de la méthode « SKK » [Daoqiang and Zhang 2002].