Modèles de mélange de von Mises-Fisher

Wafia Bouberima Parr Bouberima

Université Paris Descartes, France Université de Sétif 1, Algérie

THÈSE EN COTUTELLE ENTRE L'UNIVERSITÉ PARIS DÉSCARTES ET L'UNIVERSITÉ DE SÉTIF 1

Pour l'obtention du grade de

Docteur de l'université Paris Descartes
Mention Intelligence artificielle et décision
Docteur en sciences
Mention Probabilité et Analyse des Données

par

Wafia Parr Bouberima

École doctorale : Informatique, Télécommunication et Électronique Edite de Paris

Laboratoire d'accueil : Laboratoire d'Informatique Paris Descartes

Équipe d'accueil : Gestion et fouille de données

Faculté des Sciences, Université de Sétif 1

Laboratoire d'accueil : Laboratoire de Mathématiques Fondamentales et Numériques

Équipe d'accueil : Optimisation globale et probabilités

Modèles de mélange de von Mises-Fisher

---

soutenue publiquement le 15 Novembre 2013 devant le jury composé de :

Gérard Govaert	**(Professeur, Université de technologie de Compiègne, rapporteur)**
Mohamed Hanafi	**(MCF, HDR, ONIRIS Nantes, rapporteur)**
Mohamed Nadif	**(Professeur, Université Paris Descartes, directeur)**
Yamina Khemal Bencheikh	**(Professeur, Université de Sétif, Algérie, directeur)**
Yves Lechevallier	**(Directeur INRIA, Paris, examinateur)**
Ahlame Douzal	**(MCF, HDR, Université de Grenoble, examinateur)**
Farid Naït-Abdesselam	**(Professeur, Université Paris Descartes, examinateur)**

A Tayba

Remerciements

Je tiens dans un premier temps à exprimer mes sincères remerciements à mes encadreurs Mme. Yamina Khemal Bencheikh, professeur à l'université de Sétif, Algérie et M. Mohamed Nadif, professeur à LIPADE, pour m'avoir encouragée et permise de côtoyer le monde de la recherche en m'accueillant dans leurs équipes de recherche. Je les remercie de leur disponibilité et enthousiasme, de leur gentillesse et pour leurs conseils judicieux ; qu'ils acceptent l'expression de ma grande estime pour leurs personnes et ma profonde considération pour leurs compétences professionnelles.

Je tiens tout particulièrement à remercier M. Gérard Govaert, Professeur à l'université de technologie de Compiègne et M. Mohamed Hanafi, HDR à l'ONIRIS Nantes d'avoir accepté de rapporter ma thèse.

Je remercie également Mme Ahlame Douzal HDR à l'université de Grenoble, M. Yves Lechevallier, Directeur de recherche à INRIA et M. Farid Naït-Abdesselam, Professeur, à l'université Paris Descartes pour avoir fait l´honneur d´examiner mon travail.

Je tiens ensuite à remercier Marylin Galopin et Gislaine Montebello pour leur gentillesse et pour m'avoir aidée à régler mes papiers administratives.

Mes remerciements vont également aux professeurs Lahcène Bencheikh et Kamel Bencheikh de l'université de Sétif pour l'aide qu'il m'ont apportée durant la réalisation de ce travail.

Je remercie mon mari pour sa présence permanente à mes côtés, il a su me supporter, m'encourager et m'accompagner pendant ce travail, pour ses conseils judicieux le long de nos discussions, surtout pour sa patience durant ce long parcours

Je remercie aussi le professeur Said Berimi pour son soutient et conseils.

La thèse n'est pas qu'une aventure scientifique. Elle fût également l'occasion de rencontrer des amis. Je remercie notamment Nabila et Feryal pour les bons moments qu'on a passé ensemble et pour leur gentillesse, leur disponibilité et leurs encouragements. Je remercie également Mahdi, Denis, Imen et son mari Ahmad, Sabrina, Sonia, Nacira, Mariyam et Haifa pour m'avoir soutenue et encouragée à leur tour là où j'avais besoin, en particulier Wassila.

Ces remerciements ne seraient pas complets sans une pensée à ma famille en particulier ma mère, mon père, mes sœurs et mes frères, leurs petites familles ; ainsi que ma petite belle famille surtout mon beau-père, pour m'avoir encouragée tout au long de mes études de près ensuite de loin.

Mes remerciements aussi à tous ceux qui m'ont aidée de loin ou de près.

Résumé

Dans la vie actuelle, les données directionnelles sont présentes dans la majorité des domaines, sous plusieurs formes, différents aspects et de grandes tailles/dimensions, d'où le besoin de méthodes d'étude efficaces des problématiques posées dans ce domaine. Pour aborder le problème de la classification automatique, l'approche probabiliste est devenue une approche classique, reposant sur l'idée simple : étant donné que les g classes sont différentes entre elles, on suppose que chacune suit une loi de probabilité connue, dont les paramètres sont en général différents d'une classe à une autre ; on parle alors de modèle de mélange de lois de probabilités. Sous cette hypothèse, les données initiales sont considérées comme un échantillon d'une variable aléatoire d-dimensionnelle dont la densité est un mélange de g distributions de probabilités spécifiques à chaque classe.

Dans cette thèse nous nous sommes intéressés à la classification automatique de données directionnelles, en utilisant des méthodes de classification les mieux adaptées sous deux approches : géométrique et probabiliste. Dans la première, en explorant et comparant des algorithmes de type $k$means ; dans la seconde, en s'attaquant directement à l'estimation des paramètres à partir desquels se déduit une partition à travers la maximisation de la log-vraisemblance, représentée par l'algorithme EM. Pour cette dernière approche, nous avons repris le modèle de mélange de distributions de von Mises-Fisher (Banerjee *et al.*, 2005), nous avons proposé des variantes de l'algorithme EM$_{vMF}$, soit CEM$_{vMF}$, le SEM$_{vMF}$ et le SAEM$_{vMF}$, dans le même contexte, nous avons traité le problème de recherche du nombre de composants et le choix du modèle de mélange, ceci en utilisant quelques critères d'information : Bic, Aic, Aic3, Aic4, Aicc, Aicu, Caic, Clc, Icl-Bic, Ll, Icl, Awe. Nous terminons notre étude par une comparaison du modèle vMF avec un modèle exponentiel plus simple (Phuong et Vinh, 2008) ; à l'origine ce modèle part du principe que l'ensemble des données est distribué sur une hypersphère de rayon ρ prédéfini, supérieur ou égal à un.

Nous proposons une amélioration du modèle exponentiel qui sera basé sur une étape estimation du rayon ρ au cours de l'algorithme NEM. Ceci nous a permis dans la plupart de nos applications de trouver de meilleurs résultats ; en proposant de nouvelles variantes

de l'algorithme NEM qui sont le NEM$_\rho$, NCEM$_\rho$ et le NSEM$_\rho$.

L'expérimentation des algorithmes proposés dans ce travail a été faite sur une variété de données textuelles, de données génétiques et de données simulées suivant le modèle de von Mises-Fisher (vMF). Ces applications nous ont permis une meilleure compréhension des différentes approches étudiées le long de cette thèse.

Abstract

In contemporary life directional data are present in most areas, in several forms, aspects and large sizes / dimensions ; hence the need for effective methods of studying the existing problems in these fields. To solve the problem of clustering, the probabilistic approach has become a classic approach, based on the simple idea : since the g classes are different from each other, it is assumed that each class follows a distribution of probability, whose parameters are generally different from one class to another. We are concerned here with mixture modelling. Under this assumption, the initial data are considered as a sample of a d-dimensional random variable whose density is a mixture of g distributions of probability where each one is specific to a class.

In this thesis we are interested in the clustering of directional data that has been treated using known classification methods which are the most appropriate for this case. In which both approaches the geometric and the probabilistic one have been considered. In the first, some $k$means like algorithms have been explored and considered. In the second, by directly handling the estimation of parameters from which is deduced the partition maximizing the log-likelihood, this approach is represented by the EM algorithm. For the latter approach, model mixtures of distributions of von Mises-Fisher (Banerjee *et al.*, 2005)have been used, proposing variants of the EM algorithm : EM_{vMF}, the CEM_{vMF}, the SEM_{vMF} and the SAEM_{vMF}. In the same context, the problem of finding the number of the components in the mixture and the choice of the model, using some information criteria {Bic, Aic, Aic3, Aic4, AICC, AICU, CAIC, Clc, Icl-Bic, LI, Icl, Awe } have been discussed. The study concludes with a comparison of the used vMF model with a simpler exponential model (Phuong et Vinh, 2008). In the latter, it is assumed that all data are distributed on a hypersphere of a predetermined radius greater than one, instead of a unit hypersphere in the case of the vMF model. An improvement of this method based on the estimation step of the radius in the algorithm NEM_{ρ} has been proposed : this allowed us in most of our applications to find the best partitions ; we have developed also the NCEM_{ρ} and NSEM_{ρ} algorithms. The algorithms proposed in this work were performed on a variety of textual data, genetic data and simulated data according to the vMF model ; these applications

gave us a better understanding of the different studied approaches throughout this thesis.

Table des matières

Table des figures

Liste des tableaux

Notations

d : nombre de colonnes (variables, attributs, mots, gènes)

n : nombre de lignes (objets, observations, cas, documents, tissus,...)

x_i^j : valeur prise par l'individu i pour la variable j

$\|.\|$: norme L_2

$\langle ., . \rangle$: produit scalaire

$\mathbf{x}_i = \left(x_i^1, ..., x_i^d\right) \in \mathbb{R}^d$, $\|x_i\| = 1$

I : ensemble des individus

J : ensemble des variables

$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$: matrice de données définie dans $I \times J$

g : nombre total de classes

P : partition des données en g classes $\{P_1, ..., P_g\}$

$S^{(d-1)}$: hypersphère de rayon 1

$S_\rho^{(d-1)}$: hypersphère de rayon ρ

$(z_{ik}; i = 1 \ldots n; k = 1, \ldots g)$: matrice de classification définie par $z_{ik} = 1$ si x_i appartient à la $k^{\text{ème}}$ classe et 0 sinon

$\mathbf{y} \in \mathbb{R}^n \times \mathbb{R}^{d+g}$ est le tableau de données complétées $(\mathbf{x}, \mathbf{z})$

θ : vecteur paramètre du mélange de lois de probabilités

t_{ik} : probabilité a posteriori, probabilité qu'un individu i appartient à la classe k, $\sum_k t_{ik} = 1$

$L(\theta)$: vraisemblance et log-vraisemblance du mélange

$N(\mu, V)$: loi normale de moyenne μ et de variance V

$M(\mu, \xi)$: loi de von Mises-Fisher de moyenne μ et de concentration ξ

$NEM_\rho(\mu, \rho)$: loi exponentielle normalisée de moyenne μ et de rayon ρ

Γ : fonction gamma.

I_ν : fonction de Bessel du premier type et d'ordre ν.

Abréviations

ML : Maximum de log-vraisemblance

CML : Maximum de log-vraisemblance complétée

Acc : classification occurrence

SPK-means : algorithme $k$means sphérique

EM : algorithme expectation maximisation

CEM : algorithme classification EM

SEM : algorithme stochastique EM

SAEM : simulated annealing (hybride) EM

vMF : modèle de von Mises-Fisher

EM_{vMF} : algorithme EM sur modèle de distribution de von Mises-Fisher

CEM_{vMF} : algorithme CEM sur modèle de distribution de von Mises-Fisher

SEM_{vMF} : algorithme SEM sur modèle de distribution de von Mises-Fisher

SAEM_{vMF} : algorithme SAEM sur modèle de distribution de von Mises-Fisher

NEM : algorithme EM sur modèle de distribution exponentielle normalisée

NEM_{ρ} : algorithme EM sur modèle de distribution exponentielle normalisée de rayon ρ

CEM_{ρ} : algorithme CEM sur modèle de distribution exponentielle normalisée de rayon ρ

SEM_{ρ} : algorithme SEM sur modèle de distribution exponentielle normalisée de rayon ρ

Introduction générale

La classification a un rôle majeur dans différentes disciplines. Son principal objectif consiste en la recherche des objets en classes homogènes. Plusieurs applications d'exploration de données à grande échelle, tels que la catégorisation de textes et l'analyse de l'expression des gènes, portent sur des données de grande dimension qui sont aussi intrinsèquement de nature directionnelle. Souvent de telles données sont L_2 normalisées de sorte qu'elles reposent sur la surface d'une hypersphère unitaire. Par exemple, des études dans le domaine de recherche documentaire démontrent de façon convaincante que la similarité cosinus est très efficace dans la classification des documents de texte. Dans ce domaine, il existe une justification empirique montrant que la normalisation des vecteurs de données permet d'éliminer les biais induits par la longueur d'un document et de fournir des résultats meilleurs Salton et Buckley (1988). Par conséquent, les données textuelles ou issues de la bioinformatique, où les mesures de similarité tel que cosinus, possèdent intrinsèquement des caractéristiques de type "directionnel" et sont donc mieux modélisées comme étant des données directionnelles Mardia et Jupp (2009).
Dans son usage principal, le clustering prend en charge deux approches classiques principales :

– La première est géométrique : la qualité de la segmentation dépend de la distance choisie ; l'idée naturelle est de résumer chaque classe de données par la plus proche moyenne directionnelle par rapport à une distance appropriée. Cet algorithme a fait l'objet de plusieurs études de recherche, parmi lesquelles des propositions de méthodes plus adéquates au type de données à classifier. Lelu (Lelu, 1993) a proposé la méthode des $k$means axiales en passant par une normalisation des données et en utilisant le produit scalaire comme mesure de similarité qui est très rapide dans le cas de données textuelles où la matrice des données est creuse en général. Une autre extension du $k$means : le $k$means sphérique (Dhillon et Modha, 2001) très similaire au précédent, il traite des données qui se trouvent sur la sphère unitaire en utilisant le cosinus de similarité dont le principal avantage est qu'il est indépendant de la taille des données, ce qui permet de comparer des documents de

tailles différentes et d'importance comparable comme par exemple un texte et son résumé. Le regroupement des données sur une hypersphère $S^{(d-1)}$ est assez fréquent, les utilisateurs du $k$means sphérique (Dhillon et Modha, 2001) sont habitués au cosinus de similarité, l'algorithme fournit g moyennes directionnelles en maximisant un critère géométrique :

$$W = \sum_{k=1}^{g} \sum_{\mathbf{x}_i \in P_k} \mathbf{x}_i^t \mu_k$$

Cependant, ces algorithmes géométriques, peuvent fournir des résultats décevants surtout pour des tableaux de données de petites tailles ou des classes assez mélangées et tendent à avoir des optimums locaux loin de la meilleure solution.

– La seconde approche est probabiliste, elle est devenue comme une approche standard (McLachlan et Peel, 2000) et couvre les méthodes de classification les plus largement utilisées. Dans cette approche, les données sont supposées provenir d'un mélange de g composants qui sont modélisés par une distribution de probabilités. Cette approche peut prendre en charge plusieurs situations, en fonction des paramètres du modèle, pour obtenir une meilleure description d'une population hétérogène en considérant un modèle sélectionné.

  Le regroupement basé sur un modèle de mélange considère deux approches : le maximum de vraisemblance (ML) et sa version classifiante (CML). La première est basée sur la maximisation de la vraisemblance des données observées, et la seconde est basée sur la maximisation de la vraisemblance des données complétées. Ces procédures peuvent être effectuées respectivement en utilisant l'algorithme EM (Dempster *et al.*, 1977) et l'algorithme CEM (Celeux et Govaert, 1992).

  L'approche Classification CEM est basée sur la maximisation de la vraisemblance classifiante et s'attaque directement au problème de la classification, l'approche Estimation EM est basée sur la maximisation de la vraisemblance, elle consiste d'abord à estimer les paramètres du modèle puis d'en déduire les classes. Grâce à l'algorithme d'estimation EM, particulièrement adapté à cette situation, les modèles de mélange ont fait l'objet de nombreux développements en statistiques et notamment en classification automatique.

Dans ce travail, nous étudions la classification automatique sous les deux approches : géométrique et probabiliste pour les données directionnelles. Pour nos applications on a traité deux groupes de données connus, il s'agit des données textuelles et des données génétiques. Notre travail est basé sur un modèle de mélange d'une loi de probabilité bien adapté au type de données directionnelles : le modèle de distribution de von Mises-Fisher (Dhillon et Sra, 2003). La loi de von Mises-Fisher est une loi multidimensionnelle établie spécialement pour l'étude des données directionnelles.

L'objectif principal de notre travail consiste dans un premier lieu à offrir de nouvelles extensions, soit le Stochastique EM (SEM) (Broniatowski *et al.*, 1983; Celeux et Diebolt, 1986, 1987) et le Simulated Annealing EM (SAEM) (Celeux *et al.*, 1995; Celeux et Diebolt, 1992; Celeux *et al.*, 1991) ; nous abordons ensuite, le problème du choix du modèle et du nombre de classes détecté dans un mélange de distributions de vMF, ceci en utilisant des critères d'information connus ; enfin, nous comparons le modèle de vMF à un modèle exponentiel qui est de type similaire ; les résultats obtenus sont intéressants. Le but de ce travail de recherche est de mettre en cause la classification automatique de données directionnelles, pour cela, nous nous intéressons au modèle de mélange de lois de von Mises-Fisher, en utilisant plusieurs outils statistiques et probabilistes, des algorithmes de type EM adaptés : EM_{vMF}, CEM_{vMF}, SEM_{vMF} et SAEM_{vMF} ; des critères d'information : Bic, Aic, Aic3, Aic4, Aicc, Aicu, Caic, Clc, Icl-Bic, Ll, Icl, Awe ; aussi des outils informatiques, soit le logiciel Matlab qui nous a servi comme langage d'environnement interactif pour toutes nos applications et simulations. Ce manuscrit est organisé en sept chapitres :

1. Le chapitre 1 présente la classification sous l'approche géométrique, où l'on donne des concepts de quelques méthodes classiques.
2. Le chapitre 2 présente la classification sous l'approche probabiliste, on définit les modèles de mélanges de probabilités et les quelques algorithmes utilisés dans ce manuscrit pour la classification automatique, à savoir EM, CEM, SEM, SAEM.
3. Le chapitre 3 aborde les statistiques des données directionnelles et quelques lois de probabilités adaptées.
4. Le chapitre 4 expose et compare deux méthodes de type $k$means conçues pour le traitement de données directionnelles : il s'agit du $k$means axial et du $k$means sphérique. Les comparaisons numériques ont été effectuées sur des données simulées.
5. Le chapitre 5 définit le modèle de mélange de lois de von Mises-Fisher, et son utilité en classification automatique, en expliquant les approches existantes à savoir EM_{vMF} et CEM_{vMF}. Nous proposons deux autres extensions : le SEM_{vMF} et le SAEM_{vMF} que nous évaluons sur des données simulées et réelles .
6. Le chapitre 6 étudie le choix du modèle et du nombre de classes dans le cas vMF, ceci en utilisant quelques critères d'information, des applications numériques sur des données obtenues par des simulations de Monté Carlo ont servi à l'évaluation.
7. Le chapitre 7 est consacré au modèle de mélange de lois exponentielles normalisées, une nouvelle approche basée sur le calcul du rayon ρ est révélée, ensuite comparée au modèle étudié dans les chapitres précédents. Des applications sur des données simulées ont confirmé l'efficacité de cette approche.

Enfin, nous terminons par une conclusion avec une perspective importante des travaux étudiés.

Chapitre 1

Classification automatique, méthodes géométriques

La classification automatique ou non supervisée représente un outil incontournable dans la fouille de données. En effet, avec l'avènement des *Big Data* on assiste à un engouement pour les différentes méthodes de classification cherchant à subdiviser l'ensemble des données en classes homogènes. On peut définir cet objectif par « le processus d'organisation des objets dans des groupes dont les membres sont semblables en quelque sorte ». Une classe est donc une collection d'objets qui sont similaires entre eux et sont dissimilaires aux objets appartenant à d'autres classes ; on peut voir ceci avec un exemple graphique simple illustré dans figure 1.1 :

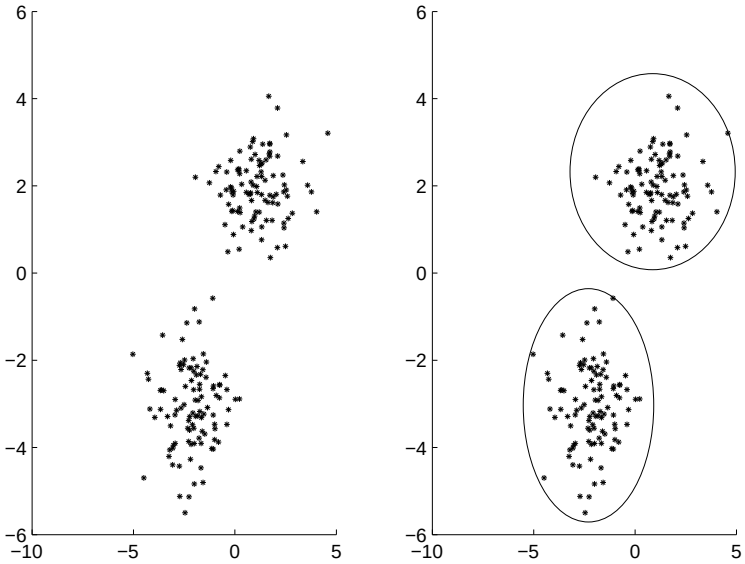

FIGURE 1.1 – Visualisation de deux classes dans un groupe de points aléatoires

Dans figure 1.1, on peut facilement identifier les deux groupes visuellement ; ou même en utilisant un critère de distance mesurant la similarité entre les individus, c'est une classification basée sur la notion de proximité. Il y a aussi le regroupement conceptuel : deux ou plusieurs objets appartiennent à la même classe si celle-ci définit un concept commun pour tous les objets. En d'autres termes, les objets sont regroupés selon leur accord aux concepts descriptifs du même groupe, par exemple des individus différents qui encouragent la même équipe sportive appartiennent à la même classe.

Donc, l'objectif de la classification est de déterminer les groupements intrinsèques dans un ensemble d'observations non étiquetées. Ce problème est bien classique et il n'existe pas actuellement une méthode de classification qui donne toujours la meilleure partition. Par conséquent, c'est l'utilisateur qui doit choisir la méthode, de telle sorte que le résultat de la classification réponde à ses besoins, autrement dit la structure en classes obtenue soit profitable.

La classification est utile dans plusieurs domaines de la vie, ci-dessous quelques exemples :

- Médecine : classification de gènes, maladies, etc.
- Éducation : détection des groupes d'élèves de mêmes capacités, classification des outils et besoins éducatifs pour l'amélioration du système éducatif.
- Marketing : recherche des groupes de clients avec un comportement similaire au regard d'une enquête ou une récolte de tickets de caisse.
- Biologie : classification des plantes et des animaux compte tenu de leurs caractéristiques.
- Bibliothèques : classification des livres et commandes.
- Assurance : identification des groupes de preneurs d'assurance d'automobile ayant un taux moyen de revendication élevé ; identification des fraudes.
- Planification de ville : identification des groupes de maisons en fonction de leur type, valeur et emplacement géographique.
- Tremblement de terre : la classification des épicentres des séismes, pour identifier les zones dangereuses.
- Web : classification des documents, classification de données via les fichiers log (Web usage mining) pour découvrir les groupes de modes d'accès similaires.
- Analyse textuelle : classification de documents ou mots (Text mining)

Parmi les problèmes connus en classification automatique, on cite :

- Taille des données et grande dimension : les données actuelles sont de plus en plus volumineuses et de grande dimension notamment dans le domaine de la génomique ; ce qui représente un temps de calcul très important et souvent non envisageable.
- Choix de la mesure de proximité : les méthodes de classification dépendent directement ou implicitement d'une mesure de similarité, son choix conditionne la qualité

de la partition et particulièrement dans la grand dimension.
- Initialisation : les algorithmes employés dépendent de la situation initiale qui peut être surmontée par différentes stratégies.
- Sélection de modèle : hormis la classification hiérarchique, la plupart des méthodes de classification considèrent que le nombre de classes est supposée connu. Une contrainte qui peut être surmontée par l'utilisation de critères de choix de modèles.

Les algorithmes de classification habituellement utilisés peuvent être de type géométrique : de partitionnement exclusif, flou ou bien hiérarchique ; ou encore de type probabiliste. Dans le partitionnement exclusif, les données sont regroupées d'une manière exclusive, de sorte que si une certaine donnée (individu ou variable) appartienne à un groupe, elle ne peut pas être incluse dans un autre. Au contraire, dans la classification floue, chaque donnée peut appartenir à un ou plusieurs groupes avec différents degrés d'appartenance.

Dans ce chapitre, nous rappellerons les définitions et propriétés des différentes méthodes de classification sous l'approche géométrique, c'est une étape nécessaire pour la suite de notre travail.

1 La classification hiérarchique

Il existe essentiellement deux approches : La classification descendante où l'on divise l'ensemble I en classes, puis on recommence sur chacune de ces classes et ainsi de suite jusqu'à ce que les classes soient réduites à des singletons. La classification ascendante, cette fois on part de la partition de I où chaque classe est un singleton, on procède alors par fusion successive des classes qui se ressemblent le plus, jusqu'à obtenir une seule classe, c'est à dire l'ensemble I lui même. Cette procédure est la plus utilisée des deux. Les exposés les plus anciens sont ceux de (Sokal et Sneath, 1963), puis de (Lance et Williams, 1967).

1.1 Définition

I étant un ensemble fini, un ensemble H de parties non-vides de I est une hiérarchie sur I si :

- $I \in H$
- $\forall x \in I \quad \{x\} \in H$
- $\forall h, h' \in H \quad h \cap h' = \emptyset$ ou $h \subseteq h'$ ou $h' \subseteq h$

Une hiérarchie est souvent représentée par une structure arborescente représentée par un arbre hiérarchique dit aussi dendrogramme (1.2).

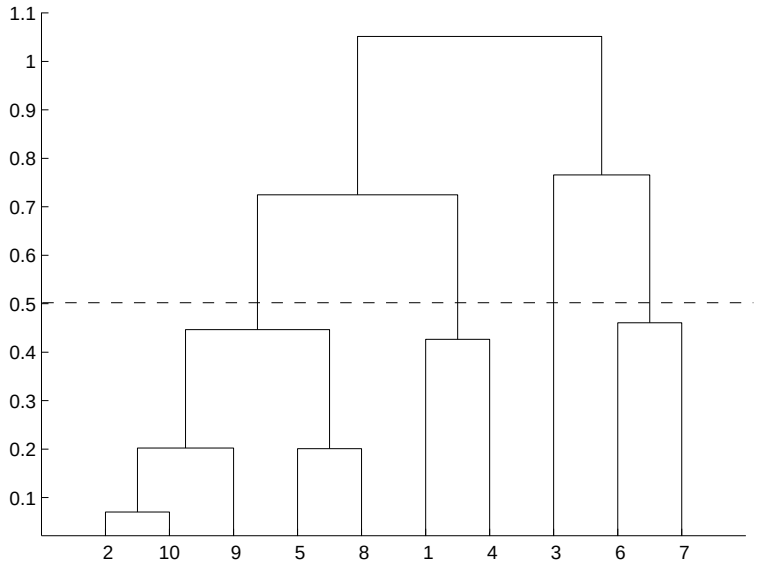

FIGURE 1.2 – Dendrogramme

1.2 L'algorithme

I étant l'ensemble à classifier et D une mesure de dissimilarité sur cet ensemble I, on définit à partir de D, une mesure de proximité d entre les parties de I. Cette mesure est en réalité une mesure de dissimilarité sur l'ensemble des parties de I définies à partir de la mesure de la dissimilarité D sur I ; d est appelée critère d'agrégation.

Algorithme 1: classification ascendante hiérarchique (CAH)

Entrée : partition des singletons.

Répéter

1 : On regroupe les deux classes les plus proches au sens de d,

2 : On recalcule les distances entre la nouvelle classe et les anciennes classes qui n'ont pas été regroupées.

jusqu'à *ce que le nombre de classes* $= 1$;

1.3 Critères d'agrégations

Un critère d'agrégation est une fonction numérique mesurant la qualité de l'homogénéité entre les classes. Il existe de nombreux critères d'agrégation, les plus utilisés sont :

– Le critère du saut minimal (Single linkage), (Sneath, 1957) :

$$d\left(h, h'\right) = \inf\{D\left(x, y\right) | x \in h \text{ et } y \in h'\}$$

d'où la formule de récurrence de Lance william : $d\left(h, h' \cup h''\right) = \inf\{d\left(h, h'\right), d\left(h, h''\right)\}$

– Le critère du saut maximal (complete linkage) : dit aussi du diamètre

$$d\left(h,h'\right)=\max\{D\left(h,h'\right)|x\in h \text{ et } y\in h'\}$$

d'où la formule de récurrence de Lance william : $d\left(h,h'\cup h''\right)=\max\ d\left(h,h'\right),d\left(h,h''\right)$

– Le critère de la distance moyenne (average linkage) :

$$d\left(h,h'\right)=\frac{1}{n_h n_{h'}}\left[\sum_{x\in h}\sum_{y\in h'}p_x p_y D\left(x,y\right)\right]$$

d'où la formule de récurrence de Lance william $d\left(h,h'\cup h''\right)=\dfrac{n_{h'}d\left(h,h'\right)+n_{h''}d\left(h,h''\right)}{n_{h'}+n_{h''}}$

où n_h et $n_{h'}$ désignent respectivement les cardinalités des deux ensembles h et h', p_x et p_y sont les poids associés respectivement aux points x de h et y de h'. Sur une hiérarchie H, on peut définir un indice i par une fonction strictement croissante de H dans $\mathbb{R}^+$, qui vérifie :

– $h\subset h'$ et $h\neq h'\Longrightarrow i(h)<i(h')$.
– $x\in\Omega\ \ i(\{x\})=0$.

A chaque hiérarchie indicée (H,i), on lui associe le dendrogramme (1.2). Il est possible d'associer à chaque niveau de (H,i) une partition en g classes, donc une hiérarchie fournit une suite de partitions emboitées, $g\in\{2,\ldots,n\}$, sur le dendrogramme, en coupant l'arbre selon une horizontale, on obtient une partition en recueillant les morceaux. Notons qu'il existe des critères pour une découpe automatique. Par ailleurs, on peut associer à une hiérarchie un indice i construit à partir de la distance D par :

– l'indice d'un singleton est égal à 0.
– A chaque classe construite au cours de l'algorithme, la valeur de l'indice associé est la distance D entre les deux classes fusionnées pour fournir cette même classe.

En analysant l'évolution du critère, nous sommes capables de déterminer un nombre de classes approprié. Donc, à l'inverse des méthodes de classification directe, nous n'avons donc pas besoin ici de la connaissance a priori du nombre de classes.

L'algorithme de la classification hiérarchique est parfait dans le cas de tableaux de taille raisonnable. Mais dans un contexte de fouille de données, un tel algorithme est assez peu utilisé, on se contente souvent de l'appliquer sur des échantillons de l'ensemble des données ou encore sur des résumés de données obtenus précédemment avec une autre méthode. On peut citer l'algorithme mixte proposé par (Wong, 1982) qu'on décrira ultérieurement.

2 Méthodes de partitionnement

Contrairement aux méthodes hiérarchiques, les méthodes de partitionnement proposent directement une partition en classes sachant un nombre de classes fixé au départ. Tout d'abord rappelons la définition d'une partition P en g classes définie sur un ensemble fini I. Cette partition est un ensemble de parties non-vides de cet ensemble vérifiant :

- $\cup_{k=1}^{g} P_k = I$
- $P_i \cap P_j = \emptyset$, $\forall i \neq j$
- Les éléments de P sont les classes de la partition.
- Le poids de chaque classe P_k est : $p_k = \sum_{x_i \in P_k} p_i$ Les p_i sont les poids respectifs associés aux points x_i.

Plusieurs méthodes ont été proposées : des méthodes qui recherchent la partition optimisant une fonction numérique définie sur l'ensemble des partitions, appelée en général critère de classification (Jensen, 1969; Régnier, 1965; Ruspini, 1969). Des méthodes algorithmiques, telles que la méthode de (Ball et J.Hall, 1965) qui dépend d'un certain nombre de seuils donnés a priori. Ou encore, des algorithmes célèbres de type $k$means proposés par (Forgy, 1965) et(McQueen, 1967). Ces dernières méthodes mesurent la qualité d'une partition par la somme des inerties des classes par rapport à leur centre de gravité. Rappelons que ce critère ne permet pas de comparer des partitions n'ayant pas le même nombre de classes. Ces dernières sont souvent utilisées et le critère à optimiser

$$W(P, L) = \sum_{k=1}^{g} \sum_{x_i \in P_k} \delta^2(x_i, \mu_k)$$

où δ est une distance euclidienne, μ_k est le centre de gravité de la classe P_k et $L = \{\mu_1, \ldots, \mu_g\}$

Nous décrirons ci-après et en détail ce type de méthodes de partitionnement.

2.1 Méthode des centres mobiles ou $k$means

L'algorithme des centres mobiles ou communément connu sous le nom de $k$means et qui repose sur les centres de gravité des classes est le suivant :

Si on note $P = \{P_1, \ldots, P_g\}$ une partition de I en g classes et $L = \{\mu_1, \ldots, \mu_g\}$ un ensemble de g points de $\mathbb{R}^d$, l'algorithme construit une suite que l'on notera :

$L^0 \longrightarrow P^0 \longrightarrow L^1 \longrightarrow P^1 \longrightarrow \ldots \longrightarrow P^{m-1} \longrightarrow L^m \longrightarrow P^m \longrightarrow \ldots$

La qualité d'un couple partition-centres est mesurée par le critère $W(P, L)$. L'algorithme s'arrête soit lorsque deux itérations successives conduisent à la même partition, soit lorsque le critère cesse de décroitre de façon sensible, soit lorsqu'un nombre maximal d'itérations a été fixé a priori ; le résultat dépend du choix initial des centres.

Algorithme 2: centres mobiles

Entrée : g : le nombre de classes ;

x : la matrice des données.

Initialisation : Tirage au hasard de g points de I qui forment les centres initiaux des g classes.

Sortie : La partition

Répéter

1 : Construction de la partition en affectant chaque point de I à la classe dont il est le plus près du centre,

2 : Les centres de gravité de la partition qui vient d'être calculée deviennent les nouveaux centres de gravité.

Fin tant que convergence.;

2.1.1 La méthode des nuées dynamiques

Sous le nom de méthode des nuées dynamiques, (Diday, 1972) a proposé une technique de classification qui présente de nombreux avantages. L'idée de base de cette méthode est la suivante : au lieu de regrouper les éléments de l'ensemble I à classifier autour d'éléments qui n'appartiennent d'ailleurs pas nécessairement à l'ensemble I comme c'est le cas de $k$means. On fait un regroupement autour d'ensemble de plusieurs éléments, appelés noyaux, qui seront des parties de I. Une classe d'une partition de I, au lieu d'être représentée par un seul élément, tel son centre de gravité, elle le sera par plusieurs de ces éléments (le noyau de la classe) ; s'ils sont bien choisis, ces éléments seront typiques de la classe et en formeront un résumé plus riche que peut l'être un centre de gravité. Cette façon de procéder, qui admet de nombreuses variantes, présente bien des avantages, dont principalement :

- Une grande souplesse : des contraintes peuvent être imposées aux noyaux dont les éléments par exemple peuvent être choisis parmi des éléments particuliers de I, ou simplement de même nature que les éléments de I.
- Des facilités au niveau de l'interprétation des résultats qui peut être faite en examinant les seuls noyaux.

Pour ces raisons, la plupart des méthodes de classification automatique géométrique proposées jusqu'à présent, reposent sur le principe des nuées dynamiques. Ce principe a été repris dans (Diday, 1979) sous la forme suivante :

Considérons un ensemble I de n individus représentés par un ensemble de n points inclus dans un espace E (par exemple $\mathbb{R}^d$). On définit l'espace des noyaux θ qui seront associés aux classes de la partition comme caractéristique propre à chacune d'entre elles.

On définit le critère W de la classification par la fonction suivante :

$$W(P,L) = \sum_{k=1}^{g} \sum_{x_i \in P_k} D(x_i, \lambda_k)$$

où $P = (P_1, \ldots, P_g)$ est une partition de l'ensemble I. $L = (\lambda_1, \ldots, \lambda_g)$ est l'ensemble des noyaux des classes de la partition P.

L'algorithme construit de manière itérative une suite de partitions et de noyaux : P^0, L^0, P^1, $L^1, \ldots,$ P^m, L^m en minimisant à chaque étape le critère $W(P,L)$. Cette construction repose sur la définition de deux fonctions

1. une fonction d'affectation f qui consiste à affecter chaque individu à l'une des classes de la partition de manière à optimiser à chaque fois le critère $W(f(L), L)$:

   $$f(L) = f(\lambda_1, \ldots, \lambda_g) = P = (P_1, \ldots, P_g)$$

   où $P_k = \{x \in I | D(x, \lambda_k) \leqslant D(x, \lambda_{k'})$ avec $k < k'$ en cas d'égalité. La classe P_k sera donc constituée des éléments de I qui seront plus proches de λ_k au sens de la distance D que de tout autre noyau de L.

2. une fonction de représentation g qui permet de déterminer les noyaux de la partition de manière à optimiser, à chaque fois, le critère $W(P, g(P))$.

   $$g(P) = g(P_1, \ldots, P_g) = (\lambda_1, \ldots, \lambda_g) = L$$

L'algorithme utilisé dans la méthode des nuées dynamiques consiste en la construction de deux suites :

- $\{V_m | m \in \mathbb{N}\}$: suite de $L_k \times P_k$, c'est à dire que : $\forall m,\ V_m = (L^m, P^m)$.
- $\{U_m | m \in \mathbb{N}\}$: suite de valeurs du critère sur les V_m, c'est à dire :

$$\forall m \quad U_m = W(L^m, P^m) = W(V_m)$$

Soit P^0 une partition initiale quelconque prise au hasard ou choisie, si L^0 est l'ensemble des noyaux qui lui sont associés ($L^0 = g\left(P^0\right)$) alors :

$$V_0 = \left(L^0, P^0\right) = \left(g\left(P^0\right), P^0\right)$$

La suite V_m est ensuite définie par récurrence : si $V_m = (L^m, P^m)$ alors
$V_{m+1} = \left(L^{m+1}, P^{m+1}\right)$ où $P^{m+1} = f(L^m)$ et $L^{m+1} = \mathrm{g}\left(P^{m+1}\right) = g \circ f(L^m)$.

Sous certaines conditions (Diday, 1972; Govaert, 1975; Schwarz, 1974), la suite $U_m = W(V_m)$ décroît, converge et atteint sa limite :

$$\exists\ M \in \mathbb{N} : \forall\ m \geqslant M\ \ U_m = U^*$$

le couple $V^* = (L^*, P^*)$ tel que $W(V^*) = U^*$ sera appelé optimum local.

3 Classification floue

La classification floue est un processus de regroupement d'éléments dans un ensemble flou (Zadeh, 1965), c'est une méthode de classification non-supervisée. Contrairement à la classification dure, cette méthode dite floue associe à chaque objet un degré d'appartenance à une classe donnée. Pour chaque objet, la somme de ces degrés est égale à un. L'un des algorithmes de classification floue le plus utilisé est le fuzzy *c*means (FCM) (Ball et J.Hall, 1967), similaire au *k*means décrit précédemment. Cet algorithme est basé sur la minimisation du critère (Dunn, 1974) :

$$W(P;L) = \sum_{k=1}^{g} \sum_{i=1}^{n} \mu_{ik}^2 d_M^2 \left(\mathbf{x}_i, \mathbf{g}_k\right)$$

$P = [\mu_{ik}]$ correspond à une partition floue en g classes ($\mu_{ik} \in [0;1]$ $\forall i;k$ et $\sum_{k=1}^{g} \mu_{ik} = 1$ $\forall i$), en utilisant l'algorithme FCM.

Algorithme 3: fuzzy *c*means (FCM)

Initialisation : g centres choisis au hasard,

Sortie : La partition

Répéter

1 : Calcul des μ_{ik} : degrés d'appartenance aux classes,

2 : Calcul des $\mathbf{g}_k$: centres des classes

jusqu'à *jusqu'à la convergence de* W;

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

4 Formes fortes et groupements stables

Les algorithmes d'agrégation autour des centres mobiles convergent vers des optimums locaux dépendant en général des premiers centres choisis (ce n'est pas satisfaisant). La procédure de recherche de regroupements stables (Diday, 1972) permet de remédier à cet inconvénient ; à partir de plusieurs centres on effectue plusieurs partitions, les groupements stables retenus sont les ensembles d'individus qui ont toujours été affectés à une même classe dans chacune des partitions, mais ceci n'est qu'une exploration des zones de forte densité dans l'espace. En pratique, on peut utiliser un nombre l de groupements stables d'effectifs importants, ainsi les g classes seront obtenues par réaffectation des individus restants aux groupements dont ils sont plus proches.

5 Classification mixte

Les algorithmes de classification sont plus ou moins bien adaptés à la gestion d'un nombre important d'objets à classer. La méthode d'agrégation autour des centres mobiles offre des avantages incontestables puisqu'elle permet d'obtenir une partition sur un ensemble volumineux de données à un faible coût, mais elle présente l'inconvénient de produire des partitions dépendantes des premiers centres choisis et celui de fixer a priori le nombre de classes. Au contraire, la classification hiérarchique est une famille d'algorithmes qui donnent toujours les mêmes résultats à partir des mêmes données. De plus, ces algorithmes donnent des indications sur le nombre de classes à retenir, mais ils sont mal adaptés aux vastes recueils de données.

Plusieurs auteurs se sont inspirés des deux techniques précédentes pour déterminer le nombre de classes préalable en fournissant des partitions de vastes ensembles de données, par exemple la méthode hybrid clustering (Wong, 1982). Ce type de méthode se déroule en trois temps :

1. Réallocation itérative : où l'on effectue pour commencer une classification des individus en un grand nombre de classes avec choix aléatoire des centres initiaux (partition initiale). Ce premier classement permet de réaliser par la suite une classification ascendante hiérarchique sur un nombre plus restreint d'individus.
2. Classification ascendante hiérarchique, dont le dendrogramme suggérera éventuellement le nombre de classes finales à retenir.
3. Appliquer une deuxième fois la méthode des nuées dynamiques (classification), pour améliorer la classification obtenue avec la CAH, on effectue une deuxième réallocation itérative sur l'ensemble, éventuellement pondéré, des individus avec comme noyaux de départ les centres déterminés par la classification hiérarchique précédente. Cette méthode vise à augmenter l'inertie inter-classe. De ce point de vue, la classification ne peut donc que s'améliorer.

6 Mise en œuvre

On a vu que le problème initial de la classification se traduit en un problème d'optimisation d'un critère choisi en fonction d'une mesure de dissimilarité ce qui exprime l'importance de ce choix ; la méthode des nuées dynamiques se révèle être une bonne approche pour proposer de tels critères.

La recherche d'une partition optimisant un critère choisi nécessite le choix d'un algorithme d'optimisation, là on peut penser à un algorithme qui compare toutes les partitions

possibles mais malheureusement cette idée n'est pas pratique, car le nombre de partitions devient vite très grand, même pour un tableau de données de dimension modérée. Le nombre de partitions d'un ensemble I de n éléments en g classes est donné par la formule suivante :

$$S(n,g) = \frac{1}{g!}\sum_{i=1}^{g} C_g^i\,(-1)^{i-1}\,i^n$$

ceci est remarquable, par exemple si l'on prend $(n,g) = (10,3)$, le nombre de partitions possibles est 9330 et si on a $(n,g) = (100,3)$ le nombre de partitions possibles devient 8589.6253×10^{43} ; le plus souvent il est impossible de trouver un algorithme qui donne un optimum global. L'algorithme des nuées dynamiques, comme d'autres algorithmes d'optimisation locale, fournit une suite de partitions améliorant à chaque fois le critère. Cet enchaînement s'arrête lorsqu'on n'a pas d'amélioration considérable dans la valeur de ce critère.

Le résultat donné par un algorithme d'optimisation locale dépend de la partition initiale, ici on exploitera les optimums locaux obtenus à partir de plusieurs partitions initiales différentes :

- On applique l'algorithme plusieurs fois en partant de partitions initiales différentes. Différents optima sont atteints, si possible on retient directement l'optimum global, sinon on peut utiliser la méthode des formes fortes.
- On sélectionne au départ une bonne initialisation à l'aide d'informations supplémentaires ou à l'aide d'une procédure automatique.

Généralement, le critère à optimiser n'est pas indépendant du nombre de classes qu'on désire avoir, par exemple dans une hiérarchie, plus le nombre de classes est grand, plus le critère d'inertie inter-classes est petit, d'où la nécessité de fixer un nombre de classes avant d'appliquer l'algorithme. Pour résoudre ce problème très difficile, plusieurs solutions ont été proposées :

- Avoir une idée du nombre de classes désiré.
- Appliquer la méthode du coude, où on cherche le bon nombre de classes qui réalise la meilleure valeur du critère parmi différents nombres par rapport au même critère.
- Ajouter des contraintes supplémentaires, comme par exemple : le nombre d'individus par classe.

Dans le chapitre qui suit, nous présentons l'approche probabiliste du problème de classification automatique, en définissant les modèles de mélange et quelques algorithmes de type EM, que nous utilisons par la suite.

Chapitre 2

Classification automatique, modèles de mélange

En classification automatique, où le but est de déceler des groupes homogènes dans une population, il paraît naturel de supposer à l'avance que la population est formée de g sous populations ou classes ayant des caractéristiques différentes puis d'estimer les caractéristiques de ces classes. Les modèles de mélange finis permettent de formaliser cette idée intuitive et sont tout particulièrement adaptés au problème de classification.

L'attention accordée aux modèles de mélange finis au fil des ans témoigne de leur utilité comme une méthode de modélisation extrêmement souple, et de son importance tant d'un point de vue théorique que pratique (McLachlan et Peel, 2000). Dans ce contexte, une grande variété de techniques peut être considérée selon les problématiques, telles que les méthodes graphiques, les méthodes des moments, les méthodes du maximum de vraisemblance et les approches bayésiennes. Depuis plus de 30 ans, des progrès considérables ont été réalisés dans ce domaine, en particulier via la méthode du maximum de vraisemblance, grâce aux travaux de Dempster et al. (Dempster *et al.*, 1977) et à l'introduction de l'algorithme EM.

Nous utilisons dans notre travail les modèles de mélange de lois de probabilités pour la classification automatique de données directionnelles, nous exposons dans ce chapitre le modèle de mélange de lois de probabilités, son utilisation en présentant le principe du ML et du CML, ainsi que l'algorithme EM et ses variantes à savoir par exemple CEM, SEM, SAEM.

1 Une approche classique (ML)

Les modèles de mélange considèrent que les données sont générées à partir d'un mélange de différentes distributions de probabilités. Ces dernières, sont généralement, du même type. Le problème d'identification des composants d'un mélange par l'estimation des paramètres de ses distributions est devenu une approche classique, il fut étudié depuis longtemps dans le cadre univarié et multivarié (Day, 1969; Wolf, 1970). Sous cette approche, le tableau de données I est considéré comme un échantillon de taille n d'une variable aléatoire à valeurs dans $\mathbb{R}^d$ dont la loi de probabilité admet la fonction de densité suivante :

$$f(\mathbf{x}) = \sum_{k=1}^{g} \pi_k f(\mathbf{x}; \alpha_k)$$

$$\forall\ \mathbf{x} \in \mathbb{R}^d \quad \forall\ k = 1, ..., g \qquad 0 \leqslant \pi_k \leqslant 1 \text{ et } \sum_{k=1}^{g} \pi_k = 1$$

$f(\mathbf{x}; \alpha_k)$ est la densité de la distribution du $k^{\text{ème}}$ composant et α_k est la valeur du paramètre α dont dépend le $k^{\text{ème}}$ composant, π_k est la probabilité a priori d'apparition du $k^{\text{ème}}$ composant dans le mélange.

Il s'agit d'un problème d'estimation des paramètres : le nombre g de composants du mélange et les paramètres inconnus $(\theta_k = (\pi_k, \alpha_k) \quad k = 1, \ldots, g)$ au vu des g composants du mélange. L'approche au sens du maximum de vraisemblance via la maximisation de la vraisemblance est une approche naturelle, cette vraisemblance s'écrit :

$$L(\mathbf{x}; \theta) = \prod_{i=1}^{n} \sum_{k=1}^{g} \pi_k f(\mathbf{x}_i; \alpha_k)$$

$f(.; \alpha_k)$ est une loi de probabilité appartenant à une famille de lois identifiables. Dans son utilité générale et pour simplifier les calculs, la vraisemblance est remplacée par son logarithme, on gardera la même notation L pour la log-vraisemblance le long de notre travail :

$$L(\mathbf{x}; \theta) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{g} \pi_k f(\mathbf{x}_i; \alpha_k) \right\}$$

2 Approche classification (CML)

Dans cette approche, on cherche une partition $\mathbf{P}$ en g classes, g étant supposé connu. Pour ce faire, on utilisera le principe du maximum a posteriori à partir des paramètres estimés.

En suivant l'approche modèle fourni dans les représentations proposées par Celeux (Celeux, 1988) et Govaert (Govaert et Nadif, 1990), qui transforment le problème d'optimisation du critère de vraisemblance en un problème d'optimisation de critère de vraisem-

blance classifiante, chaque élément $\mathbf{x}_i$ est affecté à une classe k avec une probabilité 1. On est par conséquent, amené à maximiser le critère de vraisemblance classifiante suivant :

$$L_C(\mathbf{x}, \mathbf{P}; \theta) = \sum_{k=1}^{g} \sum_{i=1}^{n} z_{ik} \log \pi_k f(\mathbf{x}_i; \alpha_k)$$

tel que :

$$z_{ik} = \begin{cases} 1 & \text{si } x_i \in P_k \\ 0 & \text{sinon} \end{cases}$$

d'où :

$$L_C(\mathbf{x}, \mathbf{P}; \theta) = \sum_{k=1}^{g} \sum_{\mathbf{x}_i \in P_k} \log \pi_k f(\mathbf{x}_i; \alpha_k)$$

3 Présentation générale de l'algorithme EM

La maximisation de la log-vraisemblance dans le cas non-supervisé d'optimisation des modèles de mélange, conduit généralement à des équations de vraisemblance qui ne possèdent pas de solutions analytiques. Parmi les méthodes les plus utilisées pour résoudre ce problème l'algorithme itératif Expectation-Maximization, EM (Dempster *et al.*, 1977).

L'algorithme EM est une technique connue qui permet l'estimation des paramètres d'un mélange de distributions probabilistes, en maximisant la log-vraisemblance des données complétées $L(\mathbf{x}, \mathbf{y}; \theta)$, le tableau de données complétées notées $\mathbf{y} = (\mathbf{x}, \mathbf{z})$ et $\mathbf{z}$ n'est pas observé. On a :

$$L(\mathbf{x}; \theta) = L(\mathbf{y}; \theta) - L(\mathbf{y}; \mathbf{x}; \theta)$$

puisque la vraisemblance complète $L(\mathbf{y}; \theta)$ n'est pas calculable, (Dempster *et al.*, 1977) ont proposé une méthode itérative basée sur la maximisation de l'espérance conditionnelle de la vraisemblance à une valeur du paramètre θ.

L'algorithme EM construit itérativement une suite de paramètres $\{\theta^{(c)}\}$, maximisant à chaque itération (c) $L(\mathbf{x}; \theta)$ qui revient à maximiser :

$$Q(\theta, \theta') = E[L(y; \theta); \mathbf{x}; \theta']$$

$$\Leftrightarrow Q(\theta, \theta') = \sum_{k=1}^{g} \sum_{i=1}^{n} t_{ik} log \pi_k f(\mathbf{x}_i; \alpha_k), \tag{2.1}$$

où t_{ik} la probabilité conditionnelle qu'un individu i provienne du composant k.

$$t_{ik} = E(z_{ik}; \mathbf{x}; \theta') = p(\mathbf{x}_i \in P_k; \mathbf{x}; \theta') = p(z_{ik} = 1; \mathbf{x}; \theta')$$

$$t_{ik} = \frac{\pi_k f(\mathbf{x}_i; \alpha_k)}{\sum_{l=1}^{g} \pi_l f(\mathbf{x}_i; \alpha_l)}$$

Dans (Dempster *et al.*, 1977), il a été prouvé qu'en faisant croître $L(\mathbf{x}; \theta)$, la suite ainsi générée converge vers un maximum local (sous certaines conditions de régularité (Govaert, 2003)). L'algorithme 4 ci-dessous associé à EM alterne les deux itérations : Expectation et Maximisation, à partir d'une situation initiale $\theta^{(0)}$ (ou même une partition initiale).

Algorithme 4: L'algorithme EM

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Sortie : Les paramètres obtenus à la convergence constituant θ.

Initialisation : Choix aléatoire des paramètres du mélange $\theta_k, k = 1, \ldots, g$;

Répéter

1 : Étape Estimation E : Calcul des t_{ik} :

$$t_{ik} = \frac{\pi_k f(\mathbf{x}_i; \alpha_k)}{\sum_{l=1}^{g} \pi_l f(\mathbf{x}_i; \alpha_l)}$$

2 : Étape Maximisation M : Estimation des paramètres $\theta = \arg\max_{\theta'} Q(\theta, \theta')$.

jusqu'à *la convergence*;

A chaque itération (c), il est facile de montrer que les proportions du mélange sont estimées par :

$$\pi_k^{(c)} = \frac{n_k^{(c)}}{n}$$

$n_k^{(c)}$ est le cardinal de $P_k^{(c)}$: la classe obtenue à l'itération (c).

Les $\alpha_k^{(c)}$ sont à leurs tour estimés en fonction du modèle probabiliste choisi. Analytiquement ceci représente une solution de l'équation $\frac{\partial Q(\theta, \theta^{(c)})}{\partial \theta} = 0$.

Cet algorithme converge vers un maximum local ou un point selle (McLachlan et Krishnan, 1997).

Dans notre travail, pour chaque algorithme, la convergence est atteinte lorsqu'on aboutit à un état fixé au départ, par exemple : un nombre donné d'itérations ou la stationnarité. Dans la plupart du temps, pour l'algorithme EM on fixe le nombre d'itérations ou on définit un seuil de convergence.

Malgré son utilité, l'algorithme EM présente quelques inconvénients, par exemple les solutions estimées peuvent être fortement liées aux valeurs initiales. Il peut converger lentement lorsque les classes sont mal séparées. Il s'agit d'un algorithme itératif et la convergence vers un optimum global n'est pas garantie. La solution analytique dans certains cas est impossible (Wu, 1983). Dans certaines situations, il peut rester bloqué dans un point selle de la vraisemblance.

4 Quelques variantes de l'algorithme EM

Plusieurs variantes de l'algorithme EM ont été proposées pour pallier ses limitations. Nous résumons ici les variantes les plus pertinentes et les plus utilisées dans ce travail.

4.1 Algorithme CEM

L'algorithme CEM répond à l'approche CML des modèles de mélange, il a été proposé par (Celeux et Govaert, 1992), en introduisant une étape classification entre les deux étapes de l'algorithme EM, cette variante est une extension de l'algorithme géométrique des nuées dynamiques, à condition que les proportions du mélange soient égales. Dans ce cas les étapes maximisation et classification correspondent respectivement aux étapes représentation et affectation.

En gardant les deux étapes E et M telle qu'elles sont, l'algorithme 5 ci dessous procède de la manière suivante : à une itération (c), chaque individu i est affecté à la classe k dont la probabilité t_{ik} est maximale, cette dernière sera remplacée par $z_{ik} = 1$ si $\mathbf{x}_i$ est affecté à la classe k, 0 si non. A partir d'une partition initiale représentée par $\mathbf{z}^{(0)}$ (ou bien par $\theta^{(0)}$) ;

Algorithme 5: L'algorithme CEM

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Sortie : La partition : $P_1, ..., P_g$.

Initialisation : Choix aléatoire des paramètres du mélange $\theta_k, k = 1, \ldots, g$;

Répéter

1 : Étape Estimation E : Calcul des t_{ik} :

$$t_{ik} = \frac{\pi_k f(\mathbf{x}_i; \alpha_k)}{\sum_{l=1}^{g} \pi_l f(\mathbf{x}_i; \alpha_l)}$$

2 : Étape Classification C : Affectation des individus et construction des classes, telle que : $P_k = \{x_i | t_{ik} = \max_{k'} t_{ik'}\}$ suite au résultat :

$$z_{ik} = \begin{cases} 1 & \text{si } x_i \in P_k \\ 0 & \text{si non} \end{cases}$$

3 : Étape Maximisation M : Estimation des paramètres $\theta = \arg\max_{\theta'} L_C(\mathbf{x}; \theta')$.

jusqu'à *Un état fixé au départ est atteint*;

Les deux approches EM et CEM sont bien placées dans la littérature de modélisation non supervisée, des problèmes liés aux modèles de mélange (estimation, classification).

Dans certaines situations notamment lorsque les classes sont disproportionnées, l'algorithme CEM peut s'avérer meilleur que EM. Cependant, étant donné que CEM travaille par classe, l'estimation des paramètres est biaisée. Signalons toutefois que l'algorithme CEM est beaucoup plus rapide que l'algorithme EM qui notons le peut être vu comme un algorithme de classification flou

Proposition (Hathaway, 1986)
La log-vraisemblance $L(\mathbf{x};\theta)$, peut se décomposer comme suit :

$$L(\mathbf{x};\theta) = \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log \pi_k f(\mathbf{x}_i;\alpha_k) - \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log t_{ik}^{(c)}$$

Ceci est équivalant à écrire :

$$L(\mathbf{x};\theta) = Q\left(\theta,\theta'\right) - H\left(\theta,\theta'\right)$$

$H(\theta,\theta')$ représente une mesure d'entropie du mélange (Bezdek, 1981), il est connu que cette valeur est décroissante lorsque les composants du mélange sont bien séparés, on a :
$\forall \mathbf{x}_i; \exists k = 1,...,g;\ t_{ik} \simeq 1$ et $\sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log t_{ik}^{(c)} = 0$
et

$$L(\mathbf{x};\theta) = Q\left(\theta,\theta'\right) = \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log \pi_k f(\mathbf{x}_i;\alpha_k)$$

cette expression correspond à la log-vraisemblance classifiante, où chaque classe P_k est définie par $P_k = \{\mathbf{x}_i;\ t_{ik} \simeq 1\}$.

Pour exploiter les avantages de CEM et EM, une manière simple consiste à lancer CEM plusieurs fois (convergence rapide), la solution obtenue sera le point initial pour lancer EM.

4.2 Algorithme SEM

Proposé par (Celeux et Diebolt, 1986), c'est un algorithme d'apprentissage probabiliste pour la reconnaissance de mélange de densité. Une étape stochastique S est introduite entre les deux étape E et M dans le but d'estimer par une approche d'apprentissage probabiliste, g le nombre de composants et les g composants d'un mélange fini de lois de probabilités (reconnaissance du mélange). L'étape intermédiaire S est une simulation de Monté Carlo, elle consiste à tirer à chaque itération une partition selon la probabilité a posteriori $t_{ik}^{(c)} = p(\mathbf{x}_i \in P_k|\mathbf{x};\theta')$ calculée à l'étape E, l'estimation des paramètres est basée sur ce tirage aléatoire, à l'étape maximisation M, le nombre de classes est ajusté aux données.

Algorithme 6: L'algorithme SEM

Entrée : $\mathbf{x}$: la matrice des données, le paramètre g majorant du nombre de composants du mélange et d'un seuil $c(n)$ compris entre 0 et 1.

Sortie : Le nombre de classes final. Les classes : $P_1, ..., P_g$.

Initialisation : Choix aléatoire des paramètres du mélange $\theta_k, k = 1, \ldots, g$;

Répéter

1 : Étape Estimation E : estimation des t_{ik} $k = 1, ..., g$:

$$t_{ik} = \frac{\pi_k f(\mathbf{x}_i; \alpha_k)}{\sum_{k=1}^{g} \pi_k f(\mathbf{x}_i; \alpha_k)}$$

2 : Étape Stochastique S : On tire en chaque point $\mathbf{x}_i$ la variable aléatoire multinomiale $e^n(\mathbf{x}_i)$ d'ordre 1 et de paramètre $t_{ik}, k = 1, ..., g$, telle que :

$$e^n(\mathbf{x}_i) = (e_k^n(\mathbf{x}_i); k = 1, \ldots, g)$$

Les réalisations $e^n(\mathbf{x}_i)$ définissent une partition : $P^n = (P_1^n, ..., P_k^n)$ de l'échantillon, avec :

$$P_k^n = \{\mathbf{x}_i | e_k^n(\mathbf{x}_i) = 1\}$$

si pour un certain k, *card* $P_k < nc(n)$, l'algorithme doit être réinitialisé ;

3 : Étape Maximisation M : maximisation de la vraisemblance dans le but d'estimer les paramètres du mélange $\theta_k = (\pi_k, \alpha_k)$ sur la base des probabilités e_{ik}, la distribution choisie et des sous échantillons P_k, $k = 1, ..., g$. Les π_k sont estimés par :

$$\pi_k = \frac{1}{n} \sum_{i=1}^{n} e_{ik}$$

jusqu'à *Un état fixé au départ est atteint*;

A la convergence, l'algorithme fournit une classe de partitions statistiquement admissibles pour les estimations des paramètres du mélange ; l'algorithme converge en loi, ce qui justifie la stationnarité de la suite des paramètres estimés $\theta^{(c)}$, les tirages aléatoires à chaque itération empêchent une convergence de la vraisemblance vers un maximum local instable, comme dans le cas de l'algorithme EM ; comparé à ce dernier, l'algorithme SEM converge plus rapidement et indépendamment de l'initialisation, ce qui favorise l'utilisation de ce dernier par rapport aux algorithmes classiques et à l'algorithme EM lui même. Il représente aussi un outil puissant d'estimation du nombre de composants dans un mélange, à savoir une borne supérieure.

4.3 Algorithme SAEM

L'algorithme SAEM : Stochastic Approximation Expectation Maximization ou bien Simulated Annealing (Celeux et Diebolt, 1992) utilise une approximation stochastique pour calculer l'espérance conditionnelle, cet algorithme est une variante de l'algorithme SEM ; il commence par une initialisation des paramètres des composants du mélange et la simulation des probabilités a posteriori $t_{ik}^{(0)}$, l'algorithme 7 est illustré ci-dessous :

γ^c représente une suite de nombres positifs convergeant vers 0. En pratique, il est préférable qu'elle soit plus proche de 1 pendant les premières itérations ; pour éviter les valeurs sous optimales ou stationnaires de la log-vraisemblance (Celeux et Diebolt, 1992).

$$\theta_{SAEM}^{(c)} = \theta_{SEM}^{(c)} + (1 - \gamma^{(c)})\theta_{EM}^{(c)}$$

$\theta_{SEM}^{(c)}$ et $\theta_{EM}^{(c)}$ sont les paramètres estimés à une étape (c) par SEM et EM respectivement.

La suite $\gamma^{(c)}$ introduit une excitation stochastique dans l'algorithme, ce qui permet d'éviter la convergence vers des points stationnaires non intéressants. Cet algorithme est de type recuit-simulé, il dépend fortement de la suite $\gamma_{(c)}$ qui décroit au cours des itérations : au début des itérations cet algorithme ressemble le plus à SEM, à la convergence il est plus proche de EM. Il convient beaucoup plus aux données de petite taille .

5 Propriétés de convergence de l'algorithme EM

Il est connu que l'algorithme EM génère une suite L^m jusqu'à un point stationnaire de la probabilité des données incomplètes, défini par un maximum local, global ou même par un point selle dépendant du choix de la situation initiale. La convergence de l'algorithme EM vers un maximum local de la vraisemblance des données incomplètes a été démontrée par Wu (Wu, 1983) sous des hypothèses restrictives, qui ont été définies par Delyon et al. (Delyon *et al.*, 1999). En outre, dans de nombreux cas pratiques, à la convergence, on obtient un maximum local.

6 Amélioration et accélération de EM

Malgré que l'algorithme EM assure une convergence vers un maximum local de la vraisemblance, suivant certaines conditions (Wu, 1983), cependant, il possède d'autres limitations ; parmi lesquelles, le résultat final dépend fortement de l'initialisation, en plus de sa lenteur, il peut même se trouver coincé dans un point selle de la vraisemblance.

Des solutions à ces problèmes ont été proposées dans le passé. Certaines stratégies d'initialisation pour éviter la dépendance aux points de départ, parmi lesquelles : lancer

Algorithme 7: L'algorithme SAEM

Entrée : $\mathbf{x}$: la matrice des données, le paramètre g majorant du nombre de composants du mélange et d'un seuil $c(n)$ compris entre 0 et 1.

Sortie : Le nombre de classes final. La partition : $P_1, ..., P_g$.

Initialisation : Choix aléatoire des paramètres du mélange $\theta_k, k = 1, \ldots, g$;

Répéter

1 : Étape Estimation E : estimation des t_{ik} $k = 1, ..., g$:

$$t_{ik} = \frac{\pi_k f\left(\mathbf{x}_i; \alpha_k\right)}{\sum_{k=1}^{g} \pi_k f\left(\mathbf{x}_i; \alpha_k\right)}$$

2 : Étape Stochastique S : On tire en chaque point $\mathbf{x}_i$ la variable aléatoire multinomiale $e^n(\mathbf{x}_i)$ d'ordre 1 et de paramètre $t_{ik}, k = 1, ..., g$, telle que :

$$e^n(\mathbf{x}_i) = (e_k^n(\mathbf{x}_i); k = 1, \ldots, g)$$

Les réalisations $e^n(\mathbf{x}_i)$ définissent une partition : $P^n = (P_1^n, ..., P_k^n)$ de l'échantillon, avec :

$$P_k^n = \{\mathbf{x}_i | e_k^n(\mathbf{x}_i) = 1\}$$

3 : Étape Hybrid (Annealing) A : $i = 1, \ldots, n$, $k = 1, \ldots, g$

$$r_{ik}^n = t_{ik} + \gamma(e_k^n - t_{ik})$$

si $card\ P_k < nc(n)$, l'algorithme doit être réinitialisé ;

4 : Étape Maximisation M : maximisation de vraisemblance dans le but d'estimer les paramètres du mélange $\theta_k = (\pi_k, \alpha_k)$ sur la base de la 2éme étape et de la distribution choisie.

Les π_k sont estimés par :

$$\pi_k = \frac{1}{n} \sum_{i=1}^{n} r_{ik}^n$$

jusqu'à *Un état fixé au départ est atteint*;

plusieurs fois EM pour un nombre précis d'itérations depuis des initialisations aléatoires, le résultat associé à la plus grande vraisemblance initialisera un autre EM qui sera itéré jusqu'à la convergence. D'autres initialisations sont possibles avec les résultats à la convergence d'un algorithme géométrique ($k$means) ou bien des algorithmes de type EM, soit les plus rapides que EM ; ceci est très pratique et aussi répandu. Les variantes de EM, elles

aussi peuvent être utilisées comme version accélérée de l'algorithme (CEM, SEM...). Pour éviter que EM soit bloqué dans un point selle de la vraisemblance, SEM peut être utilisé au lieu de EM, mais à condition qu'il soit arrêté après un nombre fixé d'itérations, car il ne converge pas ponctuellement mais converge en loi. Il est également possible d'utiliser l'algorithme SAEM qui à chaque itération, diminue l'influence des perturbations aléatoires causées par l'étape stochastique et qui permet de s'arrêter à la stationnarité de la vraisemblance.

La lenteur de l'algorithme EM pourrait poser un problème sérieux lors de l'étude des données de taille importante. Plusieurs études ont été destinées à l'accélération de EM et plusieurs solutions ont été proposées. Ces méthodes peuvent être réparties en deux catégories (Berlinet et Roland, 2009), l'une opère sur les étapes de EM (E ou bien M) ; ou comporte des méthodes qui insèrent des étapes supplémentaires ou bien qui prolongent l'ensemble des paramètres ; comme par exemple : SpEM (Sparse EM) (Neal et Hinton, 1998), LEM (Lazy EM) (Thiesson *et al.*, 2001), PX-EM (EM Parameter expansion EM) (Liu *et al.*, 1998) et eLEM (Jollois et Nadif, 2007). Dans la deuxième catégorie, on trouve des méthodes générales conçues à la résolution des problèmes de point fixe ou bien plus adaptées à des cas particuliers de l'algorithme EM (Berlinet et Roland, 2009).

7 Conséquence

Dans ce chapitre, nous avons illustré quelques outils de base utilisés dans notre travail. L'approche ML et l'approche CML sont deux requêtes clés dans la démarche classique de la classification par les modèles de mélange. On a commencé par définir ces deux approches, ensuite introduire l'algorithme EM l'une des plus importantes procédures de maximisation de vraisemblance et la plus utilisée, son importance vient du fait que la partition cherchée est estimée à la fin de l'algorithme. Puis on a présenté les trois algorithmes : CEM (CML) SEM et SAEM. On a expliqué l'importance de chaque variante vis à vis de l'algorithme EM lui même. Enfin on a rappelé les propriétés de convergence de l'algorithme EM et on a discuté des moyens les plus populaires pour l'amélioration et l'accélération de ce dernier. Après ce chapitre, une étape nécessaire et très importante avant d'aborder notre problème : c'est de définir l'espace du problème dans lequel toutes les méthodes qu'on a développées sont appliquées. Le chapitre suivant est consacré aux données directionnelles, leurs statistiques et quelques distributions favorables à leurs structures spécifiques.

Chapitre 3

Données directionnelles

Les données qui seront traitées dans la suite de la thèse sont de grande dimension et sont aussi intrinsèquement de nature directionnelle. Dans un objectif de classification les algorithmes utilisant des mesures standards ne sont pas appropriés Mardia et Jupp (2009). Les données directionnelles sont souvent normalisées de telle sorte qu'elles appartiennent à la surface d'une hypersphère de rayon 1. Ainsi, dans le plan ou dans un espace à trois dimensions, l'espace est un cercle ou une sphère. Dans ce domaine, il existe des travaux montrant que la normalisation des vecteurs de données contribue à éliminer les biais induits par la longueur d'un vecteur et de fournir des résultats meilleurs (Salton et Buckley, 1988; Salton et McGil, 1983).

Dans les trente dernières années, les données directionnelles ont fait l'objet de plusieurs ouvrages ; dont nous citons quelques uns : Watson (1983), dans son livre, l'auteur parle principalement de la théorie de l'inférence pour les distributions sur les sphères de dimensions arbitraires. Fisher (1995) et (Fisher *et al.*, 1993), se sont concentrés plutôt sur les méthodes d'analyse de données sur le cercle unitaire (2-dimensions) et la sphère unitaire (3-dimensions) ; Batschelet (1981) regroupe une large variété d'applications en biologie. Dans le livre Upton et Fingleton (1989), les chapitres (9-10) sont consacrés à la théorie et aux applications des statistiques circulaires et sphériques. Parmi les ouvrages les plus récents, celui de Mardia et Jupp (2009) englobe la théorie des statistiques directionnelles sur le cercle unitaire, puis sur une sphère de dimension arbitraire, le troisième chapitre de Declan (1996) est consacré à l'analyse des données directionnelles ; ce livre est destiné aux géologues, il résume les changements rapides dans les applications résultantes de la révolution des ordinateurs personnels. Jammalamadaka et SenGupta (2001) présentent une recherche monographique sur l'analyse des données circulaires, couvrant certains sujets avancés récents dans le domaine. Ils présentent aussi dans leur travail certaines inconvénients des méthodes et modèles existants. Le livre Chikuse (2003) s'intéresse à l'ana-

lyse statistique sur deux collecteurs spéciaux, la variété de Stiefel, représentée par un ensemble de matrices dont les colonnes sont des variables aléatoires orthogonales de longueur unitaire ; et le collecteur de Grassmann, représenté par un ensemble de matrices, constituées par des projections orthogonales idempotentes. Le livre de de Sá (2003) destiné aux applications d'analyse statistique pour une grande variété de problèmes pratiques en utilisant SPSS, Matlab et Statistica. On y trouve les principaux sujets d'analyse statistique parmi lesquels les statistiques directionnelles. Stergiou (2004) est un ouvrage dédié aux plus récentes et plus appropriées méthodes mathématiques et statistiques conçues à l'analyse des petits et grands ensembles de données biomécaniques (chapitre 5). Hudson et Keatley (2010) représente un ouvrage de référence d'études dans divers domaines : écologie, santé humaine, pêche, foresterie, agriculture et gestion des ressources naturelles ; au chapitre 16, les auteurs entament des applications en statistiques circulaires dans la phénologie des plantes. Lebart et Salem (1994) : ce livre présente les concepts de base et les fondements des méthodes de la statistique textuelle.
Dans ce chapitre, on propose de donner quelques définitions concernant les statistiques directionnelles, afin de les utiliser dans les prochains chapitres consacrés à la classification automatique et les modèles de mélange utilisant les données directionnelles. On définit tout d'abord les données circulaires, en introduisant quelques exemples de données de ce type, pour ensuite entamer l'étude de quelques statistiques des tableaux de données directionnelles ; on définit brièvement quelques unes des distributions adaptées à ce type de données. Enfin, on introduit les mêmes notions sur les données sphériques, les données circulaires et les données directionnelles multidimensionnelles qui seront traitées par les même méthodes et de la même manière (Mardia et Jupp, 2009).

1 Données circulaires

Les données circulaires peuvent être vues comme des points sur un cercle unitaire, ou même comme vecteurs unitaires sur un plan, dans ce cas les points précisent la direction des données. Chaque donnée est associée à l'angle, ceci par rapport à une direction initiale choisie sur le même cercle ; les données axiales sont un type bien particulier des données circulaires, les données distribuées sur le cercle unitaire sur le même axe sont équivalentes, telle que $\theta \leftarrow \theta + \pi$.

La représentation graphique la plus simple des données circulaires, consiste à les représenter par des points sur un cercle de rayon un.

Les plus anciennes groupes de données circulaires observées et étudiées, ceux de l'immigration des animaux, les plus célèbres sont les données de tortues qui ont été amenés sur l'orientation des animaux après avoir pondu leurs œufs (Shi et Tsai, 1963).

De nos jours, le besoin du traitement des données directionnelles apparait comme nécessaire dans plusieurs domaines , parmi lesquels on cite :

- Sciences de la terre, la surface sphérique du globe terrestre a favorisé la projection des données sur la sphère, par exemple, les coordonnées des épicentres de tremblement de terre ; l'étude du paléomagnétisme des rochets et dans beaucoup d'autres domaines des sciences de la terre (Watson, 1970).
- Météorologie, plusieurs données sont représentées en fonction de leurs directions, comme par exemple la direction du vent en plus de sa vitesse, ainsi que les dates et les périodes des événements météorologiques (pluies, neiges ...).
- Biologie, ce domaine est très riche d'exemples intéressants, comme l'étude des données colverts de l'union des ornithologues britanniques, les directions de natation de l'Daphnia (Waterman, 1963).
- Physique, von Mises (von Mises, 1918) avait proposé d'utiliser sa fameuse distribution circulaire pour l'étude des poids atomiques, ceci bien avant les isotopes, les directions des axes optiques des cristaux, les vagues de sons (Rayleigh, 1919) et les liaisons moléculaires (Rayleigh, 1919; Kuhn et Grün, 1942).
- Psychologie, plusieurs données sont représentées sous forme de données directionnelles, par exemple les cartes mentales sur lesquelles les gens représentent leurs environnements par exemple, sont des données circulaires (Gordon *et al.*, 1989).
- L'analyse de l'image, ici aussi l'analyse des données directionnelles est bien connue (Mardia *et al.*, 1996; Blake et Marinos, 1990; Mardia *et al.*, 1997).
- Médecine, des tableaux de données sphériques peuvent se produire en cardiologie, les informations fournies sur l'activité électrique du cœur pendant un battement décrite en termes d'orbite quasi-planaire dans l'espace tridimensionnel (Downs et Liebman, 1969; Gould, 1969).
- Astronomie, plusieurs hypothèses ont été envisagées au sujet des distributions de divers objets astronomiques traités comme données directionnelles (Pólya, 1919; Jupp, 1995).
- Génétique, les données d'expression génétique ont toujours été considérées de caractéristiques directionnelles uniques qui suggèrent l'utilisation d'un modèle directionnel. Dans (Dhillon et Sra, 2003), les auteurs ont traité le problème de classification des tableaux de données génétiques en utilisant des modèles directionnelles.
- Étude de données textuelles, l'utilisation de métriques appropriées ou la normalisation des données s'avèrent des étapes nécessaires que l'utilisateur ne peut éviter, les données textuelles sont vues naturellement comme des données directionnelles (Dhillon *et al.*, 2002).

Les statistiques textuelles ont été conçues et exploitées en plusieurs disciplines : la statistique classique, l'analyse du discours, l'informatique, le traitement des enquêtes et le traitement de texte (Lebart et Salem, 1994).

2 Statistiques des tableaux de données directionnelles

En normalisant les données de telle sorte qu'elles appartiennent à une hypersphère unitaire, plusieurs statistiques de type directionnel sont utilisées. Par exemple, sur un cercle chaque point $\mathbf{x}$ est représenté par l'angle α entre ce point et un axe référence, dans le cas général l'angle est remplacé par un nombre complexe $z = \exp i\alpha$ (Mardia et Jupp, 2009). Dans ce cas la description des données se fait en utilisant les paramètres statistiques de position et de dispersion :

1. Les paramètres de position sont des mesures communes d'emplacement, ou de tendance centrale, ils caractérisent le positionnement des éléments de la série statistique, parmi lesquels on a :
   - La moyenne directionnelle : c'est la moyenne des éléments statistiques, divisée par sa norme, c'est la direction de la masse des données.
   - La médiane directionnelle : c'est l'élément qui divise l'ensemble en deux groupes de même effectif, séparant la moitié supérieure de la moitié inférieure de l'ensemble de données.
   - Le mode : c'est la valeur la plus fréquente, ainsi certains groupes de données sont uni-modale, d'autres multimodales.
2. Les paramètres de dispersion : sont des caractéristiques représentant l'étendue de l'ensemble des données ; les plus courantes sont :
   - L'étendue circulaire : c'est le plus petit arc du cercle qui contient toutes les observations.
   - La variance sphérique : exprime la dispersion des données autour de la moyenne directionnelle, Mardia (Mardia et Jupp, 2009) a donné son expression dans le cas bidimensionnel et il a introduit la longueur du vecteur centre de masse $\bar{R}$. Soit un ensemble I de n point sur la sphère unitaire, avec $\theta_i,\ i = 1, \ldots, n$ les angles correspondant. Les coordonnées cartésiennes du vecteur centre de masse de I sont $\left(\frac{1}{n}\sum_{i=1}^{n}\cos\alpha_i,\ \frac{1}{n}\sum_{i=1}^{n}\sin\alpha_i\right)$. La moyenne directionnelle $\bar{\theta}$ est solution du système :

$$\begin{cases} \bar{R}\cos\bar{\theta} = \frac{1}{n}\sum_{i=1}^{n}\cos\alpha_i \\ \bar{R}\sin\bar{\theta} = \frac{1}{n}\sum_{i=1}^{n}\sin\alpha_i \end{cases}$$

Et la longueur du vecteur centre de masse est :

$$\bar{R} = \frac{1}{n}\sqrt{\left[\left(\sum_{i=1}^{n}\cos\alpha_i\right)^2 + \left(\sum_{i=1}^{n}\sin\alpha_i\right)^2\right]}$$

qui vérifie $0 \leq \bar{R} \leq 1$. Cette valeur exprime une faible concentration autour de la direction moyenne si elle est proche de un et une concentration ponctuelle si elle est proche de 0. Dans ce cas, la valeur de la variance directionnelle est : $V = 1 - \bar{R}$, cette expression n'est pas unique, dans (Batschelet, 1981), on trouve $V = 2(1 - \bar{R})$, qui vérifie toujours $0 \leq V \leq 1$ mais de caractéristique inverse à $\bar{R}$.

– L'écart-type sphérique : dans (Mardia et Jupp, 2009), les auteurs définissent l'écart type circulaire par $v = \sqrt{-2\log(1-V)}$ et définissent aussi, d'autres moments et paramètres à partir de $\bar{R}$.

3 Distributions circulaires

Les distributions directionnelles ont été présentées dans plusieurs ouvrages (Mardia et Jupp, 2009), elles ont toutes des formes exponentielles, la plus importante est la distribution de von Mises, nous nous limitons ici, à des définitions bibliographiques dans le cas circulaire, afin de les utiliser dans les chapitres suivants.

3.1 Distribution lattice

La distribution lattice est une distribution discrète, qui prend des valeurs de la forme $a + bl$ où $a, b \neq 0$ et l un entier (Abramowitz et Stegun, 1972). Cette notion a été reprise dans (Mardia et Jupp, 2009), avec $b = \frac{2\pi}{n}$ et $l = 0, 1, \ldots, n-1$. Les n points distribués ainsi sur le cercle sont les sommets d'un n-polygone régulier sur le cercle, de probabilité égale à $\frac{1}{n}$, en cas d'équiprobabilité.

3.2 Distribution uniforme simple

Comme dans le cas de n'importe quelle loi uniforme, pour les données directionnelles disposées sur un cercle, la loi uniforme est de densité :

$$f(x) = \frac{1}{2\pi}$$

et Donc pour un arc $\alpha\beta$:

$$p(\alpha < \theta \leq \beta) = \frac{\beta - \alpha}{2\pi}$$

3.3 Distribution de von Mises

En statistique inférentielle, cette loi peut être la plus utile sur le cercle (Mardia et Jupp, 2009), elle dépend de la moyenne directionnelle μ et d'un paramètre de concentration ξ, sa fonction de distribution est de la forme :

$$f(x;\mu,\xi) = \frac{\exp\left[\xi \mathbf{x}^t \mu\right]}{2\pi I_0(\xi)}$$

où $I_0(\xi) =$ représente la fonction de Bessel modifiée du 1^{er} type d'ordre 0 définie par : $I_0(\xi) = \frac{1}{2\pi}\int_0^{2\pi} e^{\xi\cos\theta} d\theta$.
Cette distribution fait l'objet principal de notre travail de thèse ; elle sera étudiée en détail au chapitre 5, en donnant plus d'importance au cas des données ayant une dimension supérieure à deux.

3.4 Distribution normale projetée

Cette distribution comme l'indique son nom est une résultante de la normalisation d'une loi normale (Mardia, 1972), pour $\mathbf{x} = (cos\theta, sin\theta)^t$, sa fonction de distribution est de la forme :

$$p(\theta;\mu,\Sigma) = \frac{\psi(\theta;0,\Sigma) + |\Sigma|^{\frac{-1}{2}}\, D(\theta)\Psi(D(\theta))\psi(|\Sigma|^{\frac{-1}{2}}\,(\mathbf{x}^t\Sigma^{-1}\mathbf{x})^{\frac{-1}{2}}\mu \wedge \mathbf{x})}{\mathbf{x}^t\Sigma^{-1}\mathbf{x}}$$

où $\psi(\theta(\mathbf{x});0,\Sigma)$ représente la densité de la loi normale de paramètres $(0,\Sigma)$, Ψ est la fonction de densité cumulative de la fonction de densité de la loi normale centrée réduite ; $\mu \wedge \mathbf{x} = \mu' \sin\theta(\mathbf{x}) - \mu'' \cos\theta(\mathbf{x})$, avec : $\mu = (\mu', \mu'')^t$ et $D(\theta(\mathbf{x})) = (\mu^t\Sigma^{-1}\mathbf{x})(\mathbf{x}^t\Sigma^{-1}\mathbf{x})^{\frac{-1}{2}}$

3.5 Distribution enveloppée

On peut obtenir une distribution enveloppée, en enveloppant une distribution connue sur le cercle unitaire, cette notion simple enrichit l'ensemble des distributions circulaires. Ceci est possible pour n'importe quelle distribution connue, une variable aléatoire $\mathbf{x}$ admet une variable aléatoire enveloppée de la manière suivante : $\tilde{\mathbf{x}} = \mathbf{x}(mod(2\pi))$, et si notre distribution est définie sur $\mathbb{R}$, la densité de la distribution résultante enveloppée sur le cercle, prend la forme :

$$\mathbf{f}(\theta(\mathbf{x})) = \sum_{-\infty}^{\infty} \mathbf{f}(\theta(\mathbf{x}) + 2k\pi),$$

Une fonction de ce type est un homomorphisme de $\mathbb{R}$ dans le cercle S^1, elle vérifie : $(\mathbf{x} \tilde{+} \mathbf{y}) = \tilde{\mathbf{x}} + \tilde{\mathbf{y}}$. La fonction caractéristique de $\tilde{\mathbf{x}}$, une variable enveloppée de $\mathbf{x}$, dont la fonction caractéristique ϕ, est :

$$\phi = \phi(p); p \in \mathbf{Z}$$

Plusieurs distributions dans $\mathbb{R}$ peuvent être enveloppées autour du cercle unitaire, parmi lesquelles :

- Distribution de Poisson enveloppée : pour une variable aléatoire de Poisson $\mathbf{x}$ de paramètre λ, la variable aléatoire de Poisson enveloppée résultante est de fonction de probabilité (Ball et Blackwell, 1992) :

$$p\left(\theta(\mathbf{x})\right) = \frac{2\pi l}{n} = \frac{1}{n}\sum_{i=0}^{n-1} \exp\left(w^i\lambda\right) w^{-li}$$

  où w est la $n^{\text{ième}}$ racine complexe de 1, $l = 0, \ldots, n-1$
- Distribution normale enveloppée : soit une variable aléatoire suivant une loi normale, $\aleph(\mu, \sigma^2)$ sur $\mathbb{R}$, la loi normale enveloppée de celle-ci est de distribution (Abramowitz et Stegun, 1972) :

$$\tilde{\phi}\left(\theta(\mathbf{x}); \tilde{\mu}, \beta\right) = \frac{1}{2\pi}\left(1 + 2\sum_{i=1}^{\infty} \beta^{i^2} \cos i(\theta(\mathbf{x}) - \tilde{\mu})\right)$$

  où $\beta = \exp\left(\frac{\sigma^2}{2}\right)$, $0 \leq \beta \leq 1$ et $\tilde{\mu} = \mu(mod(2\pi))$.
  Cette distribution apparaît dans le mouvement Brownien, dans ce cas on dit mouvement Brownien enveloppé sur le cercle (Shi et Tsai, 1963).
- Distribution de Cauchy enveloppée : si on a une distribution de Cauchy $C(\mu, \alpha)$ sur $\mathbb{R}$, la distribution de Cauchy enveloppée correspondante est définie par (Wintner, 1947) :

$$\tilde{C}\left(\theta(\mathbf{x}); \tilde{\mu}, \beta\right) = \frac{1}{2\pi}\left(1 + 2\sum_{i=1}^{\infty} \beta^{i} \cos i(\theta(\mathbf{x}) - \tilde{\mu})\right)$$

  où $\tilde{\mu} = \mu(mod(2\pi))$ et $\beta = \exp\left(-\alpha\right)$. cette distribution est équivalente à la distribution normale projetée (Mardia, 1972), de telle sorte que si $\theta(\mathbf{x})$ suit la loi normale projetée centrée, alors $2\theta(\mathbf{x})$ suit une loi de Cauchy enveloppée, tel que :

$$\beta^2 = \frac{\Sigma^t - 2\left|\Sigma\right|^{\frac{1}{2}}}{\Sigma^t + 2\left|\Sigma\right|^{\frac{1}{2}}},\ \tan\mu = \frac{2\sigma_{12}}{\sigma_{11} - \sigma_{22}} \text{ et } \Sigma = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{12} & \sigma_{22}^2 \end{pmatrix}.$$

4 Données sphériques

L'aspect tridimensionnel de la vie réelle, nous permet de percevoir les données sphériques autour de nous, les directions des paléomagnétismes dans les roches et aussi les directions des axes optiques dans les cristaux de quartz.

Sur le plan théorique, il existe trois approches de base, propres aux statistiques directionnelles : le plongement, l'enveloppement et l'approche intrinsèque :

- L'approche du plongement : dans cette approche, la sphère S^3 est considérée comme un sous-ensemble de $\mathbb{R}^3$, cette approche est la base de construction de la distribution normale projetée.
- L'approche enveloppement : dans cette approche d'emballage, un vecteur $\mathbf{x}$ tangent à la sphère en μ, est enroulé sur la sphère par la formule $\mu \sin \|\mathbf{x}\| + \xi \cos \|\mathbf{x}\|$, où $\mathbf{x}^t\mu = 0$, ceci a été la base constructive des distributions enveloppées et des mouvements Browniens.
- L'approche intrinsèque : pour cette approche la sphère est considérée comme un collecteur par elle même, sans référence à un plongement, en utilisant la diffusion sur le cercle, avec cette approche, on peut obtenir la distribution de von Mises.

Soit un échantillon statistique sphérique, la moyenne normalisée est dite moyenne directionnelle :

$$\mu = \frac{\sum_{i=1}^{n} \mathbf{x}_i}{\|\sum_{i=1}^{n} \mathbf{x}_i\|}$$

quelques fois la quantité $2(1 - \|\sum_{i=1}^{n} \mathbf{x}_i\|)$ représente la variance sphérique de l'échantillon (Mardia et Jupp, 2009), cette notion exprime la dispersion des éléments autour de la moyenne, plus elle se rapproche de 0 plus la dispersion est forte, quand elle est proche de 1 la dispersion est faible et l'échantillon tend vers sa moyenne directionnelle. Comme exemples de distributions sphériques, on cite ici quelques unes qui sont les plus utilisées, soit :

- La distribution uniforme : est une distribution basique sur la sphère unitaire, comme dans $\mathbb{R}^d$, cette distribution est réflexive et invariante par rapport à la rotation, dans ce cas, elle est d'espérance nulle.
- La distribution de von Mises-Fisher : la plus utilisée, elle sera étudiée en détail au chapitre 5.
- La distribution du mouvement Brownien : est la distribution de paramètres μ, k, d'un point aléatoire circulant sur la sphère unitaire S^{d-1}, à partir du point μ, à un moment k^{-1}, en diffusion isotopique. Dans le cas circulaire , elle n'est que la distribution normale enveloppée de moyenne $(\cos \mu, \sin \mu)^t$.
- La distribution de Fisher-Bingham : créée à partir d'une généralisation de la distribution de Fisher (Beran, 1979), dans (Mardia, 1975) sa densité est donnée par :

$$f(\mathbf{x}; \mu, k, A) = \frac{1}{a(k, A) \exp(k\mu^t\mathbf{x} + \mathbf{x}^t A\mathbf{x})},$$

  A est une matrice $d \times d$ symétrique, $k \geq 0$ et $a(k, A)$ est une constante de normalisation.
- La distribution de Kent : introduite par Kent (Kent, 1982) pour modéliser quelques groupes de données dont les anciens modèles ne couvrent pas. Cette distribution

peut être obtenue à partir de la distribution de Fisher-Bingham, ayant la même forme que cette dernière, en imposant la restriction $A\mu = 0$.

- La distribution de Fisher-Watson : un modèle de Watson introduit par Wood (Wood, 1988) et obtenu en imposant à la matrice A d'avoir un rang égal à un dans le modèle de Fisher.
- La distribution de Bingham-Mardia : (Bingham et Mardia, 1978), elle traite des modèles plus compliqués, par exemple en sciences de la terre, où il s'agit de distributions sphériques de rotation-symétriques. Elle est de la forme :

$$f(\mathbf{x}; \mu, k, \upsilon) = \frac{1}{a(k)} \exp\left(k\left(\mu^t \mathbf{x} - \upsilon\right)\right),$$

  $a(k)$ est une constante de normalisation.
- La distribution de Wood : une modification de la distribution de Fisher qui sert à modéliser les données bimodales sur la sphère S^{d-1} (Wood, 1982), ce type de distribution admet deux modes de la même puissance.
- La distribution projetée : une projection radiale d'une distribution sur $\mathbb{R}^d$ peut fournir une distribution sur S^{d-1}, une telle projection vérifie $p(\mathbf{x} = 0) = 0$ et les vecteurs obtenus sont de la forme $\frac{\mathbf{x}}{\|\mathbf{x}\|}$, cette notion est connue et plusieurs distributions en résultent : (Watson, 1983; Pukkila et Rao, 1988; Kent et Mardia, 1997; Boulerice et Ducharme, 1994).

Dans Mardia et Jupp (2009), les auteurs font la distinction entre les distributions directionnelles et les distribution axiales, qui ont été conçues aux cas de données fournies sous forme d'axes. Rappelons que dans ce cas, l'utilisateur peut confondre entre $\mathbf{x}$ et son opposé $-\mathbf{x}$, ici il été plus convenable de considérer les fonctions de distributions sur S^{d-1} qui sont plutôt symétriques. Dans cette étude, on se limite aux distributions sphériques.

5 Discussion

Dans ce chapitre, on a illustré quelques outils de base utilisés en statistiques des données circulaires et sphériques, en rappelant quelques anciennes notions de distributions sur le cercle, puis sur la sphère unitaire. Ceci nous a permis de prendre en considération, l'importance des données directionnelles.

Dans le prochain chapitre, et pour aborder le problème de classification automatique des données directionnelles, on commence d'abord par présenter et comparer des méthodes géométriques de type $k$means. Une illustration rapide nous permettra ensuite de sélectionner les algorithmes qui seront utilisés dans nos applications sur les données de type directionnel.

Chapitre 4

Méthodes de type $k$means pour le traitement de données directionnelles

Parmi les algorithmes géométriques destinés à la classification automatique, il y a bien sûr l'algorithme $k$means (Forgy, 1965) dont plusieurs variantes ont été proposées. Le nom $k$means est du à McQueen (McQueen, 1967), cet algorithme est une version de la méthode des nuées dynamiques où les noyaux étant des centres de gravité. Pour les données directionnelles, plusieurs variantes ont été proposées, nous en retiendrons deux. La première est due à Lelu (Lelu, 1993) qui propose la méthode des $k$means axiales en passant par une normalisation des données et en utilisant le produit scalaire comme mesure de similarité, cette démarche est très efficace et rapide dans le cas de données textuelles où la matrice des données est creuse en général. Cette méthode est de type séquentiel car les centres sont mis à jour à chaque affectation. La seconde est une autre extension du $k$means appelée *spherical* $k$means (Dhillon et Modha, 2001), dans la suite on la note SPK. Elle est définie sur l'hypersphère unitaire avec le cosinus en tant que mesure de similarité. Le principal avantage de ce dernier est qu'il est indépendant de la taille des données, ce qui permet de comparer des documents de tailles différentes et d'importance comparable comme par exemple un texte et son résumé. Une version séquentielle a été proposée par (Zhong, 2005).

Le but de ce chapitre est de présenter ces différentes variantes particulièrement intéressantes dans le cadre des données directionnelles. Nous montrons les différences entre le $k$means axiale et le $k$means sphérique. Nous évaluerons leur performance à partir de données simulées.

1 La méthode $k$means

La classification est effectuée par le regroupement des données autour d'un ensemble de plusieurs éléments, appelés noyaux ou centres (Forgy, 1965). Chaque noyau noté μ_k est un représentant de la $k^{\text{ième}}$ classe et constituera un résumé des données. Des contraintes peuvent être imposées aux noyaux afin de faciliter l'interprétation des classes. Pour ces raisons, la plupart des méthodes de classification automatique reposent sur le principe de $k$means qui consiste à minimiser un critère d'inertie intra-classe :

$$W(P,\mu) = \sum_{k=1}^{g}\sum_{i=1}^{n} z_{ik} d^2(\mathbf{x}_i, \mu_k), \tag{4.1}$$

avec $\mu = (\mu_k; k = 1, ..., g)$.

En entrée, on dispose de $\mathbf{x}$ un ensemble de points (objets, individus) dans $\mathbb{R}^d$, un entier positif g supposé connu qui correspond au nombre de classes, et en sortie, on dispose d'une partition P de $\mathbf{x}$ en g classes disjointes, et de l'ensemble des noyaux associés μ_k. Les classes sont obtenues, en réitérant jusqu'à la convergence, deux étapes : représentation et affectation. La première consiste à rechercher les centres des classes et la seconde à reconstruire les classes en affectant les points aux classes dont le centre est le plus proche. La suite $\left(W^{(c)}(P,\mu)\right)$ décroît, converge et atteint sa limite, le couple de partition et noyaux constituera un optimum local pour le problème de classification. L'algorithme convergeant vers un optimum local dépendra de la position initiale. Différentes stratégies peuvent être considérées, la plus classique et la plus simple est d'initialiser l'algorithme par g points tirés au hasard ou encore par une partition en g classes tirées au hasard. Les différentes étapes de l'algorithme sont décrites dans l'algorithme 8.

Algorithme 8: L'algorithme $k$means

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Initialisation : Choix aléatoire d'une partition en g classes de $\mathbf{x}$;

Sortie : Les classes : $P_1, ..., P_g$.

Répéter

1 : Étape Représentation, calcul des g centres de gravité des classes.

2 : Étape Affectation, reconstitution des g nouvelles classes. Chaque $\mathbf{x}_i$ est affecté à la $k^{\text{ième}}$ classe telle que $k = \arg\min_{k'} d^2(\mathbf{x}_i, \mu_{k'})$.

jusqu'à *la convergence*;

2 La méthode des $k$means axiales (KMA)

La méthode des $k$means axiales, a été conçue pour la classification des données textuelles, elle utilise le produit scalaire comme mesure de similarité entre des vecteurs documents normés. Basée sur l'idée de McQueen (McQueen, 1967), après chaque affectation un centre est mis à jour. L'algorithme 9 associé à la méthode KMA étant de type $k$means, à chaque itération, il minimise le critère suivant :

$$W(P,\mu) = 2n - 2\sum_{i=1}^{n}\sum_{k=1}^{g} z_{ik}\mathbf{x}_i{}^t\mu_{k_i}$$

La minimisation du critère $W(P,\mu)$ revient à la maximisation du critère :

$$\sum_{i=1}^{n}\sum_{k=1}^{g} z_{ik}\mathbf{x}_i{}^t\mu_{k_i}$$

Algorithme 9: L'algorithme $k$means axiales

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Initialisation : tirage aléatoire de g vecteurs unitaires μ_k et g valeurs $\tau_k \succeq 0$

Répéter

1 : Calcul des g projections des $\mathbf{x}_i$ sur les axes μ_k.

2 : Affecter chaque $\mathbf{x}_i$ à la $k^{\text{ième}}$ classe telle que $k = \arg\max_{k'} \langle \mu_{k'}, \mathbf{x}_i \rangle$.

3 : Mise à jour de l'axe k par $\mu_k = \mu_k + \frac{\eta_k}{\tau_k(\mathbf{x}_i - \eta_k\mu_k)}$

avec $\tau_k = \tau_k + (\eta_k)^2$ et $\eta_k = \mathbf{x}_i{}^t\mu_{k_i}$

jusqu'à *la convergence*;

Les τ_k sont des valeurs d'apprentissage, pouvant être prises comme $t\lambda_k$ (loi d'Oja) avec λ_k étant les k premières valeurs propres de la matrice $\mathbf{x}^t\mathbf{x}$ (Lelu, 1993). Dans l'application des réseaux de neurones artificiels au domaine de l'infométrie, des études ont abouti au développement de la présente méthode en s'inspirant du formalisme neuronal des cartes auto-adaptatives de Kohonen (Lelu, 1993).

3 La méthode $k$means sphérique (SPK)

L'algorithme 10 associé à la méthode de *spherical* $k$means est similaire à un algorithme $k$means classique, ou plutôt à l'algorithme des nuées dynamiques. En effet, cet algorithme est appliqué sur les données normées et les centres des classes sont de même type.

Cet algorithme donne de bons résultats dans le contexte de classification de données textuelles de grande taille et est, de ce fait, efficace pour des matrices creuses. Pour

Algorithme 10: L'algorithme SPK

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Sortie : La partition : $P_1, ..., P_g$.

Initialisation : tirage aléatoire des g vecteurs unitaires, μ_k, $k = 1, \ldots, g$;

Répéter

1 : Étape Affectation, pour $i = 1, \ldots, n$, cette étape consiste à affecter $\mathbf{x}_i$ à la $k^{\text{ième}}$ classe avec $k = \arg\max_{k'} \mathbf{x}_i{}^t \mu_{k'}$

2 : L'ensemble des g centres $\mu_k^{(c)}$ sont définis par $\mu_k = \frac{\sum_{i=1}^n z_{ik}\mathbf{x}_i}{||\sum_{i=1}^n z_{ik}||}$

jusqu'à *la convergence*;

améliorer la qualité des résultats, Zhong (Zhong, 2005) a proposé une variante séquentielle (online), l'idée est de mettre à jour les centres des classes après l'affectation d'un nouvel individu $\mathbf{x}_i$ à une classe k qu'on note k_i. A chaque itération $(c-1)$, le centre maximisant le critère

$$\sum_{i=1}^{n} z_{ik}^{(c)} \mathbf{x}_i{}^t \mu_k, \tag{4.2}$$

est noté $\mu_{k_i}^{(c-1)}$, sa mise à jour après affectation est définie par l'expression suivante :

$$\mu_{k_i}^{(c)} = \frac{\mu_{k_i}^{(c-1)} + \eta^{(c)} x_i}{\left\| \mu_{k_i}^{(c-1)} + \eta^{(c)} x_i \right\|}$$

$\eta^{(c)}$ représente le taux d'apprentissage qui peut être choisi comme constant (par exemple 0.05), ou adaptatif suivant les cardinalités des classes (Zhong, 2005).

4 Discussion

Les deux algorithmes $k$means axiales et $k$means sphérique permettent de traiter les données textuelles de grande taille, issues par exemple du croisement d'un ensemble de documents et de mots. Ils exigent une normalisation des données pour profiter d'une représentation sphérique, où seule l'information donnée par les angles entre les documents nous intéresse. En effet, cela permet de pouvoir comparer des documents de différentes tailles. Le produit scalaire et le cosinus pour ce type de données présentent la même mesure de similarité, donc aboutissent au même critère à optimiser. Pour les $k$means axiales, les g classes sont représentées par des vecteurs pointant vers les zones de forte densité, ces classes sont définies par g demi-axes passant par l'origine de l'espace géométrique, ou g vecteurs unitaires pointant dans la direction de ces demi-axes mis à jours après chaque affectation

d'un document i dans une classe. Cette méthode est aussi considérée comme une méthode de recouvrement, paramétrée par un nombre maximal de classes désirées g, un document peut appartenir à plusieurs classes à la fois (Lelu, 1993). Le $k$means sphérique crée une partition de l'espace en des régions de Dirichlet (Voronoi) séparées par des hyperplans qui passent par l'origine (Dhillon *et al.*, 2001).

5 Quelques résultats expérimentaux

Nous proposons de comparer les deux algorithmes étudiés dans ce chapitre sur des données simulées. Pour ce faire, une étude expérimentale basée sur une validation et comparaison à partir des données simulées suivant le modèle de mélange de lois de von Mises-Fisher qui seront étudiés en détail dans le chapitre 5.

Le plan d'expérience tiendra compte de la taille des données, du degré de mélange et du nombre de classes. On a effectué des comparaisons des deux algorithmes à l'aide de simulations de Monte Carlo de quatre échantillons de paramètres différents (proportions et concentrations) et de tailles $n \times 3$, suivant la loi de von Mises-Fisher, les deux algorithmes ont été initialisés par les mêmes paramètres tirés au hasard. Le tableau ci-dessous contient les taux d'éléments mal classés de SPK et de KMA. On voit bien que les deux

n	Proportions	Concentrations	degré de mélange	SPK	KMA
600	$1/3 - 1/3 - 1/3$	$10 - 10 - 10$	0.2586	0.2649	0.2640
60	$1/3 - 1/3 - 1/3$	$90 - 5 - 50$	0.1444	0.2750	0.2756
1200	$0.5 - 0.3 - 0.2$	$90 - 5 - 50$	0.0535	0.1320	0.1369
1200	$0.5 - 0.3 - 0.2$	$10 - 10 - 10$	0.2469	0.3134	0.2997

algorithmes ont fourni les mêmes taux d'éléments mal classés, sauf pour le dernier cas où le degré de mélange est élevé et ses proportions sont différentes, dans ce cas la méthode du $k$means axiales donne un résultat meilleur que celui du $k$means sphérique. Signalons que KMA utilise des valeurs d'apprentissage τ_k au cours des itérations, ces valeurs sont destinées à l'amélioration des résultats, elles sont inconnues et représentent des paramètres supplémentaires à estimer. Une fonction décroissante dépendante de ce paramètre est souvent choisie par l'utilisateur. De plus KMA a été initialement destiné à la classification de données textuelles. Dans la suite de notre travail, des applications sur des données de type directionnel seront considérées, nous proposons alors d'utiliser SPK pour sa rapidité par rapport au KMA, de plus il représente une version très simple de type $k$means, sans aucune contrainte d'estimation de paramètres supplémentaires.

Chapitre 5

Approche ML des distributions vMF, algorithme EM$_{vMF}$ et ses variantes

Les directions dans $\mathbb{R}^d$ se présentent naturellement par des points de la sphère unité S^{d-1} ou, plus précisément, par les vecteurs unitaires correspondants. La loi de von Mises-Fisher peut être considérée en tant que loi d'erreur par rapport à une direction modale ; elle est invariante par rotation autour de cette direction. La densité de probabilité est maximale au voisinage de la direction modale et va en diminuant lorsqu'on se rapproche de la direction opposée (antimodale). Cette décroissance est mesurée par un paramètre dit de concentration et la direction modale est représentée par la moyenne directionnelle.

Les distributions normales multiples ont été l'aliment de base de la modélisation des données dans la plupart des domaines. Pour certains domaines comme pour le cas des données directionnelles, les modèles qu'ils fournissent sont soit insuffisants, ou incorrects. Par ailleurs, il a été observé que pour les données textuelles multidimensionnelles, le cosinus de similarité est plus performant qu'une métrique de type L_1 (Dhillon *et al.*, 2001) ; cette observation favorise l'attribution d'un modèle directionnel par rapport à d'autres modèles pour les données textuelles. Ce même modèle a été appliqué avec sucés à la modélisation de données génétiques, l'expression de ces dernières admet des caractéristiques directionnelles uniques (Dhillon et Sra, 2003). De ce fait, le modèle de von Mises-Fisher représente une modélisation naturelle du traitement de données directionnelles. Dans ce chapitre, nous exposons les démarches présentées par plusieurs auteurs (Mardia et Jupp, 2009), (Banerjee *et al.*, 2005), en essayant d'exploiter le principe de la classification de données directionnelles. Ensuite, nous nous focalisons sur l'approche modèle de mélange et proposerons plusieurs algorithmes de type EM.

Ce chapitre est organisé comme suit. La section 2 est une présentation d'ordre historique et théorique des lois de von Mises. La section 3 est une comparaison des lois de vMF avec la loi normale multidimensionnelle et un exposé rapide de quelques lois directionnelles existantes. Dans la section 4, on traite le problème des estimations des paramètres du mélange des distributions de vMF. Dans la section 5, on présente l'algorithme EM_{vMF} et on propose de nouvelles variantes de celui-ci, telles que le CEM_{vMF}, SEM_{vMF} et le SAEM_{vMF}, sur un plan théorique. Dans ce chapitre, on se concentre sur SPK, EM_{vMF} et CEM_{vMF}, à l'aide de données simulées des comparaisons des algorithmes sont réalisées dans la section 6. Enfin, nous terminons par une conclusion sur l'apport de l'approche mélange.

1 Présentation des lois de von Mises

Les lois de probabilités de von Mises décrivent des phénomènes aléatoires directionnels et jouent dans ce domaine un rôle analogue à celui que tiennent les lois normales pour des observations ponctuelles. La loi de von Mises noté $M(\mu, \xi)$ est connue aussi sous le nom de distribution normale circulaire, introduite en 1918 par von Mises (von Mises, 1918) pour l'étude des déviations des poids atomiques mesurés à partir de valeurs intégrales ; c'est une loi de probabilité continue sur le cercle unitaire (connue aussi sous le nom de distribution circulaire normale).

En statistique inférentielle, les distributions de von Mises sont plus intéressantes sur un cercle unitaire (Mardia et Jupp, 2009). D'autre part, la distribution de von Mises est une distribution stationnaire d'un processus de diffusion sur le cercle avec une orientation préférentielle, $M(\mu + \pi, \xi) = M(\mu, -\xi)$ où $\xi \geq 0$. C'est aussi une distribution d'entropie maximale pour une espérance de valeur donnée par $\exp(i\theta)$. Pour un vecteur donné $\mathbf{x}$ cette distribution s'écrit sous la forme :

$$f(\mathbf{x}; \mu, \xi) = \frac{\exp\left[\xi \mathbf{x}^t \mu\right]}{2\pi I_0(\xi)}$$

où $I_0(\xi)$ est la fonction de Bessel modifiée d'ordre 0, avec :

$$I_0(\xi) = \frac{1}{2\pi} \int_0^{2\pi} \exp\left[\xi \cos\theta\right] d\theta$$

μ est la moyenne directionnelle, elle est analogue à la moyenne statistique et ξ est une mesure de concentration.

Soit un groupe de vecteurs aléatoires, μ la moyenne directionnelle exprime l'orientation des données, ξ indique la concentration des données autour de μ.

Si la concentration est nulle, cette distribution est uniforme ; et si $\xi \to 0$ alors notre distribution est proche de la distribution uniforme. Dans le cas contraire, si ξ est grand

($\xi \to \infty$), alors la distribution de von Mises tend vers une distribution ponctuelle autour de μ, la figure (5.1) est une représentation de la densité $f(\mathbf{x}; 0, \xi)$ pour $\xi = 0, 2, 3, 4$. On remarque que pour $\xi = 4$ plus de 99% des probabilités sont dans l'intervalle $[-\frac{\pi}{2}, \frac{\pi}{2}]$, plus ξ augmente plus cet intervalle retrécit. Notons que la distribution de von Mises ne fut

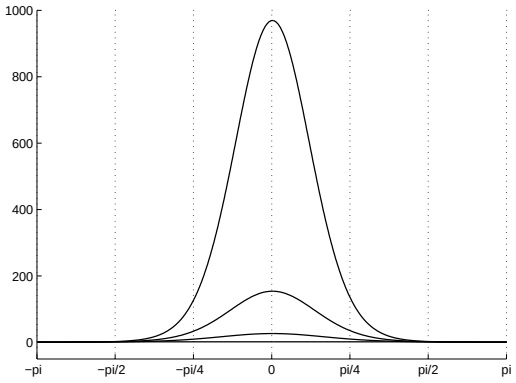

FIGURE 5.1 – Distribution $M(0, \xi)$ avec $\xi = 0, 2, 3, 4$

étendue en dimension 3 qu'en 1941 par Arnold (Arnold, 1941), elle portait le nom de loi de Fisher qui a entreprit son étude en 1953 (Fisher, 1953), définie sur S^2 en coordonnées sphériques, sa densité prend la forme :

$$f(\mathbf{x}; \theta) = \frac{\xi \sin \mathbf{x}}{4\pi sh\xi} \exp\left[\xi(\cos \mathbf{x} \cos \mu + \sin \mathbf{x} \sin \mu \cos(\mathbf{x} - \mu))\right]$$

Une généralisation à une dimension supérieure a été proposée par Stephens en 1962 (Stephens, 1962) et dont la distribution est de la forme :

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_{d-1}) = \frac{\xi^{\frac{d}{2}-1} \prod_{i=1}^{d-2} (\sin \mathbf{x}_i)^{d-i-1} \exp\left[\xi \cos \mathbf{x}_1\right]}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\xi)}$$

où la direction modale est définie par $\mathbf{x}_0 = 0$, cette forme a été améliorée pour des cas plus généraux (Dégerine, 1979).

La loi de von Mises multidimensionnelle, aussi dite von Mises-Fisher est présentée comme la loi exponentielle canonique par rapport à la loi uniforme sur la sphère unitaire S^{d-1} dans $\mathbb{R}^d$. Soit $\mathbf{x}$ un vecteur aléatoire unitaire ($\|\mathbf{x}\| = 1$) d-dimensionnel, $\mathbf{x}$ suit une variable de vMF, si sa densité de probabilité est donnée par :

$$f(\mathbf{x}; \mu, \xi) = c_d(\xi) \exp\left[\xi \mathbf{x}^t \mu\right],\ \mathbf{x},\ \mu \in S^{d-1} \subseteq \mathbb{R}^d,\ \text{ et } \xi \geqslant 0$$

La constante de normalisation $c_d(\xi)$ est définie par :

$$c_d(\xi) = \frac{\xi^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\xi)}$$

où $I_{\frac{d}{2}}(\xi)$ représente la fonction de Bessel modifiée du 1^{er} type d'ordre $\frac{d}{2}$. La distribution $M(\mu, \xi)$ est paramétrée par la moyenne directionnelle μ et le paramètre de concentration ξ caractérisant la force de concentration autour de la moyenne, ayant les mêmes propriétés que dans le cas circulaire.

Cette distribution est naturellement destinée aux données directionnelles et admet des propriétés analogues à celles d'une variable gaussienne multidimensionnelle dans $\mathbb{R}^d$. La densité qui maximise l'entropie sur S^{d-1} sous réserve que son espérance $E(\mathbf{x})$ prenne une valeur donnée, est une distribution de vMF (Mardia et Jupp, 2009) et (Rao, 1973).

2 Loi normale et distributions directionnelles

Les distributions directionnelles sont à l'origine des distributions exponentielles, dont plusieurs auteurs ont expliqué leurs liens avec la loi normale (Downs, 1972; Mardia, 1974; Kent, 1982; Kume et Walker, 2009). Nous résumons ici quelques liens entre les lois directionnelles et la loi normale.

2.1 Loi Gaussienne multidimensionnelle

La loi de Laplace-Gauss est une loi de variable aléatoire continue, elle doit son nom au prince des mathématiciens Carl Friderich Gauss, chercheur allemand (1777-1855). Cette loi est la plus importante des lois continues, des questions tant théoriques que pratiques font appel à celle-ci. Historiquement elle apparaît vers 1773 comme la forme limite de la loi binomiale (Abraham de Moivre), ensuite Gauss en 1809 et Laplace en 1812 lui donnèrent sa forme définitive. La loi gaussienne est dite aussi loi normale (dénomination revenant à K. Pearson), cette appellation est malencontreuse, mais significative, car malgré que cette loi est loin de décrire tous les phénomènes physiques, il faut se garder de considérer comme « anormal une variable ne suivant pas la loi normale » (Saporta, 2006).

La loi normale joue un rôle fondamental en probabilité et statistique mathématique, elle constitue un modèle fréquemment utilisé dans divers domaines (comme par exemple : la répartition des erreurs de mesure autour de la vraie valeur). En statistique, son rôle est principal, il provient de ce qu'elle apparaît réellement comme loi limite de caractéristiques liées à un échantillon de grande taille, ceci est démontré par le théorème de la limite centrale, on reconnaît aussi que si on a peu d'informations et beaucoup d'observations, l'échantillon considéré suit une loi gaussienne.

Les propriétés de cette loi sont très connues même dans le cas multidimensionnel, si $x \hookrightarrow N(\mu, V)$, sa densité est donnée par :

$$f(\mathbf{x}) = (2\pi)^{-\frac{d}{2}} |V|^{-\frac{1}{2}} exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^t V^{-1}(\mathbf{x}-\mu)\right]$$

où $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_d)$, μ est le vecteur moyen $E(\mathbf{x})$ et V est la matrice de variance-covariance de la distribution.

La densité de cette loi peut aussi s'écrire sous la forme :

$$f(\mathbf{x}) = (2\pi)^{-\frac{p}{2}} |V|^{-\frac{1}{2}} exp[d^2_{V^{-1}}(\mathbf{x}, \mu)]$$

où intervient un terme de distance de Mahalanobis entre $\mathbf{x}$ et la moyenne, V est régulière, alors : $\mathbf{y} = V^{\frac{-1}{2}}(\mathbf{x} - \mu)$ est un vecteur gaussien dont les composantes sont centrées-réduites, $\mathbf{y} \hookrightarrow N(0, I)$, aussi tout vecteur $\mathbf{z}$ de la forme $\mathbf{z} = A\mathbf{x} + b$ est gaussien de paramètres $(A + b, AVA^t)$. La matrice variance V est une matrice carrée symétrique de rang d, elle admet une paramétrisation en valeurs et vecteurs propres (Govaert, 1983) telle que $V = DBD^t$, où D est la matrice des vecteurs propres et B la matrice diagonale des valeurs propres. Pour obtenir une décomposition unique, les valeurs propres sont ordonnées suivant leurs valeurs décroissantes, puis la matrice B peut s'écrire sous la forme : $B = \beta A$, où β est un réel positif et A une matrice diagonale de déterminant égal à un ; alors, on obtient $V = \beta DAD^t$.

- β : réel positif représente le volume des données,
- A : matrice diagonale avec $|A| = 1$, elle représente la forme des données,
- et D : matrice orthogonale représente l'orientation des données.

Les courbes de niveau d'équation $f(\mathbf{x}) = cste$ sont des ellipsoïdes dans R^d et la distribution de la loi normale est :

$$f(\mathbf{x}) = (2\pi)^{-\frac{d}{2}} |V|^{-\frac{1}{2}} \exp\left[-\frac{1}{2\beta}(\mathbf{x} - \mu)^t (DAD^t)^{-1}(\mathbf{x} - \mu)\right]$$

2.2 Distribution de von Mises-Fisher et loi normale

On suppose que : $\mathbf{x} \hookrightarrow N(\mu, V)$, on considère le cas où $DAD^t = I$, la matrice de variance-covariance est $V = \beta I$, tel que $\beta > 0$ et I est la matrice identique, c'est le cas des formes sphériques, dans ce cas sa distribution est :

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \beta^{\frac{1}{2}}} \exp\left[-\frac{1}{2\beta}(\mathbf{x} - \mu)^t (\mathbf{x} - \mu)\right]$$

$$= \frac{1}{(2\pi)^{\frac{d}{2}} \beta^{\frac{1}{2}}} \exp\left[\frac{-1}{2\beta}(\mathbf{x}^t\mathbf{x} + \mu\mu) + \frac{1}{\beta}\mathbf{x}^t\mu\right]$$

$$= \frac{1}{(2\pi)^{\frac{d}{2}} \beta^{\frac{1}{2}}} \exp\left[\frac{-1}{2\beta}(1 + \|\mu\|^2)\right] \exp\left[\frac{1}{\beta}\mathbf{x}^t\mu\right],$$

la moyenne μ est l'espérance mathématique de la loi normale, par rapport à la moyenne directionnelle $\lambda = \frac{\mu}{\|\mu\|}$

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \beta^{\frac{1}{2}}} \exp\left[\frac{-1}{2\beta}(1 + \|\mu\|^2)\right] \exp\left[\|\mu\|\frac{1}{\beta}\mathbf{x}^t\lambda\right].$$

En posant $\varphi(\beta) = \frac{1}{(2\pi)^{\frac{d}{2}}\beta^{\frac{1}{2}}} \exp\left[\frac{-1}{2\beta}(1+\|\mu\|^2)\right]$, on obtient :

$$f(\mathbf{x}) = \varphi(\beta)\exp\left[\|\mu\|\frac{1}{\beta}\mathbf{x}^t\lambda\right].$$

A partir de cette expression, $f(\mathbf{x})$ est équivalente à une distribution de von Mises-Fisher de concentration $\xi = \frac{\|\mu\|}{\beta}$ (Mardia, 1974). Réciproquement, on peut montrer que toute loi de von Mises-Fisher est une loi normale de variance $\frac{1}{\beta}I$ où les formes des classes sont sphériques.

2.3 Distribution de Bingham

On suppose que $\mathbf{x} \hookrightarrow N(0, V)$, sa densité de probabilité est :

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|V|^{\frac{1}{2}}}\exp - \left[\frac{1}{2}\mathbf{x}^t V^{-1}\mathbf{x}\right].$$

En utilisant la décomposition spectrale de la matrice de covariance $V = MDM^t$, où M est une matrice orthogonale et D une matrice diagonale, alors :

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{(2\pi)^{\frac{d}{2}}|V|^{\frac{1}{2}}}\exp - \left[\frac{1}{2}\mathbf{x}^t(MDM^t)^{-1}\mathbf{x}\right] \\ &= \frac{1}{(2\pi)^{\frac{d}{2}}|V|^{\frac{1}{2}}}\exp - \left[\frac{1}{2}\mathbf{x}^t M^t D^{-1} M\mathbf{x}\right]. \end{aligned}$$

En posant $A = -\frac{1}{2}D$ et $B = M^t$, nous avons

$$\begin{aligned} -\frac{-1}{2}\mathbf{x}^t M^t D^{-1} M\mathbf{x} &= \mathbf{x}^t BA^{-1}B^t\mathbf{x} \\ &= trace(\mathbf{x}^t BA^{-1}B^t\mathbf{x}) \\ &= trace(A^{-1}B^t\mathbf{x}\mathbf{x}^t B). \end{aligned}$$

Donc $f(\mathbf{x})$ prend la forme suivante

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}\left|\frac{1}{2}B^t AB\right|^{\frac{1}{2}}}\exp\left[trace(A^{-1}B^t\mathbf{x}\mathbf{x}^t B)\right]$$

D'après leurs définitions : A est diagonale et B est orthogonale, on pose

$$\phi(A,B) = \frac{1}{(2\pi)^{\frac{d}{2}}\left|\frac{1}{2}B^t AB\right|^{\frac{1}{2}}}.$$

une constante de normalisation,

$$f(\mathbf{x}) = \phi(A,B)\exp\left[trace\left(A^{-1}B^t\mathbf{x}\mathbf{x}^t B\right)\right]$$

est l'expression d'une distribution de loi de Bingham (Bingham et Mardia, 1978), les deux distributions sont équivalentes, toute distribution de Bingham est une distribution normale centrée, où les formes des classes sont elliptiques.

2.4 Distribution de Fisher-Bingham

La distribution de Fisher-Bingham est construite à partir d'une distribution normale en imposant des contraintes sur la matrice de covariance dans l'espace des vecteurs normalisés (Kume et Walker, 2009), la fonction de densité de probabilité est la suivante :

$$f(\mathbf{x}) = C(\mu, \Sigma)^{-1} \exp\left[-(\mathbf{x}-\mu)^t \Sigma^{-1} (\mathbf{x}-\mu)\right]$$

où $C(\mu, \Sigma)$ est une constante de normalisation, μ est la moyenne directionnelle, Σ est la matrice de covariance.

En conclusion, Si on suppose être dans l'espace des vecteurs normalisés, la différence principale entre la loi normale et les distributions de type directionnel, n'est pas dans la forme mais dans le nombre et le type de paramètres estimés lors de l'analyse statistique :

- Pour la loi normale l'espérance mathématique est égale à la moyenne directionnelle à un coefficient multiplicatif constant.
- La matrice de covariance explique le volume, l'orientation et la forme de l'échantillon statistique. Pour la distribution de von Mises-Fisher, la concentration est proportionnelle au volume, l'orientation et la forme sont par contre non décrites.

Si on se place dans le cas de la classification automatique, cela nous ramène à un problème de reconnaissance de forme sphérique. Dans le cas de distributions directionnelles plus compliquées où le nombre de paramètres est supérieur à 5 (et même les familles exponentielles canoniques à 8 paramètres), toutes ces lois sont équivalentes aux lois de type normal. Le modèle définit par les distributions directionnelles est d'une qualité théorique comparable au modèle gaussien.

3 Estimations des paramètres de vMF

Soit $\mathbf{x}$ un échantillon de n vecteurs aléatoires générés indépendamment d'une loi de vMF, $\mathbf{x} = \mathbf{x}_1, \ldots, \mathbf{x}_n$ et

$$\mathbf{x}_i \rightsquigarrow M(\mu, \xi) \text{ pour } 1 \leqslant i \leqslant n$$

On cherche les estimateurs du maximum de vraisemblance de μ et ξ de la distribution $M(\mu, \xi)$, la log-vraisemblance de l'échantillon $\mathbf{x}$ est :

$$L(\mathbf{x}; \mu, \xi) = \log p(\mathbf{x}; \mu, \xi)$$

En supposant que les vecteurs $\mathbf{x}_i,\ i = 1, ..., n$ sont indépendants, on a :

$$\begin{aligned} p(\mathbf{x}; \mu, \xi) &= p(\mathbf{x}_1, ..., \mathbf{x}_n; \mu, \xi) \\ &= \prod_{i=1}^{n} f(\mathbf{x}_i; \mu, \xi) \\ &= \prod_{i=1}^{n} c_d(\xi) \exp\left[\xi \mathbf{x}_i^t \mu\right] \end{aligned}$$

alors :

$$\begin{aligned} \log p(\mathbf{x}; \mu, \xi) &= \log\left(\prod_{i=1}^{n} c_d(\xi) \exp\left[\xi \mathbf{x}_i^t \mu\right]\right) \\ &= \sum_{i=1}^{n} \left[\log\left(c_d(\xi)\right) + \xi \mathbf{x}_i^t \mu\right] \\ &= n \log\left(c_d(\xi)\right) + \xi \mu^t \sum_{i=1}^{n} \mathbf{x}_i. \end{aligned}$$

En posant $\sum_{i=1}^{n} \mathbf{x}_i = \mathbf{r}$, l'expression de $\log p(\mathbf{x}; \mu, \xi)$ devient

$$\log p\left(\mathbf{x}; \mu, \xi\right) = n \log\left(c_d(\xi)\right) + \xi \mu^t \mathbf{r},$$

et l'estimation de μ vérifie :

$$\frac{\partial \log p\left(\mathbf{x}; \mu, \xi\right)}{\partial \mu} = 0,$$

et optimise le Lagrangien :

$$Lag = n \log\left(c_d(\xi)\right) + \xi \mu^t \mathbf{r} - \lambda\left(\mu^t \mu - 1\right), \text{ avec } \mu^t \mu = \|\mu\|^2 = 1.$$

$$\frac{\partial Lag}{\partial \mu} = 0 \Longrightarrow \frac{\partial Lag}{\partial \mu} = \xi \mathbf{r} - 2\lambda\mu = 0 \Longrightarrow \mu = \frac{\xi \mathbf{r}}{2\lambda}.$$

On en déduit,

$$\frac{\xi^2 \|r\|^2}{4\lambda^2} = 1 \Longrightarrow \lambda = \frac{\xi \|r\|}{2},$$

d'où,

$$\Longrightarrow \widehat{\mu} = \frac{\xi r}{2\frac{\xi \|r\|}{2}} = \frac{r}{\|r\|}.$$

D'autre part, l'estimation de ξ vérifie :

$$\frac{\partial \log p\left(\mathbf{x}; \mu, \xi\right)}{\partial \xi} = 0 \Longrightarrow n \frac{c_d'\left(\xi\right)}{c_d\left(\xi\right)} + \mu^t r = 0 \Longrightarrow \frac{c_d'(\xi)}{c_d(\xi)} = -\frac{\mu^t r}{n}$$

$$\Longrightarrow \frac{c_d'(\xi)}{c_d(\xi)} = \frac{\frac{\left(\frac{d}{2}-1\right)\xi^{\frac{d}{2}-2}(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\xi) - \xi^{\frac{d}{2}-1}(2\pi)^{\frac{d}{2}} I'_{\frac{d}{2}-1}(\xi)}{(2\pi)^d I^2_{\frac{d}{2}-1}(\xi)}}{\frac{\xi^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\xi)}} = \frac{\frac{\left(\frac{d}{2}-1\right)\xi^{-1} I_{\frac{d}{2}-1}(\xi) - I'_{\frac{d}{2}-1}(\xi)}{I^2_{\frac{d}{2}-1}(\xi)}}{\frac{1}{I_{\frac{d}{2}-1}(\xi)}}$$

$$= \frac{1}{I_{\frac{d}{2}-1}(\xi)} \left[\frac{\left(\frac{d}{2}-1\right)}{\xi} I_{\frac{d}{2}-1}(\xi) - I'_{\frac{d}{2}-1}(\xi) \right]$$

En utilisant la relation de récurrence ci-dessous propre aux fonctions de Bessel :

$$I_{\frac{d}{2}}(\xi) = I'_{\frac{d}{2}-1}(\xi) - \frac{\frac{d}{2}-1}{\xi} I_{\frac{d}{2}-1}(\xi)$$

$$\Longrightarrow \frac{c'_d(\xi)}{c_d(\xi)} = -\frac{I_{\frac{d}{2}}(\xi)}{I_{\frac{d}{2}-1}(\xi)}$$

$$\Longrightarrow \frac{I_{\frac{d}{2}}(\xi)}{I_{\frac{d}{2}-1}(\xi)} = \frac{\mu^t r}{n} = \frac{\frac{r^t}{\|r\|} r}{n} = \frac{\|r\|}{n}$$

Cette dernière expression est loin de donner une estimation directe et implicite de ξ, dans (Mardia et Jupp, 2009) et (Banerjee *et al.*, 2005) ; les auteurs ont utilisé des approximations asymptotiques comme une meilleure solution du problème. Dans (Mardia et Jupp, 2009) deux approximations ont été proposées suivant la valeur de $\bar{\mathbf{r}} = \frac{\|\mathbf{r}\|}{n}$:

$$\xi \approx \begin{cases} \xi_1 = \frac{d-1}{2(1-\bar{r})}, & \text{si } \bar{r} \text{ est grand} \\ \xi_2 \bar{r} d \left(1 + \frac{d}{d+2}\bar{r}^2 + \frac{d^2(d+8)}{(d+2)^2(d+4)}\bar{r}^4\right), & \text{si } \bar{r} \text{ est petit} \end{cases} \tag{5.1}$$

Ces approximations ont été principalement évaluées avec succès pour $d = 2, 3$. Dans la grande dimension, une approximation a été proposée par Banerjee et al. (Banerjee *et al.*, 2005) :

$$\xi_3 \approx \frac{\bar{r}d - \bar{r}^3}{1 - \bar{r}^2}. \tag{5.2}$$

Nous procédons à une comparaison entre les trois expressions d'approximation, afin de sélectionner celle que nous utiliserons dans la suite de notre travail.

Dans la figure 5.2, on représente la fonction $\bar{r}(d)$, $\xi = 1 : 500$, nous remarquons que $\bar{r}$ est une fonction décroissante qui prend des valeurs positives inférieure à un. Par conséquent et dans tous ce qui suit, on prend $\bar{r} \in [0, 1]$. La figure 5.3 représente graphiquement les trois fonctions $\xi 1,\ \xi 2$ et $\xi 3$ associées aux trois expressions d'approximation citées ci-dessus en fonction de $\bar{r} \in [0, 1]$.

Pour des dimensions réduites représentées par $\bar{r}$ proche de 1, les courbes associées à ξ_1 et ξ_3 se rapprochent, pour le cas inverse de grande dimension où $\bar{r}$ est proche de 0, c'est la courbe de ξ_2 qui est plus proche de ξ_3.

Dans la figure 5.4 sont reportés les écarts : $e_i = \xi - \xi_i,\ i = 1, 2, 3$ pour différentes valeurs

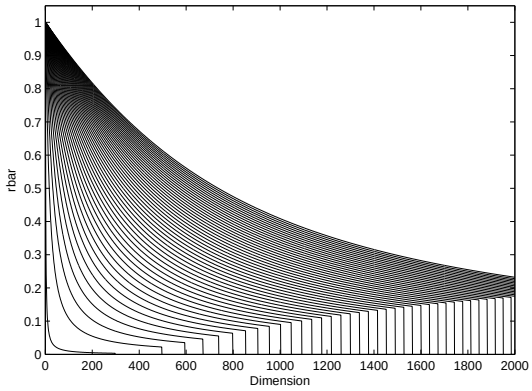

FIGURE 5.2 – Représentation graphique de la fonction $\bar{r}(d), \xi = 1 : 500$.

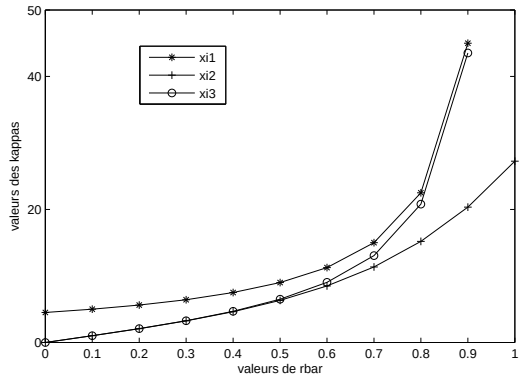

FIGURE 5.3 – Représentation graphique des trois estimateurs de ξ en fonction de $0 \leq \bar{r} \leq 1$

prédéfinies de ξ :
($\xi = 4, 10, 50, 100, 200, 300, 400, 500, 600, 700, 800, 1000$). On observe que e_1 est minimal pour des petites dimensions ($d < 100$), cet écart est visiblement grand pour des dimensions plus grandes. L'écart e_2 admet un comportement contraire, il diminue avec l'augmentation de la dimension ; alors que e_3 est stable et proche de 0 pour n'importe quelle situation.

En conclusion l'approche proposée dans (Banerjee *et al.*, 2005) représente la meilleure approximation, nous la retiendrons dans la suite de notre travail.

4 Approche ML pour un mélange de lois de vMF

Soit $\mathbf{y} = (\mathbf{x}, \mathbf{z})$, un tableau de données tel que : $\mathbf{x}$ est un tableau à n individus et d variables, $\mathbf{z}$ un vecteur de taille n qui exprime une information supplémentaire sur $\mathbf{x}$: chaque élément de ce tableau provient du composant k d'une loi de distribution de vMF, $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ telle que $\mathbf{z}_i = k$ quand $\mathbf{x}_i$ est généré suivant la loi $M(\mu_k, \xi_k)$. La distribution de la loi de probabilité définie sur le tableau de données complétées $\mathbf{y}$ s'écrit

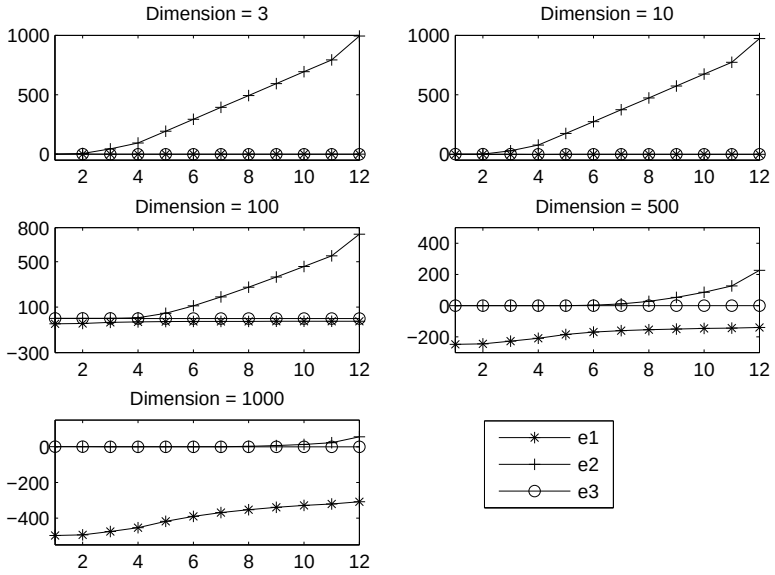

FIGURE 5.4 – Représentation graphique des erreurs $e_i,\ i = 1, 2, 3$ pour différentes valeurs de dimension

sous la forme suivante :

$$p(\mathbf{y};\theta) = p(\mathbf{x},\mathbf{z};\theta) = p(\mathbf{z},\mathbf{x};\theta)p(\mathbf{x};\theta)$$

ainsi la log-vraisemblance des données complétées prend cette forme :

$$\log p(\mathbf{y};\theta) = \log p(\mathbf{x},\mathbf{z};\theta) = \sum_{i=1}^{n} \log\left[\pi_{\mathbf{z}_i} f_{\mathbf{z}_i}\left(\mathbf{x}_i;\alpha_{\mathbf{z}_i}\right)\right]$$

A partir de cette expression on va estimer les paramètres de la loi associée au mélange, les valeurs de la variable cachée $\mathbf{z}$ peuvent être estimées à leur tour. Pour ce faire, on utilise l'algorithme EM ; la log-vraisemblance du tableau $\mathbf{x}$ obtenue à partir de la log-vraisemblance des données complétées s'écrit :

$$L(\mathbf{x};\theta) = L(\mathbf{y};\theta) - \log p(\mathbf{z},\mathbf{x};\theta)$$

L'espérance conditionnelle à $\mathbf{x}$ et au paramètre courant θ' de $L(\mathbf{x};\theta)$ est :

$$L(\mathbf{x};\theta) = E\left(L(\mathbf{y};\theta);\mathbf{x};\theta\right) - E\left(\log p(\mathbf{z},\mathbf{x};\theta);\mathbf{x};\theta'\right)$$

On pose :

$$Q\left(\theta,\theta'\right) = E\left(L(\mathbf{y};\theta);\mathbf{x};\theta'\right)$$

Maximiser $L(\mathbf{x};\theta)$ revient à maximiser $Q(\theta,\theta')$ qu'on peut simplifier de la manière suivante :

$$\begin{aligned} Q(\theta,\theta') &= E\left(L(\mathbf{y};\theta);\mathbf{x};\theta'\right) \\ &= E\left(\log p(\mathbf{y};\theta);\mathbf{x};\theta'\right) \\ &= E\left(\log p(\mathbf{x},\mathbf{z};\theta);\mathbf{x};\theta'\right) \\ &= E\left(\log \prod_{i=1}^{n} p(\mathbf{z}_i;\mathbf{x};\theta) f(\mathbf{x}_i;\theta);\mathbf{x};\theta'\right) \\ &= \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik} \log\left[\pi_k c_d(\xi)\exp\left(\xi\mu^t\mathbf{x}_i\right)\right], \end{aligned}$$

où t_{ik} est la probabilité a posteriori, que l'individu i provienne du composant k :

$$t_{ik} = p\left(\mathbf{x}_i \in P_k|\mathbf{x};\theta'\right) = p\left(z_{ik}=1|\mathbf{x};\theta'\right)$$

Ci-après, nous décrivons les principales étapes de l'algorithme EM pour l'estimation des paramètres du modèle de mélange von-Mises.

Algorithme 11: EM$_{vMF}$

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Initialisation : choix de façon arbitraire de $\theta_k = (\pi_k, \mu_k, \xi_k)$, $k = 1, ..., g$.

Sortie : paramètre : $\theta = (\pi_1, \ldots, \pi_g, \mu_1, \ldots, \mu_g, \xi_1, \ldots, \xi_g)$.

Répéter

1 : Étape Expectation E : calcul de t_{ik} :

$$t_{ik} = p(\mathbf{x}_i \in P_k|\mathbf{x};\theta) = \frac{p(\mathbf{x}_i;\theta_k)}{p(\mathbf{x}_i;\theta)} = \frac{\pi_k f(\mathbf{x}_i;\alpha_k)}{\sum_{k=1}^{g}\pi_k f(\mathbf{x}_i;\alpha_k)}$$

2 : Étape Maximisation M : Calcul de θ maximisant $Q(\theta,\theta')$

$$\theta = (\pi_k, \alpha_k), \qquad \theta = \arg\max_{\theta'} Q\left(\theta,\theta'\right)$$

jusqu'à *atteindre un état final fixé au départ*;

4.1 Descriptions des étapes de EM$_{vMF}$

Au lieu de maximiser directement la vraisemblance des données observées, l'algorithme EM$_{vMF}$ maximise à chaque itération (c) :

$$Q\left(\theta,\theta^{(c)}\right) = \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log\pi_k + \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log f(\mathbf{x};\mu_k,\xi_k)$$

Analytiquement cela conduit à résoudre les équations :

$$\begin{cases} \frac{\partial Q\left(\theta,\theta^{(c)}\right)}{\partial \pi_k} = 0 \\ \frac{\partial Q\left(\theta,\theta^{(c)}\right)}{\partial \mu_k} = 0 \\ \frac{\partial Q\left(\theta,\theta^{(c)}\right)}{\partial \xi_k} = 0 \end{cases}$$

1. Pour estimer π_k, $k = 1, \ldots, g$, on calcule le Lagrangien de $Q\left(\theta, \theta^{(c)}\right)$:

$$Lag\left(Q\left(\theta,\theta^{(c)}\right)\right) = Q\left(\theta,\theta^{(c)}\right) - \lambda\left(\sum_{k=1}^{g}\pi_k - 1\right)$$

on dérive :

$$\frac{\partial Lag\left(Q\left(\theta,\theta^{(c)}\right)\right)}{\partial \pi_k} - \lambda = 0 \Longleftrightarrow \sum_{i=1}^{n}\frac{t_{ik}^{(c)}}{\pi_k} = \lambda \Longleftrightarrow \pi_k = \sum_{i=1}^{n}\frac{t_{ik}^{(c)}}{\lambda}$$

$$\begin{cases} \sum\limits_{k=1}^{g}\pi_k = 1 \\ \sum\limits_{k=1}^{g}\sum\limits_{i=1}^{n} t_{ik} = n \end{cases} \Rightarrow \lambda = n$$

d'où :

$$\pi_k = \frac{\sum\limits_{i=1}^{n} t_{ik}^{(c)}}{n}.$$

2. Estimation de $\mu_k = (\mu_{k1}, \ldots, \mu_{kd})$ maximisant $Q\left(\theta, \theta^{(c)}\right)$ ou encore $\sum\limits_{k=1}^{g}\sum\limits_{i=1}^{n} t_{ik}^{(c)} \log f(\mathbf{x}_i; \mu_k, \xi_k)$ qui dépend de la loi de distribution de vMF.
Pour $k = 1, ..., g$ on a :

$$\sum_{i=1}^{n} t_{ik}^{(c)} \log f(\mathbf{x}_i; \mu_k, \xi_k) = \sum_{i=1}^{n} t_{ik}^{(c)} \log\left[c_d\left(\xi_k\right)\exp\left(\xi_k\left(\mu_k\right)^t \mathbf{x}_i\right)\right]$$

$$= \sum_{i=1}^{n} t_{ik}^{(c)}\left[\log c_d\left(\xi_k\right) + \xi_k\left(\mu_k\right)^t \mathbf{x}_i\right]$$

$$= \left(\log c_d\left(\xi_k\right)\right)\sum_{i=1}^{n} t_{ik}^{(c)} + \xi_k\left(\mu_k\right)^t \sum_{i=1}^{n} t_{ik}^{(c)}\mathbf{x}_i$$

Le Lagrangien de cette expression est donné par :

$$Lag = \left(\log c_d\left(\xi_k\right)\right)\sum_{i=1}^{n} t_{ik}^{(c)} + \sum_{i=1}^{n} t_{ik}^{(c)}\xi_k\left(\mu_k\right)^t \mathbf{x}_i - \lambda\left(\left(\mu_k\right)^t \mu_k - 1\right)$$

$\left(\mu_k\right)^t \mu_k = \|\mu_k\|^2 = 1$, l'estimateur de μ_k vérifie :

$$\frac{\partial Lag}{\partial \mu_k} = 0 \Longrightarrow \sum_{i=1}^{n} t_{ik}^{(c)}\xi_k\mathbf{x}_i - \lambda\left(2\mu_k\right) = 0$$

$$\Longrightarrow \mu_k = \frac{\xi_k}{2\lambda} \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i.$$

$$\|\mu_k\|^2 = 1 \Longrightarrow \left(\frac{\xi_k \left\| \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i \right\|}{2\lambda} \right)^2 = 1 \Longrightarrow \lambda = \frac{\xi_k}{2} \left\| \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i \right\|$$

$$\Longrightarrow \mu_k = \frac{\sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i}{\left\| \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i \right\|}$$

3. - On estime ξ_k, celui-ci maximise $Q\left(\theta, \theta^{(c)}\right)$ et minimise $\sum_{i=1}^{n} t_{ik}^{(c)} \log f(\mathbf{x}_i; \mu_k, \xi_k)$.

$$\frac{\partial \left[\sum_{i=1}^{n} t_{ik}^{(c)} \log f(\mathbf{x}_i; \mu_k, \xi_k) \right]}{\partial \xi_k} = 0$$

$$\Longrightarrow \frac{\partial \left[(\log c_d(\xi_k)) \sum_{i=1}^{n} t_{ik}^{(c)} + \xi_k (\mu_k)^t \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i \right]}{\partial \xi_k} = 0$$

$$\Longrightarrow \frac{c_d'(\xi_k)}{c_d(\xi_k)} \sum_{i=1}^{n} t_{ik}^{(c)} + (\mu_k)^t \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i = 0$$

$$\Longrightarrow \frac{c_d'(\xi_k)}{c_d(\xi_k)} = -\frac{(\mu_k)^t \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i}{\left(\sum_{i=1}^{n} t_{ik}^{(c)} \right)}$$

$$\frac{c_d'(\xi_k)}{c_d(\xi_k)} = \frac{\frac{\left(\frac{d}{2}-1\right)(\xi_k)^{\frac{d}{2}-2}(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\xi_k) - (\xi_k)^{\frac{d}{2}-1}(2\pi)^{\frac{d}{2}} I'_{\frac{d}{2}-1}(\xi_k)}{(2\pi)^d I^2_{\frac{d}{2}-1}(\xi_k)}}{\frac{(\xi_k)^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\xi_k)}}$$

$$= \frac{\frac{\left(\frac{d}{2}-1\right)(\xi_k)^{-1} I_{\frac{d}{2}-1}(\xi_k) - I'_{\frac{d}{2}-1}(\xi_k)}{I^2_{\frac{d}{2}-1}(\xi_k)}}{\frac{1}{I_{\frac{d}{2}-1}(\xi_k)}}$$

$$= \frac{1}{I_{\frac{d}{2}-1}(\xi_k)} \left[\frac{\left(\frac{d}{2}-1\right)}{\xi_k} I_{\frac{d}{2}-1}(\xi_k) - I'_{\frac{d}{2}-1}(\xi_k) \right]$$

$$= -\frac{I_{\frac{d}{2}}(\xi_k)}{I_{\frac{d}{2}-1}(\xi_k)}$$

$$\Longrightarrow \frac{I_{\frac{d}{2}}(\xi_k)}{I_{\frac{d}{2}-1}(\xi_k)} = \frac{(\mu_k)^t \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i}{\left(\sum_{i=1}^{n} t_{ik}^{(c)}\right)} = \frac{\left(\sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i\right)^t}{\left\|\sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i\right\|} \frac{\left(\sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i\right)}{\left(\sum_{i=1}^{n} t_{ik}^{(c)}\right)}$$

$$\Longrightarrow \frac{I_{\frac{d}{2}}(\xi_k)}{I_{\frac{d}{2}-1}(\xi_k)} = \frac{\left\|\sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i\right\|}{\left(\sum_{i=1}^{n} t_{ik}^{(c)}\right)}$$

Si on pose : $A_d(\xi_k) = \frac{I_{\frac{d}{2}}(\xi_k)}{I_{\frac{d}{2}-1}(\xi_k)}$ alors ξ_k peut s'écrire sous la forme :

$$\xi_k = A_d^{-1}\left(\frac{\left\|\sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i\right\|}{\left(\sum_{i=1}^{n} t_{ik}^{(c)}\right)}\right).$$

En reprenant l'expression de l'estimation de ξ donnée par la formule 5.2, résultat du paragraphe 3, on obtient

$$\xi_k = \frac{\bar{r}_k d - \bar{r}_k^3}{1 - \bar{r}_k^2} \text{ avec } \bar{r}_k = \frac{\left\|\sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i\right\|}{\left(\sum_{i=1}^{n} t_{ik}^{(c)}\right)}.$$

A la convergence, chaque composant du mélange est caractérisé par les paramètres estimés à l'étape M et nous obtenons $\hat{\theta}$ à partir duquel nous en déduisons

$$t_{ik} = p\left(\mathbf{x}_i \in P_k; \mathbf{x}; \hat{\theta}\right) = \frac{f\left(\mathbf{x}_i; \hat{\theta}_k\right)}{p\left(\mathbf{x}_i; \hat{\theta}\right)} = \frac{\hat{\pi}_k f\left(\mathbf{x}_i; \hat{\alpha}_k\right)}{\sum_{k=1}^{g} \hat{\pi}_k f\left(\mathbf{x}_i; \hat{\alpha}_k\right)}.$$

4.2 Variantes de l'algorithme $\mathbf{EM}_{vMF}$

On suppose que le tableau de données $\mathbf{x}$ est un mélange de g échantillons provenant de g composants suivant la loi de vMF. En se basant sur les deux schémas soft- et hard- (Kearns *et al.*, 1997), dans (Banerjee *et al.*, 2005), les auteurs ont proposé deux algorithmes d'estimation des paramètres d'un mélange de lois de vMF, l'algorithme soft-movMF et l'algorithme hard-movMF, le premier algorithme estime les paramètres du modèle suivant les calculs décrits dans le paragraphe 4.1, utilisant les probabilités a posteriori. Dans le deuxième schéma, les probabilités a posteriori en chaque point indique le composant origine du point et à une étape estimation, ces probabilités sont remplacées par une valeur z_{ik} (1 ou 0), à la convergence les deux algorithmes fournissent g paramètres $\theta = (\pi, \alpha)$ associés aux g composants du mélange.

L'algorithme Soft-movMF (Banerjee *et al.*, 2005), est décrit dans algorithme 12.

L'algorithme hard-movMF (Banerjee *et al.*, 2005), est décrit dans algorithme 13.

Algorithme 12: Soft-movMF

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Initialisation : choix de façon arbitraire de $\theta_k = (\pi_k, \mu_k, \xi_k)$, $k = 1, ..., g$.

Sortie : Les paramètres : $(\pi_1, \ldots, \pi_g, \mu_1, \ldots, \mu_g, \xi_1, \ldots, \xi_g)$.

Répéter

1 : Étape Expectation E : $i = 1, \ldots, n,\ k = 1, \ldots, g$

$$f(\mathbf{x}_i; \mu_k, \xi_k) = c_d(\xi_k) \exp(\xi_k \mathbf{x}_i^t \mu_k)$$

$$t_{ik} = \frac{\pi_k f(\mathbf{x}_i; \mu_k, \xi_k)}{\sum_{k=1}^{g} \pi_k f(\mathbf{x}_i; \mu_k, \xi_k)}$$

2 : Étape Maximisation M : $k = 1, \ldots, g$

$$\begin{cases} \pi_k = \frac{1}{n} \sum_{i=1}^{n} t_{ik} \\ \mu_k = \frac{\sum_{\mathbf{x}_i \in P_k} t_{ik} \mathbf{x}_i}{\left\| \sum_{\mathbf{x}_i \in P_k} t_{ik} \mathbf{x}_i \right\|} \\ \bar{r}_k = \frac{\left\| \sum_{i=1}^{n} t_{ik} \mathbf{x}_i \right\|}{\sum_{i=1}^{n} t_{ik}} \\ \xi_k = \frac{\bar{r}_k d - \bar{r}_k^3}{1 - \bar{r}_k^2} \end{cases}$$

jusqu'à *la convergence*;

5 Approche CML pour les mélanges de lois de vMF

Le tableau $\mathbf{x}$ des données est supposé provenir d'un mélange de g lois de vMF. Dans l'algorithme proposé dans (Celeux et Govaert, 1992)ont introduit une étape classification C entre les deux étapes E et M. On considère le tableau de données complétées $\mathbf{y} = (\mathbf{x}, \mathbf{z})$, le but est de définir une partition du tableau $\mathbf{x}$ en g classes $P = \{P_1, ..., P_g\}$. Soit $\mathbf{z}$ un vecteur décrivant la partition associée à $\mathbf{x}$, chaque composante indique l'étiquette de la classe d'appartenance. Ainsi $\mathbf{z}_i = k$ indique $\mathbf{x}_i \in P_k$. Ci-après nous décrivons les différentes étapes de cet algorithme appelé CEM_{vMF}. Notons que celui-ci consiste à maximiser une vraisemblance classifiante.

l'algorithme SPK-means est vu comme un cas particulier des algorithmes EM de type classification, lorsque $\xi_k = \xi$ et $\pi_k = \pi = \frac{1}{g}$ sont constants pour toutes les classes, dans ce cas au lieu de maximiser $p(\mathbf{x}_i; \theta_k)$, on a $k = \arg\max_{k'} \left(\mu_{k'}^t \mathbf{x}_i\right)$ et $p(k; \mathbf{x}_i, \theta_k) \longrightarrow 1$.

Algorithme 13: Hard-movMF

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Initialisation : choix de façon arbitraire de $\theta_k = (\pi_k, \mu_k, \xi_k)$, $k = 1, ..., g$.

Sortie : Les paramètres : $(\pi_1, \ldots, \pi_g, \mu_1, \ldots, \mu_g, \xi_1, \ldots, \xi_g)$ et la partition P.

Répéter

1 : Étape Expectation E : $i = 1, \ldots, n,\ k = 1, \ldots, g$

$$f(\mathbf{x}_i; \mu_k, \xi_k) = c_d(\xi_k) \exp(\xi_k \mathbf{x}_i^t \mu_k)$$

$$q(k; \mathbf{x}_i, \theta) = \begin{cases} 1, & \text{si } k = \arg\max\limits_{k'} [\pi_{k'} f(\mathbf{x}_i; \theta_{k'})] \\ 0, & \text{sinon} \end{cases}$$

2 : Étape Maximisation M : $k = 1, \ldots, g$

$$\begin{cases} \pi_k = \frac{1}{n} \sum\limits_{i=1}^{n} q(k; \mathbf{x}_i, \theta) \\ \mu_k = \frac{\sum\limits_{\mathbf{x}_i \in P_k} q(k;\mathbf{x}_i,\theta)\mathbf{x}_i}{\left\| \sum\limits_{\mathbf{x}_i \in P_k} q(k;\mathbf{x}_i,\theta)\mathbf{x}_i \right\|} \\ \bar{r}_k = \frac{\left\| \sum\limits_{i=1}^{n} q(k;\mathbf{x}_i,\theta)\mathbf{x}_i \right\|}{\sum\limits_{i=1}^{n} q(k;\mathbf{x}_i,\theta)} \\ \xi_k = \frac{\bar{r}_k d - \bar{r}_k^3}{1 - \bar{r}_k^2} \end{cases}$$

jusqu'à *la convergence*;

6 Approche stochastique, algorithme SEM$_{vMF}$

L'algorithme SEM est un algorithme de type EM, il introduit une étape stochastique S entre les deux étapes E et M dans le but d'estimer par une approche d'apprentissage probabiliste les composants d'un mélange fini de lois de probabilité (Celeux et Diebolt, 1986). Notons qu'il s'agit d'une convergence en loi. L'étape intermédiaire S est une simulation de Monte Carlo, elle consiste à tirer à chaque itération une partition selon la probabilité a posteriori $t_{ik}^{(c)} = p(\mathbf{x}_i \in P_k | \mathbf{x}; \theta')$ calculée à l'étape E. L'estimation des paramètres est basée sur ce tirage aléatoire à une étape maximisation M, en ajustant le nombre de classes aux données. Au départ, on fixe g un majorant supposé du nombre de composants du mélange et un seuil $c(n, d)$, $c \in [0, 1]$. SEM$_{vMF}$ est explicité dans l'algorithme 15.

Algorithme 14: CEM$_{vMF}$

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Initialisation : choix de façon arbitraire de $\theta_k = (\pi_k, \mu_k, \xi_k)$, $k = 1, ..., g$.

Sortie : Les paramètres : $(\pi_1, \ldots, \pi_g, \mu_1, \ldots, \mu_g, \xi_1, \ldots, \xi_g)$ et la partition P.

Répéter

1 : Étape Expectation E : calcule de t_{ik} :

$$t_{ik} = p\left(\mathbf{x}_i \in P_k | \mathbf{x}; \theta\right) = \frac{p\left(\mathbf{x}_i; \theta_k\right)}{p\left(\mathbf{x}_i; \theta\right)} = \frac{\pi_k f\left(\mathbf{x}_i; \alpha_k\right)}{\sum\limits_{k=1}^{g} \pi_k f\left(\mathbf{x}_i; \alpha_k\right)}$$

2 : Étape Classification C : classification et création de la partition :

$$P_k = \left\{ \mathbf{x}_i | t_{ik} = \max_{k'} t_{ik'} \right\}$$

$$z_{ik} = \begin{cases} 1, & \text{si } \mathbf{x}_i \in P_k \\ 0, & \text{sinon} \end{cases}$$

3 : Étape Maximisation M : maximiser la vraisemblance classifiante $L_C(\theta)$:

$$\begin{cases} \pi_k = \frac{1}{n} \sum\limits_{i=1}^{n} z_{ik} \\ \mu_k = \frac{\sum\limits_{\mathbf{x}_i \in P_k} z_{ik} \mathbf{x}_i}{\left\| \sum\limits_{\mathbf{x}_i \in P_k} z_{ik} \mathbf{x}_i \right\|} \\ \bar{r}_k = \frac{\left\| \sum\limits_{i=1}^{n} z_{ik} \mathbf{x}_i \right\|}{\sum\limits_{i=1}^{n} z_{ik}} \\ \xi_k = \frac{\bar{r}_k d - \bar{r}_k^3}{1 - \bar{r}_k^2} \end{cases}$$

jusqu'à *atteindre un état final fixé au départ*;

7 Approche hybride, algorithme SAEM$_{vMF}$

Malgré sa popularité, l'algorithme EM présente quelques défauts : dépendance de la condition initiale qui peut conduire à des optimums locaux non pertinents, la lenteur de la convergence, nécessite la connaissance du nombre de composants et son inefficacité devant des tableaux de données de taille modeste. L'algorithme SEM peut en partie remédier à certains défauts telle que la dépendance de la position initiale et le choix du nombre de composants du mélange. Pour exploiter les avantages de l'algorithme EM et SEM, un autre algorithme de type EM appelé SAEM (simulated annealing) et qui est une version de type

Algorithme 15: SEM$_{vMF}$

Entrée : $\mathbf{x}$: la matrice des données, le paramètre g majorant du nombre de composants du mélange et d'un seuil $c(n)$ compris entre 0 et 1.

Sortie : Le nombre de classes final. La partition : $P_1, ..., P_g$.

Initialisation : Choix aléatoire des paramètres du mélange $\theta_k, k = 1, \ldots, g$;

Répéter

1 : Étape Estimation E : estimation des t_{ik} $k = 1, ..., g$:

$$t_{ik} = \frac{\pi_k f(\mathbf{x}_i; \alpha_k)}{\sum_{k=1}^{g} \pi_k f(\mathbf{x}_i; \alpha_k)}$$

2 : Étape Stochastique S : On tire en chaque point $\mathbf{x}_i$ la variable aléatoire multinomiale $e^n(\mathbf{x}_i)$ d'ordre 1 et de paramètre $t_{ik}, k = 1, ..., g$, telle que :

$$e^n(\mathbf{x}_i) = (e_k^n(\mathbf{x}_i); k = 1, \ldots, g)$$

Les réalisations $e^n(\mathbf{x}_i)$ définissent une partition : $P^n = (P_1^n, ..., P_k^n)$ de l'échantillon, avec :

$$P_k^n = \{\mathbf{x}_i | e_k^n(\mathbf{x}_i) = 1\}$$

si pour un certain k, *card* $P_k < nc(n)$, l'algorithme doit être réinitialisé ;

3 : Étape Maximisation M : calculer les estimateurs du maximum de vraisemblance :

$$\begin{cases} \pi_k = \frac{1}{n} \sum_{i=1}^{n} e_{ik} \\ \mu_k = \frac{\sum_{\mathbf{x}_i \in P_k} e_{ik} \mathbf{x}_i}{\left\| \sum_{\mathbf{x}_i \in P_k} e_{ik} \mathbf{x}_i \right\|} \\ \bar{r}_k = \frac{\left\| \sum_{i=1}^{n} e_{ik} \mathbf{x}_i \right\|}{\sum_{i=1}^{n} e_{ik}} \\ \xi_k = \frac{\bar{r}_k d - \bar{r}_k^3}{1 - \bar{r}_k^2} \end{cases}$$

jusqu'à *atteindre un état de convergence fixé au départ*;

recuit simulé a été proposé dans Celeux et Diebolt (1989). On présente ici l'algorithme SAEM$_{vEM}$.

Comme pour l'algorithme SEM$_{vMF}$, on commence par fixer g, un majorant supposé du nombre de composants du mélange et un seuil $c(n, d)$, $c \in [0, 1]$. (algorithme 16).

Algorithme 16: SAEM$_{vMF}$

Entrée : $\mathbf{x}$: la matrice des données, le paramètre g majorant du nombre de composants du mélange et d'un seuil $c(n)$ compris entre 0 et 1.

Sortie : Le nombre de classes final. La partition : $P_1, ..., P_g$.

Initialisation : Choix aléatoire des paramètres du mélange $\theta_k, k = 1, \ldots, g$;

Répéter

1 : Étape Estimation E : estimation des t_{ik} $k = 1, ..., g$:

$$t_{ik} = \frac{\pi_k f(\mathbf{x}_i; \alpha_k)}{\sum_{k=1}^{g} \pi_k f(\mathbf{x}_i; \alpha_k)}$$

2 : Étape Stochastique S : On tire en chaque point $\mathbf{x}_i$ la variable aléatoire multinomiale $e^n(\mathbf{x}_i)$ d'ordre 1 et de paramètre $t_{ik}, k = 1, ..., g$, telle que :

$$e^n(\mathbf{x}_i) = (e_k^n(\mathbf{x}_i); k = 1, \ldots, g)$$

Les réalisations $e^n(\mathbf{x}_i)$ définissent une partition : $P^n = (P_1^n, ..., P_k^n)$ de l'échantillon, avec :

$$P_k^n = \{\mathbf{x}_i | e_k^n(\mathbf{x}_i) = 1\}$$

3 : Étape Hybrid (Annealing) A : $i = 1, \ldots, n$, $k = 1, \ldots, g$

$$r_{ik}^n = t_{ik} + \gamma(e_k^n - t_{ik})$$

si $card\ P_k < nc(n)$, l'algorithme doit être réinitialisé ;

4 : Étape Maximisation M : Estimation des paramètres :

$$\begin{cases} \pi_k = \frac{1}{n} \sum_{i=1}^{n} r_{ik} \\ \mu_k = \frac{\sum_{\mathbf{x}_i \in P_k} r_{ik} \mathbf{x}_i}{\left\| \sum_{\mathbf{x}_i \in P_k} r_{ik} \mathbf{x}_i \right\|} \\ \bar{r}_k = \frac{\left\| \sum_{i=1}^{n} r_{ik} \mathbf{x}_i \right\|}{\sum_{i=1}^{n} r_{ik}} \\ \xi_k = \frac{\bar{r}_k d - \bar{r}_k^3}{1 - \bar{r}_k^2} \end{cases}$$

jusqu'à *Un état fixé au départ est atteint*;

8 Simulation de mélange de lois de von Mises-Fisher

Depuis l'apparition de la distribution de von Mises, plusieurs tentatives de génération d'échantillons synthétiques à partir de cette loi sont apparues. Certains auteurs sont par-

venus à développer différentes méthodes de simulation d'échantillons suivant ce modèle, par exemple : en utilisant la loi normale enveloppée (Mardia, 1972), ou encore la distribution de Cauchy enveloppée (Best et Fisher, 1979). Aussi, une autre méthode basée sur la distribution unimodale symétrique par rotation (Ulrich, 1984).
Nous avons utilisé un algorithme développé à l'origine par Wood (Wood, 1994), cet algorithme a été conçu spécialement pour la simulation de données suivant la loi de von Mises-Fisher, ainsi que la loi de Bingham. Ce dernier a été légèrement modifié par Banerjee et al. (Banerjee *et al.*, 2005).

9 Applications numériques

Nos simulations ont été effectuées en utilisant la loi de von Mises-Fisher. Pour les tirages réalisés, on tiendra compte du concept du degré de mélange des classes, ce schéma bien particulier expliquera la notion de séparation des classes. Ceci est difficile à visualiser pour notre modèle, mais il peut être mesuré par un taux d'erreur basé sur la comparaison des classes ayant servi à la simulation à celles obtenues par une simple étape de classification.
Les données ont été simulées en tenant compte des trois degrés de mélange $\approx 5\%, 15\%, 25\%$. Pour mieux évaluer le concept du degré de mélange, nous proposons dans le paragraphe suivant (Simulation 1) des représentations graphiques, où l'on visualise 6 exemples de données simulées suivant trois degrés de mélanges différents, sur lesquels nous avons simplement appliqué les trois algorithmes SPK, EM_{vMF} et CEM_{vMF}. Ensuite, une étude basée sur des expériences numériques intensives présentée dans le paragraphe intitulé Simulation 2 afin de comparer ces trois algorithmes.

9.1 Simulation 1

Les 6 graphiques (figure (5.5), figure (5.6) et figure (5.7)) représentent 6 échantillons de taille : 1200×3 composé chacun de trois classes, selon le schéma suivant :

- proportions égales.
- proportions différentes.

Les concentrations sont identiques pour les 6 échantillons et sont respectivement égales à 100, 10 et 80 par classes. En faisant varier les centres des classes, nous avons simulé nos tableaux suivant différents degrés de mélange désirés :

- 5% : figure (5.5),
- 15% : figure (5.6) et
- 25% : figure (5.7).

FIGURE 5.5 – Echantillon 1 et $2,5\%$ de degré de mélange, proportions des classes égales et différentes

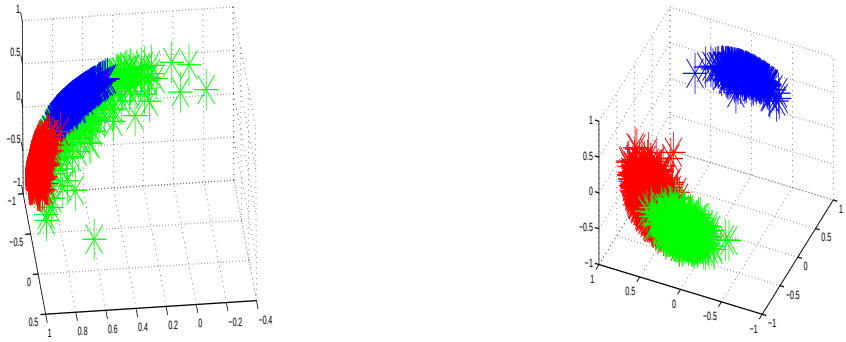

FIGURE 5.6 – Echantillon 3 et 4, 15% de degré de mélange, proportions des classes égales et différentes

FIGURE 5.7 – Echantillon 5 et 6, 25% de degré de mélange, proportions des classes égales et différentes

Nous avons appliqué sur les 6 échantillons précédents, les trois algorithmes SPK-means, EM_{vMF} et CEM_{vMF}, en les initialisant par les mêmes paramètres tirés au hasard : moyennes, proportions et concentrations du mélange. Les résultats obtenus sont résumés dans le tableau 5.1.

TABLE 5.1 – Taux de mal classés, résultats d'application de SPK, EM_{vMF} et CEM_{vMF} sur 6 échantillons

	proportions égales			proportions différentes		
degrés de mélange	5%	15%	25%	5%	15%	25%
SPK-means	10.60	25.00	30.60	11.60	29.90	39.50
CEM_{vMF}	5.50	15.83	28.58	6.00	16.08	28.25
EM_{vMF}	5.00	15.58	26.00	6.50	15.25	25.75

Comme on peut l'observer sur les différents résultats, l'algorithme EM_{vMF} est le meilleur, vient par la suite l'algorithme CEM_{vMF}, puis en dernier l'algorithme SPK-means. Par conséquent, incontestablement on note l'intérêt de l'approche modèle de mélange.

9.2 Simulation 2

Dans ce paragraphe on a procédé à des simulations de Monté Carlo, en respectant nos degrés de mélange : $\approx 5\%$, 15% et 25%, un nombre d'individus n $(60, 600, 1200)$ est simulé suivant la loi de vMF de paramètres (μ, π, ξ), en tenant compte des différentes concentrations et différentes proportions. Au total, on a construit 30 échantillons. Nous avons initialisé les trois algorithmes SPK, EM_{vMF} et CEM_{vMF} par les mêmes paramètres tirés au hasard, les taux d'éléments mal classés résultant de l'application de chacun des trois algorithmes sont reportés dans les tableaux 5.2, 5.3, 5.4 et 5.5.

TABLE 5.2 – Taux de mal classés, résultats d'application de SPK, EM_{vMF} et CEM_{vMF} sur des échantillons de proportions égales et concentrations égales

	proportions sont égales $\frac{1}{3}$ et les concentrations égales 10								
	5%			15%			25%		
size	SPK	CEM_{vMF}	EM_{vMF}	SPK	CEM_{vMF}	EM_{vMF}	SPK	CEM_{vMF}	EM_{vMF}
60	4.89	11.22	7.89	16.67	22.33	23.89	29.83	33.06	37.72
600	5.09	5.19	5.19	14.67	15.30	15.56	26.57	29.71	27.21
1200	5.23	5.24	5.24	14.92	15.29	15.41	26.18	26.57	26.69

TABLE 5.3 – Taux de mal classés, performances de SPK, EM_{vMF} et CEM_{vMF} sur des échantillons de proportions égales et concentrations différentes

	proportions égales et les concentrations différentes $[90, 5, 50]$								
	5%			15%			25%		
size	SPK	CEM_{vMF}	EM_{vMF}	SPK	CEM_{vMF}	EM_{vMF}	SPK	CEM_{vMF}	EM_{vMF}
60	14.83	7.56	7.56	26.56	19.11	19.39	51.89	40.83	40.89
600	14.06	4.86	4.86	27.01	15.44	15.44	53.93	37.81	37.83
1200	14.25	5.00	5.00	27.30	15.45	15.45	54.27	36.78	34.68

TABLE 5.4 – Taux de mal classés, performances de SPK, EM_{vMF} et CEM_{vMF} sur des échantillons de proportions différentes et concentrations égales

	proportions différentes $[0.5, 0.3, 0.2]$ et les concentrations égales 10								
	5%			15%			25%		
size	SPK	CEM_{vMF}	EM_{vMF}	SPK	CEM_{vMF}	EM_{vMF}	SPK	CEM_{vMF}	EM_{vMF}
60	6.39	7.11	7.17	16.06	20.44	23.11	33.28	39.17	38.06
600	5.76	5.45	5.45	16.57	15.32	15.29	32.44	30.08	31.19
1200	5.49	5.06	5.06	16.82	15.84	16.00	30.79	31.24	32.13

TABLE 5.5 – Taux de mal classés, performances de SPK, EM_{vMF} et CEM_{vMF} sur des échantillons de proportions différentes et concentrations différentes

	proportions différentes $[0.5, 0.3, 0.2]$ et les concentrations différentes $[90, 5, 50]$								
	5%			15%			25%		
size	SPK	CEM_{vMF}	EM_{vMF}	SPK	CEM_{vMF}	EM_{vMF}	SPK	CEM_{vMF}	EM_{vMF}
60	13.78	8.94	7.00	28.61	24.17	23.78	41.06	33.39	33.67
600	13.65	5.15	5.15	26.76	15.12	15.12	41.84	28.07	29.33
1200	13.72	5.27	5.27	26.01	15.17	15.19	41.25	27.36	29.08

9.3 Commentaires

D'après les résultats précédents, on remarque que :

1. Lorsque les proportions et les concentrations sont égales, 1^{er} groupe (tableau 5.2) :
   - SPK donne de bons résultats quand les classes sont bien séparées, mais quand elles sont assez mélangées (25%), plus n est petit, moins ses résultats sont bons.
   - EM_{vMF} s'améliore avec l'augmentation de n.
   - CEM_{vMF} est moins efficace quand le degré de mélange est important (25%).

2. Lorsque les proportions sont égales et les concentrations sont différentes, 2$^{\text{ème}}$ groupe (tableau 5.3) :
   - SPK ne donne pas toujours de bons résultats.
   - EM$_{vMF}$ est toujours meilleur et s'améliore avec l'augmentation de n, ses résultats sont moins bons quand le degré de mélange est grand (25%).
   - CEM$_{vMF}$ est toujours meilleur que SPK, mais moins efficace que EM$_{vMF}$.
3. Lorsque les proportions sont différentes et les concentrations sont égales, 3$^{\text{ème}}$ groupe (tableau 5.4) :
   - SPK est moins efficace pour des classes très mélangées (25%).
   - EM$_{vMF}$ est toujours meilleur surtout avec n grand, pour des échantillons assez mélangés (25%) il est moins bon.
   - CEM$_{vMF}$ moins bon que EM$_{vMF}$ et aussi moins bon que SPK quand le degré de mélange est important (25%).
4. Lorsque les proportions et les concentrations sont différentes, 4$^{\text{ème}}$ groupe (tableau 5.5) :
   - SPK est toujours moins efficace, mais stable.
   - EM$_{vMF}$ est toujours meilleur et s'améliore avec l'augmentation de n, quand les classes sont moins séparées (25% de degré de mélange) il est moins efficace, mais reste le meilleur.
   - CEM$_{vMF}$ est moins bon que EM$_{vMF}$, la plus part du temps meilleur que SPK.

10 Conclusion

D'après les résultats expérimentaux on conclut que :

1. SPK dépend des concentrations, plus les valeurs des concentrations des classes se rapprochent, plus ses résultats sont bons, il donne de mauvais résultats pour des concentrations très différentes.
2. EM$_{vMF}$ a toujours le même comportement, il est meilleur et s'améliore avec l'augmentation de n, sauf dans le cas où les concentrations sont très différentes. Dans ce cas EM$_{vMF}$ reste meilleur ; même pour des degrés de mélange élevés (25%).
3. CEM$_{vMF}$ est toujours moins efficace que EM$_{vMF}$, mais il a le même comportement, il s'améliore avec l'augmentation de n, il est moins efficace quand les classes sont moins séparées.

Chapitre 6

Critères d'information et modèle de mélange de vMF

Le problème de sélection du nombre de composants dans un mélange de lois de probabilités est un problème bien connu en statistique. Les connaissances a priori sur les données ne permettent pas de déterminer une solution idéale. La pénalisation d'un critère est l'une des méthodes apportées pour la détection du nombre de composants dans un mélange.

Les premiers critères dits d'information apparus dans la littérature sont l'Akaike (Aic, (Akaike, 1973)), le critère d'information Bayésien (Bic, (Schwarz, 1978)). Ces deux critères ont été largement diffusés et appliqués. Théoriquement beaucoup de travaux ont été réalisés à propos de leurs propriétés statistiques et de leur adaptation à des modèles spécifiques. Plus tard, plusieurs versions corrigées du critère Aic ont été proposées, tels que : le Aicc (Hurvich et Tsai, 1989) et le CAic (Sugiura, 1978) pour les échantillons de petites tailles par rapport au nombre de paramètres à estimer. Le critère Aicr (Ronchetti, 1985) est utilisé si on a une régression avec erreurs non-Gaussiennes ; les critères QAic (Burnham et Anderson, 2002) et CQAic (Shi et Tsai, 1998) sont choisis si les données sont sur-dispersées.

Dans ce chapitre, nous proposons d'étudier le comportement des critères d'information suivants : Bic, Aic, Aic3, Aic4, Aicc, Aicu, Caic, Clc, Icl-Bic, Ll, Icl, Awe. en utilisant les distributions de von Mises-Fisher et dans un contexte de données directionnelles (textuelles).

A cette fin, ce chapitre est organisé comme suit : dans un premier lieu, nous présentons les différentes méthodes et critères servant à la discussion des problèmes (choix du modèle et estimation du nombre de classes détectées dans un mélange de lois de probabilités). Dans un second lieu, on présente les applications numériques faites en considérant des simulations de Monte Carlo. Les résultats obtenus nous ont permis de distinguer les critères les plus adaptés pour ce type de modèle de mélange.

1 Choix du nombre de classes

Le critère optimisé n'est pas indépendant du nombre de classes. Donc le problème de classification peut être ramené à la recherche de la meilleure partition parmi plusieurs partitions avec différents nombres de classes. Ce problème étant jugé difficile plusieurs solutions sont proposées pour le résoudre

- Rechercher la meilleure partition pour plusieurs nombres de classes dans le but de sélectionner le meilleur nombre en utilisant la méthode du coude. Dans ce cas, nous étudions la décroissance du critère en fonction du nombre de classes.
- Ajouter des contraintes supplémentaires (méthode Isodata), par exemple le nombre d'individus par classe ou bien le volume de la classe etc.
- Utiliser des approches de type inférentielle basées sur des tests statistiques.
- Pénaliser la vraisemblance ou la vraisemblance classifiante par le nombre de paramètres. Plusieurs méthodes décrites pour déterminer le nombre de classes dans un mélange sont basées sur cette approche. De nombreux travaux ont été réalisés : le critère d'information Akaike (AIC) proposé par Akaike (Akaike, 1973) ; le AIC3 (AIC modifié) proposé par Bozdogan, (Bozdogan, 1994) ; le critère AWE (approximate weight of evidence) proposé par Banfield et Raftery (Banfield et Raftery, 1993). D'autres critères traitant le problème de l'évaluation du nombre de classes sont proposés dans un cadre bayésien : BIC, CS, ICL, ICL-BIC (voir par exemple, (Jollois, 2003)).
- Combiner des méthodes de partitionnement avec celles de la classification hiérarchique nous permet de déterminer un nombre de classes judicieux ainsi qu'une initialisation intéressante pour l'algorithme de classification, rappelons ici la méthode de Wong (Wong, 1982). D'autre part, il y a des stratégies qui associent la méthode EM à CAH, ou encore avec la méthode CEM afin d'estimer le nombre de classes (Jollois, 2003).

2 Choix du modèle

Due à la liaison entre la classification automatique et les modèles probabilistes, la recherche d'une meilleure partition est étroitement liée au choix du modèle qui convient au mieux à la structure des données. Dans certaines situations, où le concept de composante a une signification physique tout à fait précise, le nombre de composantes peut parfaitement être déterminé ; mais généralement, ce nombre est inconnu et doit être estimé. Ce problème est placé dans le cadre le plus général du choix de modèles probabilistes.

L'utilisation du simple critère de vraisemblance pour la sélection du nombre de compo-

santes dans le mélange conduit généralement à sélectionner le plus grand nombre de classes, car la vraisemblance a tendance à croître avec le nombre de composants du modèle. Les critères d'information sont utilisés en raison de leur simplicité de mise en œuvre, le principe d'un critère d'information consiste à choisir le modèle qui fait croître la vraisemblance et minimise la complexité du modèle à la fois.

Si M dénote un modèle, $L_{\max}(M)$ le maximum de log-vraisemblance avec ce modèle et $v_{d,g}(M)$ le nombre de paramètres libres de celui-ci ; la forme générale du critère à minimiser sur différents modèles est le suivant :

$$Crit(M) = -2L_{\max}(M) + \eta v_{d,g}(M)$$

η est le coefficient de pénalisation par la complexité du modèle et spécifique au critère choisi, dans le critère AIC (Akaike (Akaike, 1973)), $\eta = 2$ et le critère s'écrit :

$$AIC(M) = -2L_{\max}(M) + 2v_{d,g}(M).$$

Une autre variante AIC3 proposée par (Bozdogan, 1994) :

$$AIC3(M) = -2L_{\max}(M) + 3v_{d,g}(M).$$

Dans l'approche classification, le critère ICL (Biernacki, 1997), peut être mieux adapté pour choisir à la fois un modèle M et un nombre de classes g approprié. Il s'écrit :

$$ICL(M) = -2L_{C\max}(M) + \log(n)v_{d,g}(M).$$

avec $L_{C\max}(M)$ est le maximum de log-vraisemblance complétée avec M.

3 Modèles de mélange de von Mises-Fisher (vMF)

Nous reprenons dans cette partie, la loi de von Mises-Fisher telle qu'elle a été présentée au chapitre précédent. Le tableau de données est supposé provenir d'un mélange de g composants de lois de von Mises-Fisher, chaque composant représente une classe distincte. Nous rappelons ici la fonction de densité de probabilité du mélange :

$$f(\mathbf{x}_i;\theta) = \sum_{k=1}^{g} \pi_k f_k(\mathbf{x}_i;\mu_k,\xi_k) \tag{6.1}$$

où $f_k(\mathbf{x}_i;\mu_k,\xi_k)$ est la fonction de distribution de vMF, θ est le vecteur des paramètres inconnus $(\pi_1,\ldots,\pi_g;\mu_1,\ldots,\mu_g;\xi_1,\ldots,\xi_g)$.

Pour détecter les propres classes du mélange, l'optimisation du critère de la log-vraisemblance est nécessaire.

A une étape (c), le critère équivalent à maximiser est :

$$Q\left(\theta, \theta^{(c)}\right) = \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log \pi_k + \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log f\left(\mathbf{x}; \mu_k, \xi_k\right)$$

Nous allons considérer quatre modèles de mélange de distributions de von Mises-Fisher, en imposant différentes contraintes aux proportions π_k et aux concentrations ξ_k .

– Le modèle $[\pi, \xi]$ aux proportions et concentrations égales, le critère à maximiser est :

$$\sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log f\left(\mathbf{x}; \mu_k, \xi\right).$$

– Le modèle $[\pi_k, \xi]$ à proportions différentes et concentrations égales, le critère à maximiser est :

$$\sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log \pi_k + \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log f\left(\mathbf{x}; \mu_k, \xi\right).$$

– Le modèle $[\pi, \xi_k]$ à proportions égales et concentrations différentes, le critère à maximiser est :

$$\sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log f\left(\mathbf{x}; \mu_k, \xi_k\right).$$

– Et enfin le modèle général $[\pi_k, \xi_k]$ à proportions et concentrations différentes, le critère à maximiser est :

$$\sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log \pi_k + \sum_{k=1}^{g}\sum_{i=1}^{n} t_{ik}^{(c)} \log f\left(\mathbf{x}; \mu_k, \xi_k\right).$$

4 Sélection du nombre de classes

La sélection du nombre de composants g peut être vue à travers le problème de sélection de modèle, ce type de problème admet plusieurs méthodes de résolution, parmi lesquelles : les critère d'information, les méthodes basées sur l'intervalle de confiance et d'autres méthodes empiriques (Bubna et Stewart, 2000). Les techniques les plus populaires sont les critères d'information. Nous proposons dans la suite de ce travail de tester quelques-uns. Ces critères consistent à pénaliser le modèle avec d'autres paramètres, ils se composent de deux termes : l'un correspond au modèle (vraisemblance où vraisemblance complétée), l'autre représente la complexité du modèle. Il existe des méthodes destinées uniquement au problème d'estimation du nombre de composants dans un objectif de classification, celles-ci conduisent à exclure les classes vides ou peu condensées. En classification hiérarchique, une agglomération successive des classes est fusionnée pour composer une seule classe à un certain niveau de dissimilarité fixé (Dumais et Chen, 2000). D'autre part, comme nous l'avons vue précédemment, l'algorithme SEM, version stochastique de EM, estime particulièrement g. Itérativement cette méthode rejette les classes dont le nombre d'éléments est inférieur à un seuil fixé.

5 Critères d'informations

On désigne par L la log-vraisemblance des données observées, L_c la log-vraisemblance complétée et $\hat{\theta}$ le paramètre estimé par l'algorithme EM, υ le nombre de paramètres libres dans le mélange et $E = \sum_{i,k} t_{ik} \log(t_{ik})$ le critère d'entropie. Les termes L, L_c, υ et E dépendent de g. Nous proposons dans la suite de ce travail, de tester les 12 critères suivants :

- $Bic(g) = -2L(g) + \upsilon \log n$, proposé par Schwarz (Schwarz, 1978) et Rissanen (Rissanen, 1978)
- $Aic(g) = -2L(g) + 2\upsilon$, proposé par Akaike (Akaike, 1973)
- $Aic3(g) = -2L(g) + 3\upsilon$, proposé par Bozdogan (Bozdogan, 1994)
- $Aic4(g) = -2L(g) + 4\upsilon$, proposé par Bozdogan (Bozdogan, 1994)
- $Aicc(g) = Aic(g) + \frac{2\upsilon(\upsilon+1)}{n-\upsilon-1}$, proposé par Hurvich and Tsai (Hurvich et Tsai, 1989)
- $Aicu(g) = Aicc(g) + n \log \frac{n}{n-\upsilon-1}$, proposé par McQuarrie, Schwarz et Tsai (McQuarrie *et al.*, 1997)
- $CAic(g) = -2L(g) + \upsilon(1 + \log n)$, proposé par Bozdogan (Bozdogan, 1987)
- $Clc(g) = -2L(g) + 2E(g)$, proposé par Biernacki (Biernacki, 1997)
- $IclBic(g) = Bic(g) + 2E(g)$, proposé par Biernacki, Celeux and Govaert (Biernacki *et al.*, 2000)
- $Ll(g) = -L(g) + \frac{\upsilon}{2} \sum_k \log \frac{n\pi_k}{2} + \frac{g}{2} log \frac{n}{12} + \frac{g(\upsilon+1)}{2}$, proposé par Figueiredo et Jain (Figueiredo et Jain, 2002)
- $Icl(g) = -2L_c(g) + \upsilon \log n$, proposé par Biernacki, Celeux et Govaert (Biernacki *et al.*, 2000)
- $Awe(g) = -2L_c(g) + 2\upsilon(\frac{3}{2} + \log n)$, proposé par Banfield et Raftery (Banfield et Raftery, 1993)

6 Étude expérimentale

Avant de commencer l'évaluation des critères précédemment cités sur les modèles de mélange de lois de von Mises-Fisher, nous rappelons que nous utilisons les mêmes conditions expérimentales faites au chapitre précédent, dans lequel nous avons effectué une étude en tenant compte des degrés de mélange des classes. Différentes tailles de données ont été sélectionnées : 600×3, 1800×3, 6000×3, 6000×50 et 6000×50 avec différents degrés de mélanges $\approx (5\%, 15\%, 25\%)$. Chaque tableau de données est composé de 3 sous échantillons.

Avant toute application, nous remarquons que la complexité du modèle dans les différents critères exposés auparavant est due au nombre de paramètres libres dans le

mélange, par exemple pour un modèle de mélange de von Mises-Fisher ayant les paramètres inconnus (μ_k, π_k, ξ_k) avec : $\upsilon = g(d+2) - 1$, alors si $g = 2, \cdots, 5$, un calcul rapide nous donne une idée sur cette quantité et sur le $log(n)$, dans la table suivante :

g/d	3	50	100	n	$log(n)$
2	9	103	203	600	6.3969
3	14	155	305	1800	7.4955
4	19	207	407	3000	8.0063
5	24	259	509	6000	8.6995

TABLE 6.1 – Nombre de paramètres libres pour g classes.

En outre, on calcule le terme de pénalité pour tous les critères, ce terme est indépendant des itérations de l'algorithme EM (table 6.2). Il est clair que ce terme dépend de g, n et d.

Les valeurs des critères croissent uniformément, en fonction de l'accroissement des valeurs de g. Les critères Aic, $Aic3$, $Aic4$ et $Aicc$ sont indépendants du nombre de lignes n ; pour ces critères le terme de pénalité croît d'une manière modeste quand le nombre de colonnes d augmente. Les termes de pénalité des critères Bic, $Icl - Bic$, Icl et $Caic$ dépendent des deux paramètres n et d, la croissance de ces critères dépend elle aussi des mêmes paramètres, enfin la pénalité de Awe croît rapidement.

Pour évaluer l'algorithme EM et les critères précédents, nous avons réalisé plusieurs applications sur des données simulées. Pour tout θ conduisant à un degré de mélange, nous avons généré 20 échantillons. Pour chaque échantillon, et pour éviter les optimums locaux dans ce processus, l'algorithme $EM(g)$ $(g = 2, \ldots, 5)$ appliqué au modèle général $[\pi_k, \xi_k]$, est répété 20 fois en démarrant de la meilleure partition obtenue par le $k$means sphérique (Banerjee *et al.*, 2005) qui n'est autre que le CEM appliqué avec le modèle $[\pi, \xi]$.

1. on calcule le pourcentage des documents mal classés en comparant avec la vraie partition pour le vrai nombre de classes,
2. on calcule tous les critères cités auparavant en fonction des différentes valeurs de g,
3. On compte le nombre de fois sur 20, où chaque critère estime le nombre de classes originales *fit*, le surestime *over-fit* ou le sous-estime *under-fit*. Les résultats obtenus sont reportés dans les trois tableaux 6.3, 6.4 et 6.5.

Les principaux résultats sont :

– L'algorithme EM donne de bons résultats en comparant la vraie partition à la résultante de EM avec $g = 3$, EM(3).
– Quand les classes sont bien, ou moins séparées les critères Aic3, Aic4, Aicu et Bic sont plus efficaces pour les tailles étudiées.

TABLE 6.2 – Valeurs du terme de pénalité pour quelques critères d'information.

Size	*Clusters*	*Bic* $\frac{Icl-Bic}{Icl}$	*Aic*	*Aic3*	*Aic4*	*Aicc*	*Aicu*	*Caic*	*Awe*
600×3	2	57.572	18	27	36	18.033	28.118	66.572	142.144
	3	89.557	28	42	56	28.051	43.241	103.557	221.114
	4	121.541	38	57	76	38.068	58.409	140.541	300.083
	5	153.526	48	72	96	48.086	73.622	177.526	379.052
1800×3	2	67.459	18	27	36	18.011	28.039	76.459	161.919
	3	104.937	28	42	56	28.016	43.079	118.937	251.875
	4	142.415	38	57	76	38.022	58.134	161.415	341.830
	5	179.893	48	72	96	48.028	73.203	203.893	431.786
6000×3	2	78.295	18	27	36	18.003	28.011	87.295	183.591
	3	121.793	28	42	56	28.005	43.023	135.793	285.586
	4	165.290	38	57	76	38.006	58.040	184.290	387.581
	5	208.788	48	72	96	48.008	73.060	232.788	489.576
3000×50	2	824.655	206	309	412	206.071	311.917	927.655	1958.311
	3	1240.986	310	465	620	310.109	470.312	1395.986	2946.973
	4	1657.318	414	621	828	414.148	629.711	1864.318	3935.636
	5	2073.649	518	777	1036	518.189	790.152	2332.649	4924.298
6000×50	2	896.050	206	309	412	206.035	310.947	999.050	2101.100
	3	1348.424	310	465	620	310.053	468.117	1503.424	3161.849
	4	1800.799	414	621	828	414.071	625.762	2007.799	4222.599
	5	2253.174	518	777	1036	518.090	783.892	2512.174	5283.348
6000×100	2	1766.001	406	609	812	406.070	613.619	1969.001	4141.002
	3	2653.351	610	915	1220	610.107	924.186	2958.351	6221.703
	4	3540.702	814	1221	1628	814.145	1236.680	3947.702	8302.405
	5	4428.053	1018	1527	2036	1018.185	1551.173	4937.053	10383.106

– Quand les classes sont mal séparées, la qualité des critères s'améliore avec la taille n et quand $n >> d$.
– Par ailleurs, on note que les critères Aic3 et Aicu sont plus efficaces que le critère Bic. Dans la plupart des situations, ils restent plus intéressants, même quand le nombre de colonnes augmente. En fait, le critère Bic semble très sensible à la grande dimension, il sous-estime le nombre de classes.

Dand les tableaux 6.3, 6.4 et 6.5, on note par $IB = Icl - Bic$ et par : (1)=*under-fit*, (2)=*fit* et (3)=*over-fit*. Dans ces premières expériences, nous pouvons dire que Aic3 et Aicu sont les meilleurs critères. Le critère Aic3 est aussi intéressant pour les modèles de mélange de Bernoulli pour les données binaires (Nadif et Govaert, 1998). Cependant, nous avons noté que leur performance diminue quand on a une grande dimension. Nous

illustrons le comportement de tous les critères en utilisant un groupe de données bien connues Classic3 comme application sur données réelles.

La collection Classic [1] est un ensemble de données de référence bien connue utilisée dans l'exploration de texte. Cet ensemble de données est composé de 4 collections de documents différents :

- CACM : 3204 documents
- CISI : 1460 documents
- CRAN : 1398 documents
- MED : 1033 documents

Cet ensemble de données est appelé parfois Classic4, généralement dénommé Classic3 quand il ne s'agit que des trois groupes : CISI, CRAN et MED. Chaque vecteur a été normalisé afin d'être utilisé comme un vecteur unitaire. Dans le but de sélectionner un certain nombre de classes $g = 2, \ldots, 5$; nous avons utilisé les mêmes critères que précédemment et nous avons appliqué l'algorithme EM(g) pour le modèle général $[\pi_k, \xi_k]$. Les résultats obtenus sont les suivants :

- Les critères Bic, Caic, Icl-Bic, Icl ont surestimé le nombre de classes et ont donné 4 classes.
- Les critères Aic, Aic3, Aic4, Aicc, Clc ont surestimé le nombre de classes et ont donné 5 classes.
- Les critères Aicu, Ll, Awe ont sous-estimé le nombre de classes et ont donné 2 classes.

7 Conclusion

Dans le contexte, approche mélange pour la classification de données directionnelles, nous avons effectué quelques expériences, afin d'évaluer la performance de l'algorithme EM et d'estimer le nombre de classes. Différents critères d'information ont été testés sur différents formats de données selon différents degrés de mélange. Nous avons observé que certains d'entre eux tels que les critères Aic3, Aic, AICU et Bic ont été les plus intéressants. En outre, nous avons constaté que leur performance s'améliore avec l'augmentation de la taille des données et les critères Aic3 et AICU apparaissent comme les meilleurs.

1. ftp ://ftp.cs.cornell.edu/pub/smart/

TABLE 6.3 – Évaluation de EM et des critères d'information pour le modèle $[\pi_k, \xi_k]$. Pour chaque critère, le nombre de fois sur 20 où le critère détecte ou ne détecte pas le vrai nombre de classes (a).

size	*degree*	$EM(3)$	*fit*	*Bic*	*Aic*	*Aic3*	*Aic4*	*Aicc*	*Aicu*	*CAic*	*Clc*	*IB*	*Ll*	*Icl*	*Awe*
600×3	4.88%	5.17%	(1)	0	0	0	0	0	0	0	0	0	0	0	0
			(2)	20	15	19	20	15	19	20	15	20	20	20	20
			(3)	0	5	1	0	5	1	0	5	0	0	0	0
1800×3	5.16%	4.83%	(1)	0	0	0	0	0	0	0	0	0	0	0	0
			(2)	20	18	19	20	18	19	20	20	20	20	20	20
			(3)	0	2	1	0	2	1	0	0	0	0	0	0
3000×50	4.74%	6.85%	(1)	0	0	0	0	0	0	0	0	1	7	0	4
			(2)	20	1	20	20	1	20	20	6	19	13	20	16
			(3)	0	19	0	0	19	0	0	14	0	0	0	0
3000×400	5.75%	12.27%	(1)	20	0	0	20	0	4	20	0	20	20	20	20
			(2)	0	20	20	0	20	16	0	0	0	0	0	0
			(3)	0	0	0	0	0	0	0	20	0	0	0	0
6000×200	5.06%	7.20%	(1)	0	0	0	0	0	0	0	0	0	12	0	0
			(2)	20	0	20	20	0	20	20	0	20	8	20	20
			(3)	0	20	0	0	20	0	0	20	0	0	0	0
6000×300	4.72%	6.86%	(1)	0	0	0	0	0	0	0	0	0	20	0	20
			(2)	20	0	20	20	0	20	20	0	20	0	20	0
			(3)	0	20	0	0	20	0	0	20	0	0	0	0
6000×400	6.97%	10.56%	(1)	20	0	0	0	0	0	20	0	20	20	20	20
			(2)	0	0	20	20	0	20	0	0	0	0	0	0
			(3)	0	20	0	0	20	0	0	20	0	0	0	0

TABLE 6.4 – Évaluation de EM et des critères d'information pour le modèle $[\pi_k, \xi_k]$. Pour chaque critère, le nombre de fois sur 20 où le critère détecte ou ne détecte pas le vrai nombre de classes (b).

size	*degree*	$EM(3)$	*fit*	*Bic*	*Aic*	*Aic3*	*Aic4*	*Aicc*	*Aicu*	*CAic*	*Clc*	*IB*	*Ll*	*Icl*	*Awe*
600×3	14.63%	16.33%	(1)	0	0	0	0	0	0	0	9	16	7	7	16
			(2)	20	17	20	20	17	20	20	9	4	13	13	4
			(3)	0	3	0	0	3	0	0	2	0	0	0	0
1800×3	15.10%	15.83%	(1)	0	0	0	0	0	0	0	14	18	0	2	7
			(2)	20	19	20	20	19	20	20	6	2	20	18	13
			(3)	0	1	0	0	1	0	0	0	0	0	0	0
3000×50	13.68%	14.10%	(1)	0	0	0	0	0	0	0	0	3	0	0	20
			(2)	20	10	20	20	10	20	20	18	17	20	20	0
			(3)	0	10	0	0	10	0	0	2	0	0	0	0
3000×100	15.84%	19.36%	(1)	20	0	0	0	0	0	20	20	20	20	20	20
			(2)	0	0	20	20	0	20	0	0	0	0	0	0
			(3)	0	20	0	0	20	0	0	0	0	0	0	0
3000×200	15.92%	19.87%	(1)	20	0	0	20	0	0	20	8	20	20	20	20
			(2)	0	0	20	0	0	20	0	0	0	0	0	0
			(3)	0	20	0	0	20	0	0	12	0	0	0	0
3000×300	15.41%	25.67%	(1)	20	4	20	20	4	20	20	0	20	20	20	20
			(2)	0	8	0	0	8	0	0	0	0	0	0	0
			(3)	0	8	0	0	8	0	0	20	0	0	0	0
6000×100	15.11%	18.35%	(1)	0	0	0	0	0	0	4	20	20	20	20	20
			(2)	20	0	20	20	0	20	16	0	0	0	0	0
			(3)	0	20	0	0	20	0	0	0	0	0	0	0
6000×200	16.80%	23.28%	(1)	20	0	20	20	0	20	20	0	20	20	20	20
			(2)	0	0	0	0	0	0	0	0	0	0	0	0
			(3)	0	20	0	0	20	0	0	20	0	0	0	0
6000×300	15.04%	22.84%	(1)	20	0	20	20	0	20	20	0	20	20	20	20
			(2)	0	0	0	0	0	0	0	0	0	0	0	0
			(3)	0	20	0	0	20	0	0	20	0	0	0	0

TABLE 6.5 – Évaluation de EM et des critères d'information pour le modèle $[\pi_k, \xi_k]$. Pour chaque critère, le nombre de fois sur 20 où le critère détecte ou ne détecte pas le vrai nombre de classes (c).

size	*degree*	$EM(3)$	*fit*	*Bic*	*Aic*	*Aic*3	*Aic*4	*Aicc*	*Aicu*	*CAic*	*Clc*	*IB*	*Ll*	*Icl*	*Awe*
600×3	24.96%	29.17%	(1)	20	15	17	20	15	18	20	20	20	20	20	20
			(2)	0	3	3	0	3	2	0	0	0	0	0	0
			(3)	0	2	0	0	2	0	0	0	0	0	0	0
1800×3	25.19%	35.94%	(1)	20	12	17	19	12	17	20	20	20	20	20	20
			(2)	0	8	3	1	8	3	0	0	0	0	0	0
			(3)	0	0	0	0	0	0	0	0	0	0	0	0
6000×3	27.49%	30.95%	(1)	0	0	0	0	0	0	0	20	20	8	20	20
			(2)	20	20	20	20	20	20	20	0	0	12	0	0
			(3)	0	0	0	0	0	0	0	0	0	0	0	0
3000×50	24.75%	32.26%	(1)	18	0	0	0	0	0	20	20	20	20	20	20
			(2)	2	8	20	20	8	20	0	0	0	0	0	0
			(3)	0	12	0	0	12	0	0	0	0	0	0	0
6000×50	25.61%	30.17%	(1)	20	0	1	16	0	1	20	20	20	20	20	20
			(2)	0	11	19	4	11	19	0	0	0	0	0	0
			(3)	0	9	0	0	9	0	0	0	0	0	0	0
6000×100	26.74%	42.91%	(1)	20	8	20	20	8	20	20	19	20	20	20	20
			(2)	0	3	0	0	3	0	0	1	0	0	0	0
			(3)	0	9	0	0	9	0	0	0	0	0	0	0

Chapitre 7

Classification sur une hypersphère de rayon supérieur à un

Le problème de la classification est de plus en plus difficile quand la taille de l'échantillon traité est très petite par rapport à sa dimension ($n << d$).

Initialement, dans le cas de données directionnelles, ce problème est bien posé ; pour ce type de données les méthodes de classification les plus connues sont basées sur la normalisation des données, elles consistent à placer les données sur une hypersphère de la même dimension et de rayon un. Cependant, de nombreuses études récentes ont mis en évidence la classification des données directionnelles sans normalisation ((Lagona et Picone, 2012a), (Lagona et Picone, 2012b), (Mardia *et al.*, 2007), (Mardia *et al.*, 2008)). Dans ce chapitre, nous allons justifier l'importance de la mise en œuvre des formes sphériques de rayon supérieur à un ; en soulignant le problème principal du calcul d'intégrales sur une hypersphère. Ces calculs dépendent toujours et fortement de la dimension des données traitées.

Les approches mentionnées dans les chapitres précédents, ont été explorées dans des situations différentes et pour différents types de données. Pour de nombreux auteurs, les algorithmes convergent vers des optimums locaux en fonction de l'initialisation. Toutefois, cela n'a pas empêché le SPK-means, par exemple, d'être d'une grande utilité : de nombreux modèles basés sur la notion de mélange sont juste initialisés avec les classes qui en découlent ; l'algorithme EM appliqué sur le mélange de distributions de von Mises-Fisher (vMF) (Banerjee *et al.*, 2005), en fait un exemple idéal. Il est à noter une certaine sensibilité envers l'initialisation, et aussi envers la taille des données, poussant quelques auteurs à chercher une amélioration plus complexe, quelques fois avec plus de paramètres à estimer (Dortet-Bernadet et Wicker, 2007), ou bien moins de paramètres à estimer (Phuong et Vinh, 2008).

Dans (Dortet-Bernadet et Wicker, 2007), les auteurs ont considéré un type plus général d'une distribution associée à des formes et des orientations différentes, approuvant une description plus détaillée et plus souple des classes, mais cette méthode n'est pas pratique quand $n << d$.

Pour répondre à ce dernier problème, un modèle plus simple a été proposé dans (Phuong et Vinh, 2008), les données sont censées être générées suivant un mélange de g distributions exponentielles. Les auteurs ont proposé de normaliser les données en utilisant un rayon prédéfini $\rho > 0$. Ils ont employé un algorithme EM normalisé (NEM) pour estimer les paramètres du mélange. L'algorithme NEM est stable pour les dimensions élevées, mais l'utilisateur doit balayer un grand intervalle de valeurs du rayon pour déterminer une valeur appropriée. Notons que cette valeur appropriée est définie à la suite d'une comparaison de la partition résultante avec la vraie partition que l'utilisateur n'est pas sensé connaître.

Pour cette raison, nous pensons qu'il serait intéressant dans ce chapitre de proposer une nouvelle méthode qui tiendra compte de l'estimation du rayon ρ. On montre tout d'abord que l'algorithme $k$means sphérique ne dépend pas du rayon de l'hypersphère, ensuite, on précisera les majeurs difficultés du calcul d'intégrale sur une hypersphère S_ρ^{d-1}. La section suivante sera consacrée aux modèles de mélange exponentiels qui sont parcimonieux, tout d'abord avec un rayon connu ensuite avec un rayon inconnu. Nous allons alors proposé une méthode d'estimation, basée sur la maximisation de la log-vraisemblance, que nous appellerons NEM$_\rho$. Nous étudions les liens possibles avec le modèle vMF et proposons quelques variantes de cette dernière. Enfin, nous terminerons ce chapitre par une évaluation de l'algorithme NEM$_\rho$ et de ses variantes sur des données génétiques et des données simulées.

1 Un algorithme de type $k$means sphérique : SPK$_\rho$

On reprend dans ce paragraphe l'algorithme SPK définit au chapitre quatre, mais nous l'utiliserons ici sur une hypersphère de rayon fixé ($\rho \geq 1$), alors le produit scalaire entre un élément $\mathbf{x}$ et une moyenne directionnelle μ est défini par :

$$\langle \mathbf{x}, \mu \rangle = \|\mu\| \, \|\mathbf{x}\| \cos(\mathbf{x}, \mu) = \rho^2 \cos(\mathbf{x}, \mu)$$

Comme ρ est fixé, maximiser la somme des produits est équivalent à maximiser la somme des cosinus de similarités sur l'hypersphère unitaire (critère 4.2), ce qui conduit au même critère optimisé par SPK et par la suite à la même partition quelle que soit la valeur de ρ .

Dans ce cas, l'algorithme 17 associé au $k$means sphérique génère les mêmes itérations

que sur une hypersphère unitaire.

Algorithme 17: L'algorithme SPK_ρ

Entrée : g : le nombre de classes ;

$\mathbf{x}$: la matrice des données.

Initialisation : Choisir de façon arbitraire g vecteurs μ_k caractérisant une partition de g classes ;

Sortie : Les classes : $P_1, ..., P_g$.

Répéter

1 : Pour $i = 1, \ldots, n$, $\mathbf{x}_i$ est affecté à la $k^{\text{iéme}}$ classe, avec

$$k = \arg \max_{k'} \left(\rho^2 \ \mu_{k'}^t \mathbf{x}_i\right), k' = 1, \ldots, g$$

2 : Calculer $\mu_k = \rho \frac{\sum\limits_{\mathbf{x}_i \in P_k} \mathbf{x}_i}{\left\| \sum\limits_{\mathbf{x}_i \in P_k} x_i \right\|}$.

jusqu'à *la convergence*;

2 Modèle de mélange exponentiel parcimonieux

Pour le modèle de mélange exponentiel parcimonieux (Phuong et Vinh, 2008), les auteurs ont proposé de normaliser les données en utilisant un rayon prédéfini $\rho > 0$, les données sont tirées d'un mélange de g distributions exponentielles de la forme :

$$\varphi_k(\mathbf{x}_i; \alpha_k) = \gamma_\rho \exp\left(- \left\|\mathbf{x}_i - \mu_k\right\|^2\right), \tag{7.1}$$

où

$$\|\mathbf{x}_i\| = \rho; i = 1 \ldots, n \text{ et } \|\mu_k\| = \rho; k = 1 \ldots, g$$

et

$$\gamma_\rho = \frac{1}{\int_{\mathbf{x}_i \in S_\rho} \exp\left(- \left\|\mathbf{x}_i - \mu_k\right\|^2\right)}.$$

2.1 L'algorithme EM normalisé (NEM)

Comme pour n'importe quel algorithme de type EM, à une itération (c), l'algorithme NEM utilise les deux étapes E et M. Dans l'étape E, il calcule les valeurs t_{ik} et dans l'étape M, il estime les paramètres du mélange de la même manière que pour les mélanges de lois de vMF, on obtient alors :

$$\pi_k^{(c)} = \frac{\sum_{i=1}^n t_{ik}^{(c)}}{n}$$

et

$$\mu_k^{(c)} = \rho \frac{\sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i}{\left\| \sum_{i=1}^{n} t_{ik}^{(c)} \mathbf{x}_i \right\|}$$

Dans (Phuong et Vinh, 2008), Phuong et Vinh ont fixé la valeur de ρ^2 de l'hypersphère dans un intervalle de valeurs allant de 1 à 350. Pour chaque valeur de ρ, l'algorithme NEM est exécuté jusqu' à la convergence. En ce qui concerne les données de grande dimensionnalité, cet algorithme serait plus efficace si l'utilisateur avait une valeur appropriée du rayon ρ.

A l'aide d'un exemple de tableau de données simulées, on montrera dans le paragraphe ci-dessous l'effet de la connaissance de la valeur du rayon sur le résultat de la classification par l'algorithme NEM.

2.2 Effet de la connaissance de la valeur du rayon ρ sur la classification

Lorsque les auteurs ont testé l'algorithme normalisé EM avec de nombreuses valeurs de ρ^2, sur un grand intervalle qui est $[1, 350]$, certaines valeurs donnent des partitions comparables à celle de l'origine, où la qualité de la partition estimée dépend fortement de la valeur choisie du rayon ρ.

Pour illustrer ce résultat, nous avons simulé un mélange de trois échantillons de données de 2 dimensions, suivant une distribution exponentielle ; soit une distribution de von Mises-Fisher de concentrations égales.

$$\pi = \begin{bmatrix} 60 \\ 36 \\ 24 \end{bmatrix} \text{ and } \mu = \begin{bmatrix} 0.928 & -0.371 \\ -0.573 & 0.819 \\ -0.037 & -0.999 \end{bmatrix}.$$

Nous avons alors appliqué l'algorithme normalisé EM avec les rayon $\rho \in \{1, \sqrt{2}, 2\}$ de suite, initialisé avec les mêmes centres, les données ont été normalisées chaque fois avec un rayon approprié ; les résultats sont résumés dans le tableau 7.1.

On a visualisé sur la figure 7.1 le même groupe de données, représenté sur des cercles de rayon $\rho = 1, \sqrt{2}, 2$ de suite. Les moyennes directionnelles résultants des applications sont représentées par des cercles noirs. Comme nous pouvons l'observer, les trois groupes de données sont bien séparées sur le cercle de rayon $\rho = \sqrt{2}$, mais sur le cercle de rayon $\rho = 1$, l'algorithme NEM a fusionné deux groupes de données pour faire une seule classe ; une seule moyenne pour les deux classes 1 et 3 dans le tableau 7.1, cela explique le fait que le nombre des objets mal classés obtenu par l'algorithme NEM_1 est plus important (24).

Nous proposons dans la suite de ce travail, une méthode d'estimation du rayon ρ.

TABLE 7.1 – Résultats de l'algorithme NEM avec les rayon $\rho \in \{1, \sqrt{2}, 2\}$ de suite, *nb.mc* correspond aux nombre d'objets mal classés.

Algorithme	proportions	Centres	*nb.mc*
EM_{vMF}	$\begin{bmatrix} 56.97 \\ 35.99 \\ 27.02 \end{bmatrix}$	$\begin{bmatrix} 0.955 & -0.295 \\ -0.543 & 0.839 \\ 0.011 & -0.999 \end{bmatrix}$	3
$\text{NEM}_{\rho=1}$	$\begin{bmatrix} 60 \\ 60 \end{bmatrix}$	$\begin{bmatrix} 0.793 & -0.608 \\ -0.547 & 0.837 \end{bmatrix}$	24
$\text{NEM}_{\rho=\sqrt{2}}$	$\begin{bmatrix} 62 \\ 36 \\ 22 \end{bmatrix}$	$\begin{bmatrix} 0.929 & -0.368 \\ -0.544 & 0.838 \\ 0.017 & -0.999 \end{bmatrix}$	2
$\text{NEM}_{\rho=2}$	$\begin{bmatrix} 57 \\ 36 \\ 27 \end{bmatrix}$	$\begin{bmatrix} 0.960 & -0.279 \\ -0.543 & 0.839 \\ 0.013 & -0.999 \end{bmatrix}$	3

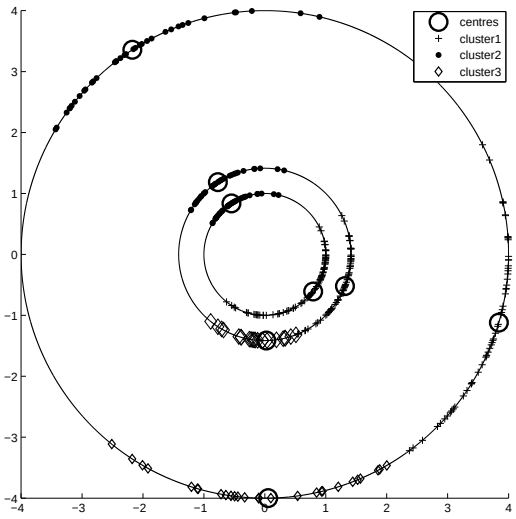

FIGURE 7.1 – Représentation des résultats de la classification avec des rayons $\rho = 1, \sqrt{2}, 2$. Deux classes avec NEM_1 et trois classes avec $\text{NEM}_{\rho=\sqrt{2}}$ et $\text{NEM}_{\rho=2}$

2.3 Du NEM au NEM_ρ : un algorithme EM normalisé de rayon ρ

Notre objectif dans ce chapitre est d'estimer la valeur du rayon de l'hypersphère, afin de l'utiliser dans l'algorithme NEM (Phuong et Vinh, 2008). Phuong Vinh et dans leurs applications considèrent le cas ou le rayon est un nombre entier supérieurs ou égale à un. Dans notre travail le rayon sera estimée comme valeur réelle supérieure ou égale à un. Les cas où les valeurs de rayon sont censés inférieure à un font l'objet de sur-apprentissage (sur-ajustement) qui n'est pas bon du tout, généralement ils ont de mauvaise performance

prédictive.

On note NEM$_\rho$ l'algorithme obtenu avec ce changement. On suppose que chaque composant appartient à la même hypersphère.

A une itération (c), l'algorithme NEM$_\rho$ succède respectivement les deux étapes E et M : Espérance et Maximisation, dans le contexte d'un modèle de mélange de g composants de distributions exponentielles.

Algorithme 18: NEM$_\rho$

1 : L'étape Espérance pour le calcul des probabilités a posteriori t_{ik}.

2 : L'étape Maximisation, pour l'estimation des paramètres (ρ, π, μ) maximisant :

$$Q(\theta, \theta) = \sum_{i,k} t_{ik} \log\left(\pi_k \gamma_\rho e^{-\|\mathbf{x}_i - \mu_k\|^2}\right)$$

3 : Répéter les étapes 1 et 2 jusqu'à la convergence.

A une itération (c), maximiser la log-vraisemblance revient à maximiser la quantité $Q(\theta, \theta^{(c)})$ définie par l'expression 2.1 :

$$\begin{aligned} Q(\theta, \theta^{(c)}) &= \sum_{i,k} t_{ik}^{(c)} \log\left(\pi_k \gamma_\rho e^{-\|\mathbf{x}_i - \mu_k\|^2}\right) \\ &= \sum_{i,k} t_{ik}^{(c)} \log(\pi_k) + \sum_{i,k} t_{ik}^{(c)} \log(\gamma_\rho) + \sum_{i,k} t_{ik}^{(c)} \log\left(e^{-\|\mathbf{x}_i - \mu_k\|^2}\right) \\ &= \sum_{i,k} t_{ik}^{(c)} \log(\pi_k) + \log(\gamma_\rho) \sum_{i,k} t_{ik}^{(c)} - \sum_{i,k} t_{ik}^{(c)} \|\mathbf{x}_i - \mu_k\|^2 \end{aligned}$$

Comme $\|\mathbf{x}_i - \mu_k\|^2 = \|\mathbf{x}_i\|^2 + \|\mu_k\|^2 - 2\langle \mathbf{x}_i, \mu_k \rangle$, et $\|\mathbf{x}_i\| = \|\mu_k\| = \rho$, la quantité $Q(\theta, \theta^{(c)})$ devient

$$Q(\theta, \theta^{(c)}) = \sum_{i,k} t_{ik}^{(c)} \log(\pi_k) + \log(\gamma_\rho) \sum_{i,k} t_{ik}^{(c)} - \sum_{i,k} t_{ik}^{(c)} \left(2\rho^2 - 2\langle \mathbf{x}_i, \mu_k \rangle\right)$$

ou

$$Q(\theta, \theta^{(c)}) = \sum_{i,k} t_{ik}^{(c)} \log(\pi_k) + \log(\gamma_\rho) \sum_{i,k} t_{ik}^{(c)} - \sum_{i,k} t_{ik}^{(c)} 2\rho^2 + \sum_{i,k} t_{ik}^{(c)} 2\langle \mathbf{x}_i, \mu_k \rangle$$

enfin,

$$Q(\theta, \theta^{(c)}) = n \log(\gamma_\rho) - 2n\rho^2 + \sum_{i,k} t_{ik}^{(c)} \left[\log(\pi_k^{(c-1)}) + 2\mu_k^{(c-1)t} \mathbf{x}_i\right].$$

Dans l'étape maximisation de l'algorithme, à chaque étape on cherche à estimer les paramètres $\theta_k = (\pi_k, \mu_k, \rho_k)$ comme suit :

1. Estimation des proportions π_k pour $k = 1, \ldots, g$:

$$\frac{\partial}{\partial \pi_k}[\sum_{i=1}^{n} t_{ik} \log(\pi_k)] = 0$$

$$\Leftrightarrow \quad \frac{\partial}{\partial \pi_k}[\sum_{i=1}^{n} t_{ik} \log(\pi_k) - \lambda(\sum_{k=1}^{g} \pi_k - 1)] = 0, \ \sum_{k=1}^{g} \pi_k = 1$$

$$\frac{1}{\pi_k} \sum_{i=1}^{n} t_{ik} = \lambda, \ \lambda = n \Rightarrow \pi_k = \frac{\sum_{i=1}^{n} t_{ik}}{n}.$$

2. Estimation des centres μ_k pour $k = 1, \ldots, g$:

$$\frac{\partial}{\partial \mu_k}[\sum_{i=1}^{n} 2 \langle \mathbf{x}_i, \mu_k \rangle t_{ik} - \lambda(\mu_k^t \mu_k - \rho^2)] = 0, \ \mu_k^t \mu_k = \rho^2$$

$$\Leftrightarrow 2 \sum_{i=1}^{n} \mathbf{x}_i t_{ik} - 2\lambda \mu_k = 0, \ \|\mu\| = \rho = \frac{\|\sum_{i=1}^{n} \mathbf{x}_i p(k/\mathbf{x}_i, \theta)\|}{\lambda}$$

$$\Rightarrow \mu_k = \rho \frac{\sum_{i=1}^{n} \mathbf{x}_i t_{ik}}{\|\sum_{i=1}^{n} \mathbf{x}_i t_{ik}\|}$$

3. Estimation du rayon ρ

   Cette étape est assez longue et compliquée du point de vue calculs.

   Nous considérons que chaque composant appartient à la même hypersphère de rayon ρ, on surnomme ϕ_{ik} l'angle entreμ_k et $\mathbf{x}_i$ alors $\langle \mathbf{x}_i, \mu_k \rangle = \rho^2 cos\phi_{ik}$

   L'estimateur de ρ devrait maximiser $Q(\theta, \theta')$, il est équivalent à maximiser :

$$n \log(\gamma_\rho) - 2n\rho^2 + 2 \sum_{k=1}^{g} \sum_{i=1}^{n} (\rho^2 cos\phi_{ik}) t_{ik}$$

$$\begin{aligned}
&\Rightarrow \quad \frac{d}{d\rho}[n \log(\gamma_\rho) - 2n\rho^2 + 2 \sum_{k=1}^{g} \sum_{i=1}^{n} (\rho^2 cos\phi_{ik}) t_{ik}] = 0 \\
&\Rightarrow \quad n \frac{\gamma'_\rho}{\gamma_\rho} - 4n\rho + 2 \sum_{k=1}^{g} \sum_{i=1}^{n} (2\rho cos\phi_{ik}) t_{ik} = 0 \\
&\Rightarrow \quad n \frac{\gamma'_\rho}{\gamma_\rho} - 4n\rho + \frac{4}{\rho} \sum_{k=1}^{g} \sum_{i=1}^{n} (\rho^2 cos\phi_{ik}) t_{ik} = 0 \\
&\Rightarrow \quad n \frac{\gamma'_\rho}{\gamma_\rho} - 4n\rho + \frac{2}{\rho} \sum_{k=1}^{g} \sum_{i=1}^{n} 2 \langle \mathbf{x}_i, \mu_k \rangle t_{ik} = 0
\end{aligned}$$

$$\Rightarrow 2n\rho^2 - \frac{n\rho}{2}\frac{\gamma'_\rho}{\gamma_\rho} = \sum_{k=1}^{g}\sum_{i=1}^{n} 2\left\langle \mathbf{x}_i, \mu_k \right\rangle t_{ik} \tag{7.2}$$

on a $\gamma_\rho = [\int_{\mathbf{x}\in S_\rho^{d-1}} \exp\left(-\left\|\mathbf{x}-\mu\right\|^2\right) d\mathbf{x}]^{-1}$
Maintenant nous avons besoin de calculer l'intégrale $I = \int_{\mathbf{x}\in S_\rho^{d-1}} \exp\left(-\left\|\mathbf{x}-\mu\right\|^2\right) d\mathbf{x}$.
En coordonnées sphériques, chaque $\mathbf{x} \in S_\rho^{d-1}$ s'écrit de la forme :
$\mathbf{x} = (\rho\sin\theta_1\sin\theta_2\ldots\sin\theta_{d-1}, \rho\sin\theta_1\sin\theta_2\ldots\sin\theta_{d-2}\cos\theta_{d-1}, \ldots, \rho\sin\theta_1\cos\theta_2, \rho\cos\theta_1)$
on suppose que μ se trouve sur un axe fixé de la sphère, alors $\mu = (0, \ldots, 0, \rho)$, donc :

$$\left\|\mathbf{x}-\mu\right\|^2 = \rho^2\prod_{i=1}^{d-1}\sin^2\theta_i + \rho^2\cos^2\theta_{d-1}\prod_{i=1}^{d-2}\sin^2\theta_i + \cdots + \rho^2\cos^2\theta_2\sin^2\theta_1 + (\rho\cos\theta_1 - \rho)^2)$$

qui peut s'écrire

$$\rho^2\sin^2\theta_{d-1}\prod_{i=1}^{d-2}\sin^2\theta_i + \rho^2\cos^2\theta_{d-1}\prod_{i=1}^{d-2}\sin^2\theta_i + \cdots + \rho^2\cos^2\theta_2\sin^2\theta_1 + \rho^2\cos^2\theta_1 - 2\rho\cos\theta_1 + \rho^2$$

En utilisant l'égalité : $\sin^2\theta_i + \cos^2\theta_i = 1$, on trouve

$$\left\|\mathbf{x}-\mu\right\|^2 = 2\rho^2(1-\cos\theta_1).$$

Remarquons que la quantité $\left\|\mathbf{x}-\mu\right\|^2$ ne dépend pas de d, alors :

$$I = \int_{\mathbf{x}\in S_\rho^{d-1}} \exp\left(-\left\|\mathbf{x}-\mu\right\|^2\right) d\mathbf{x} = \int_{\mathbf{x}\in S_\rho^{d-1}} \exp\left(-2\rho^2(1-\cos\theta_1)\right) d\theta.$$

Pour le calcul de l'intégrale I, nous devons étudier trois cas : le cas où $d = 2$ ensuite $d = 3$ et enfin le cas général et le plus compliqué $d \geq 3$.

Cas où $d = 2$

Le jacobien des coordonnées circulaires est $|\Delta| = \rho$, $\mathbf{x} = (\rho\sin\theta_1, \rho\cos\theta_1)$
on pose $\theta_1 = \theta$, l'intégrale à calculer est :

$$\begin{aligned} I &= \int_{\mathbf{x}\in S_\rho} \exp(-\left\|\mathbf{x}-\mu\right\|^2) d\mathbf{x} \\ &= \int_0^{2\pi} \rho\exp\left(-2\rho^2(1-\cos\theta)\right) d\theta \\ &= \rho\exp\left(-2\rho^2\right)\int_0^{2\pi}\exp\left(2\rho^2\cos\theta\right) d\theta \\ &= 2\pi\rho\exp\left(-2\rho^2\right) I_0(2\rho^2). \end{aligned}$$

$$\gamma'_\rho = -\frac{\exp\left(-2\rho^2\right)I_0(2\rho^2) - 4\rho^2\exp\left(-2\rho^2\right)I_0(2\rho^2) + 4\rho^2\exp\left(-2\rho^2\right)I_1(2\rho^2)}{2\pi\left[\rho\exp\left(-2\rho^2\right)I_0(2\rho^2)\right]^2}$$

alors

$$\begin{aligned}\frac{\gamma'_\rho}{\gamma_\rho} &= \frac{-I_0(2\rho^2)+4\rho^2 I_0(2\rho^2)-4\rho^2 I_1(2\rho^2)}{2\pi\exp\left(-2\rho^2\right)\left[\rho I_0(2\rho^2)\right]^2}2\pi\rho\exp\left(-2\rho^2\right)I_0(2\rho^2)\\ &= \frac{-I_0(2\rho^2)+4\rho^2 I_0(2\rho^2)-4\rho^2 I_1(2\rho^2)}{\rho I_0(2\rho^2)}\\ &= \frac{-1}{\rho}+4\rho-4\rho\frac{I_1(2\rho^2)}{I_0(2\rho^2)}\end{aligned}$$

En remplaçant l'expression de $\frac{\gamma'_\rho}{\gamma_\rho}$ dans l'équation 7.2, on trouve :

$$2n\rho^2-\frac{n\rho}{2}\left[\frac{-1}{\rho}+4\rho-4\rho\frac{I_1(2\rho^2)}{I_0(2\rho^2)}\right]=\sum_{k=1}^{g}\sum_{i=1}^{n}2\left\langle\mathbf{x}_i,\mu_k\right\rangle t_{ik}$$

ρ est donc solution de l'équation :

$$4n\rho^2\frac{I_1(2\rho^2)}{I_0(2\rho^2)}+n=2\sum_{k=1}^{g}\sum_{i=1}^{n}2\left\langle\mathbf{x}_i,\mu_k\right\rangle t_{ik}$$

$$\Leftrightarrow \rho^2\frac{I_1(2\rho^2)}{I_0(2\rho^2)}+\frac{1}{4}=\frac{1}{n}\sum_{k=1}^{g}\sum_{i=1}^{n}\left\langle\mathbf{x}_i,\mu_k\right\rangle t_{ik}, \tag{7.3}$$

L'équation 7.3 donne une estimation numérique de ρ dans le cas où $d=2$.

Cas où $d=3$ (Bouberima *et al.*, 2011)

En utilisant les coordonnées sphériques pour $d=3$:

$M(\mathbf{x})=(\rho,\theta,\phi)$
$\overrightarrow{OM}=\rho\sin\phi\cos\theta\ \overrightarrow{i}+\rho\sin\phi\sin\theta\ \overrightarrow{j}+\rho\cos\phi\ \overrightarrow{k}$
Nous savons que $d\mathbf{x}=\rho^2\sin\phi d\phi d\theta$, alors

$$\begin{aligned}I &= \int_{\mathbf{x}\in S_\rho^2}\exp(-\left\|\mathbf{x}-\mu\right\|^2)d\mathbf{x}\\ &= \int_0^{2\pi}\int_0^{\pi}\exp(-2\rho^2(1-\cos\phi))\rho^2\sin\phi d\phi d\theta\\ &= \exp(-2\rho^2)\int_0^{2\pi}\int_0^{\pi}\rho^2\exp(2\rho^2\cos\phi)\sin\phi d\phi d\theta\\ &= 2\pi\exp(-2\rho^2)\int_0^{\pi}\rho^2\exp(2\rho^2\cos\phi)\sin\phi d\phi\\ &= -\pi\exp(-2\rho^2)\int_0^{\pi}-2\rho^2\sin\phi\exp(2\rho^2\cos\phi)d\phi\\ &= -\pi\exp(-2\rho^2)[\exp(2\rho^2\cos\phi)]_0^{\pi}\\ &= -\pi\exp(-2\rho^2)[\exp(-2\rho^2)-\exp(2\rho^2)]\end{aligned}$$

Enfin

$$I=\pi[1-\exp(-4\rho^2)]$$

et

$$\gamma_\rho = I^{-1} = [\pi(1 - \exp(-4\rho^2))]^{-1}$$

donc

$$\gamma'_\rho = -8\pi\rho \exp(-4\rho^2)[\pi(1 - \exp(-4\rho^2))]^{-2}$$

par la suite :

$$\frac{\gamma'_\rho}{\gamma_\rho} = \frac{-8\rho \exp(-4\rho^2)}{(1 - \exp(-4\rho^2))}$$

En remplaçant cette dernière expression dans l'équation 7.2, on obtient :

$$2n\rho^2 + n\rho\frac{4\rho \exp(-4\rho^2)}{(1 - \exp(-4\rho^2))} = \sum_{k=1}^{g}\sum_{i=1}^{n} 2\langle \mathbf{x}_i, \mu_k\rangle t_{ik}$$

ρ est donc solution de l'équation :

$$\Leftrightarrow \rho^2\left[1 + \frac{2}{\exp(4\rho^2) - 1}\right] = \frac{1}{n}\sum_{k=1}^{g}\sum_{i=1}^{n}\langle \mathbf{x}_i, \mu_k\rangle t_{ik} \tag{7.4}$$

L'équation (7.4) donne une estimation numérique de ρ dans le cas où $d = 3$.

Cas général des dimensions $d \geq 3$

Le déterminant de la jacobienne des coordonnées sphériques dans le cas multidimensionnelle s'écrit :

$$|\Delta| = \rho^{d-1}\prod_{i=1}^{d-2}\sin^i(\theta_{d-i-1})$$

Calculons maintenant l'intégrale I :

$$\begin{aligned}
I &= \int_{\mathbf{x}\in S_\rho^{d-1}} \exp(-\|\mathbf{x} - \mu\|^2)d\mathbf{x} \\
&= \int_0^{2\pi}\int_0^{\pi}\dots\int_0^{\pi}\exp(-2\rho^2(1 - \cos\theta_1))\rho^{d-1}\prod_{i=1}^{d-2}\sin^i(\theta_{d-i-1})d\theta_1\dots d\theta_{d-1} \\
&= \int_0^{2\pi} d\theta_{d-1}\prod_{i=2}^{d-2}\int_0^{\pi}\sin^{d-i-1}(\theta_i)d\theta_i\int_0^{\pi}\exp(-2\rho^2(1 - \cos\theta_1))\rho^{d-1}\sin^{d-2}(\theta_1)d\theta_1
\end{aligned}$$

On pose :

$$K = \prod_{i=2}^{d-2}\int_0^{\pi}\sin^{d-i-1}(\theta_i)d\theta_i$$

et,

$$H = \int_0^{\pi}\exp(-2\rho^2(1 - \cos\theta_1))\rho^{d-1}\sin^{d-2}(\theta_1)d\theta_1$$

Pour calculer l'intégrale I, on doit tout d'abord calculer séparément les trois quantités : $\int_0^{2\pi} d\theta_{d-1}$, K et H.

(a) $\int_0^{2\pi} d\theta_{d-1} = 2\pi$.

(b) Calcul de K

Posons $J = \int_0^{\pi} \sin^{d-i-1}(\theta_i) d\theta_i, \forall\ i = 2, \ldots, d-2$

pour $j = 1, \ldots, d-3$, alors $J = \int_0^{\pi} \sin^{j}(\theta_{d-j-1}) d\theta_{d-j-1} = \int_0^{\pi} \sin^{j}\theta d\theta$,

c'est une intégrale de type similaire aux intégrales de Wallis qui sont de la forme :

$J_{2j} = 2\int_0^{\pi/2} \sin^{2j}(\theta) d\theta = \pi \frac{(2j)!}{2^{2j}(j!)^2}$,

et $J_{2j+1} = 2\int_0^{\pi/2} \sin^{2j+1}(\theta) d\theta = \frac{2^{2j+1}(j!)^2}{(2j+1)!}$,

et donc : $J_{2j} J_{2j+1} = \frac{2\pi}{2j+1}$, dans notre cas : $\prod_{j=1}^{d-3} \int_0^{\pi} \sin^{j}(\theta) d\theta$

si d est un nombre pair $d = 2l$, alors :

$$\begin{aligned}
\prod_{j=1}^{d-3} \int_0^{\pi} \sin^{j}(\theta) d\theta &= \prod_{j=1}^{2(l-1)-1} \int_0^{\pi} \sin^{j}(\theta) d\theta \\
&= \int_0^{\pi} \sin(\theta) d\theta \prod_{j=2}^{2(l-1)-1} \int_0^{\pi} \sin^{j}(\theta) d\theta \\
&= 2 \prod_{j=1}^{(l-1)-1} \int_0^{\pi} \sin^{2j}(\theta) d\theta \int_0^{\pi} \sin^{2j+1}(\theta) d\theta \\
&= 2 \prod_{j=1}^{(l-1)-1} \frac{2\pi}{2j+1} = 2\frac{(2\pi)^{l-2}}{3.5\ldots(2l-3)} = 2\frac{(2\pi)^{\frac{d}{2}-2}}{3.5\ldots(d-3)}
\end{aligned}$$

On sait que :

$$3.5\ldots(d-3) = \frac{(d-3)!}{2^{\frac{d}{2}-1}\left(\frac{d}{2}-1\right)!}$$

d'où :

$$\prod_{j=1}^{d-3} \int_0^{\pi} \sin^{j}(\theta) d\theta = \left(2^{\frac{d}{2}}(\frac{d}{2}-1)!\right) \frac{(2\pi)^{\frac{d}{2}-2}}{(d-3)!}$$

si d est un nombre impair $d = 2l+1$:

$$\begin{aligned}
\prod_{j=1}^{d-3} \int_0^{\pi} \sin^{j}(\theta) d\theta &= \prod_{j=1}^{2(l-1)} \int_0^{\pi} (\sin(\theta))^{j} d\theta \\
&= \prod_{j=1}^{l-1} \int_0^{\pi} \sin^{2j-1}(\theta) d\theta \int_0^{\pi} \sin^{2j}(\theta) d\theta \\
&= \prod_{j=1}^{l-1} \frac{2\pi}{2j} = \frac{(\pi)^{l-1}}{(l-1)!} = \frac{(\pi)^{\frac{d-3}{2}}}{(\frac{d-3}{2})!}
\end{aligned}$$

d'où :

$$K = \begin{cases} \frac{(\pi)^{\frac{d-3}{2}}}{(\frac{d-3}{2})!}, \text{ si } d \text{ est impaire} \\ \\ \left(2^{\frac{d}{2}}(\frac{d}{2}-1)!\right) \frac{(2\pi)^{\frac{d}{2}-2}}{(d-3)!}, \text{ si } d \text{ est paire} \end{cases}$$

(c) Calcul de H :

$$\begin{aligned} H &= \int_0^{\pi} \exp\left(-2\rho^2(1-\cos\theta_1)\right)\rho^{d-1}\sin^{d-2}(\theta_1)d\theta_1 \\ &= \rho^{d-1}\exp\left(-2\rho^2\right)\int_0^{\pi}\sin^{d-2}(\theta_1)\exp\left(2\rho^2\cos\theta_1\right)d\theta_1 \end{aligned}$$

Nous pouvons calculer H en utilisant la formule suivante (Gradshteyn et Ryzhik, 2007) :

$$\int_0^{\pi}\exp\left(\pm\beta\cos x\right)\sin^{2\nu}x dx = \sqrt{\pi}(\frac{2}{\beta})^{\nu}\Gamma(\nu+\frac{1}{2})I_{\nu}(\beta),\ Re(\nu)\succ\frac{-1}{2}$$

Γ et I_{ν} sont respectivement les deux fonctions connues, gamma et Bessel du premier type, donc,

$$\begin{aligned} H &= \rho^{d-1}\exp\left(-2\rho^2\right)\int_0^{\pi}\sin^{d-2}(\theta_1)\exp\left(2\rho^2\cos\theta_1\right)d\theta_1 \\ &= \rho^{d-1}\exp\left(-2\rho^2\right)\sqrt{\pi}(\frac{2}{2\rho^2})^{\frac{d-2}{2}}\Gamma(\frac{d-2}{2}+\frac{1}{2})I_{\frac{d-2}{2}}(2\rho^2) \\ &= \pi^{\frac{1}{2}}\rho\exp\left(-2\rho^2\right)\Gamma(\frac{d-1}{2})I_{\frac{d-2}{2}}(2\rho^2) \end{aligned}$$

on remplace maintenant les quantités calculées ci-dessus : $\int_0^{2\pi}d\theta_{d-1}$, K et H dans I, pour chaque $d\succ 3$:

$$I=\begin{cases} (2\pi)\left[\frac{(\pi)^{\frac{d-3}{2}}}{(\frac{d-3}{2})!}\right]\left[\pi^{\frac{1}{2}}\rho\exp\left(-2\rho^2\right)\Gamma(\frac{d-1}{2})I_{\frac{d}{2}-1}(2\rho^2)\right],\ \text{si } d \text{ est impaire} \\ (2\pi)\left[\left(2^{\frac{d}{2}}(\frac{d}{2}-1)!\right)\frac{(2\pi)^{\frac{d}{2}-2}}{(d-3)!}\right]\left[\pi^{\frac{1}{2}}\rho\exp\left(-2\rho^2\right)\Gamma(\frac{d-1}{2})I_{\frac{d}{2}-1}(2\rho^2)\right],\ \text{si } d \text{ est paire} \end{cases}$$

ou encore :

$$I=\begin{cases} 2\rho\exp\left(-2\rho^2\right)\left(\pi^{\frac{d}{2}}\right)I_{\frac{d}{2}-1}(2\rho^2),\ \text{si } d \text{ est impaire} \\ 2\rho\exp\left(-2\rho^2\right)\left(\pi^{\frac{d}{2}}(d-2)\right)I_{\frac{d}{2}-1}(2\rho^2),\ \text{si } d \text{ est paire} \end{cases}$$

On remplace l'expression de I dans celle de γ_{ρ}, on obtient :

$$\gamma_{\rho}=I^{-1}=\begin{cases} \left[2\rho\exp\left(-2\rho^2\right)\left(\pi^{\frac{d}{2}}\right)I_{\frac{d}{2}-1}(2\rho^2)\right]^{-1},\ \text{si } d \text{ est impaire} \\ \left[2\rho\exp\left(-2\rho^2\right)\left(\pi^{\frac{d}{2}}(d-2)\right)I_{\frac{d}{2}-1}(2\rho^2)\right]^{-1},\ \text{si } d \text{ est paire} \end{cases}$$

d'où :

$$\gamma'_\rho = \begin{cases} \dfrac{-\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]'}{\left(\pi^{\frac{d}{2}}\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2}, & \text{si } d \text{ est impaire} \\ \dfrac{-\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]'}{\left(\pi^{\frac{d}{2}}(d-2)\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2}, & \text{si } d \text{ est paire} \end{cases}$$

$$= \begin{cases} -\dfrac{\left[2\rho\exp(-2\rho^2)\right]' I_{\frac{d}{2}-1}(2\rho^2)+\left[2\rho\exp(-2\rho^2)\right]\left[I_{\frac{d}{2}-1}(2\rho^2)\right]'}{\left(\pi^{\frac{d}{2}}\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2}, & \text{si } d \text{ est impaire} \\ -\dfrac{\left[2\rho\exp(-2\rho^2)\right]' I_{\frac{d}{2}-1}(2\rho^2)+\left[2\rho\exp(-2\rho^2)\right]\left[I_{\frac{d}{2}-1}(2\rho^2)\right]'}{\left(\pi^{\frac{d}{2}}(d-2)\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2}, & \text{si } d \text{ est paire} \end{cases}$$

$$= \begin{cases} -\dfrac{\left[2\exp(-2\rho^2)-8\rho^2\exp(-2\rho^2)\right] I_{\frac{d}{2}-1}(2\rho^2)+\left[2\rho\exp(-2\rho^2)\right]4\rho I'_{\frac{d}{2}-1}(2\rho^2)}{\left(\pi^{\frac{d}{2}}\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2}, & \text{si } d \text{ est impaire} \\ -\dfrac{\left[2\exp(-2\rho^2)-8\rho^2\exp(-2\rho^2)\right] I_{\frac{d}{2}-1}(2\rho^2)+\left[2\rho\exp(-2\rho^2)\right]4\rho I'_{\frac{d}{2}-1}(2\rho^2)}{\left(\pi^{\frac{d}{2}}(d-2)\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2}, & \text{si } d \text{ est paire} \end{cases}$$

$$= \begin{cases} -\dfrac{\left[1-4\rho^2\right]2\exp(-2\rho^2) I_{\frac{d}{2}-1}(2\rho^2)+\left[2\rho\exp(-2\rho^2)\right]4\rho I'_{\frac{d}{2}-1}(2\rho^2)}{\left(\pi^{\frac{d}{2}}\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2}, & \text{si } d \text{ est impaire} \\ -\dfrac{\left[1-4\rho^2\right]2\exp(-2\rho^2) I_{\frac{d}{2}-1}(2\rho^2)+\left[2\rho\exp(-2\rho^2)\right]4\rho I'_{\frac{d}{2}-1}(2\rho^2)}{\left(\pi^{\frac{d}{2}}(d-2)\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2}, & \text{si } d \text{ est paire} \end{cases}$$

alors :

$$\frac{\gamma'_\rho}{\gamma_\rho} = \begin{cases} -\dfrac{\left[1-4\rho^2\right]2\exp(-2\rho^2) I_{\frac{d}{2}-1}(2\rho^2)+\left[2\rho\exp(-2\rho^2)\right]4\rho I'_{\frac{d}{2}-1}(2\rho^2)}{\left(\pi^{\frac{d}{2}}\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2\left[2\rho\exp(-2\rho^2)\left(\pi^{\frac{d}{2}}\right)I_{\frac{d}{2}-1}(2\rho^2)\right]^{-1}}, & \text{si } d \text{ est impaire} \\ -\dfrac{\left[1-4\rho^2\right]2\exp(-2\rho^2) I_{\frac{d}{2}-1}(2\rho^2)+\left[2\rho\exp(-2\rho^2)\right]4\rho I'_{\frac{d}{2}-1}(2\rho^2)}{\left(\pi^{\frac{d}{2}}(d-2)\right)\left[2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)\right]^2\left[2\rho\exp(-2\rho^2)\left(\pi^{\frac{d}{2}}(d-2)\right)I_{\frac{d}{2}-1}(2\rho^2)\right]^{-1}}, & \text{si } d \text{ est paire} \end{cases}$$

Après simplification des calculs, on obtient une expression unique du rapport $\frac{\gamma'_\rho}{\gamma_\rho}$ quelque soit la valeur de d (paire ou impaire)

$$\frac{\gamma'_\rho}{\gamma_\rho} = \frac{\left[4\rho^2-1\right]2\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2) - \left[2\rho\exp(-2\rho^2)\right]4\rho I'_{\frac{d}{2}-1}(2\rho^2)}{2\rho\exp(-2\rho^2)I_{\frac{d}{2}-1}(2\rho^2)}$$

$$= \frac{(4\rho^2-1)}{\rho} - \frac{4\rho I'_{\frac{d}{2}-1}(2\rho^2)}{I_{\frac{d}{2}-1}(2\rho^2)}$$

On remplace $I'_{\frac{d}{2}-1}(2\rho^2)$ par la relation de récurrence ci-dessous (Gradshteyn et Ryzhik, 2007) propre aux fonctions de Bessel.

$$2\rho^2 I'_{\frac{d}{2}-1}(2\rho^2) = -(\frac{d}{2}-1)I_{\frac{d}{2}-1}(2\rho^2) + 2\rho^2 I_{\frac{d}{2}-2}(2\rho^2)$$

on trouve :

$$\begin{aligned}\frac{\gamma'_\rho}{\gamma_\rho} &= \frac{(4\rho^2-1)}{\rho} - \frac{4\rho I_{\frac{d}{2}-2}(2\rho^2) - \frac{d-2}{\rho} I_{\frac{d}{2}-1}(2\rho^2)}{I_{\frac{d}{2}-1}(2\rho^2)} \\ &= \frac{(4\rho^2+d-3)}{\rho} - 4\rho \frac{I_{\frac{d}{2}-2}(2\rho^2)}{I_{\frac{d}{2}-1}(2\rho^2)}\end{aligned}$$

on remplace cette dernière expression dans l'équation 7.2, on obtient :

$$\rho^2 \frac{I_{\frac{d}{2}-2}(2\rho^2)}{I_{\frac{d}{2}-1}(2\rho^2)} - \frac{d-3}{4} = \frac{1}{n}\sum_{k=1}^{g}\sum_{i=1}^{n} \langle \mathbf{x}_i, \mu_k \rangle \, t_{ik}, \tag{7.5}$$

L'équation (7.5) donne une estimation numérique de ρ dans le cas général ($d \geq 3$). Cette estimation caractérisera la méthode proposée dans ce paragraphe à savoir l'algorithme NEM$_\rho$.

Dans le paragraphe qui suit, on va clarifier une fonctionnalité importante du calcul intégral sur les hypersphères ; cette particularité influence directement sur le calcul de ρ.

2.4 Difficulté du calcul intégral sur l'hypersphère

L'estimation de ρ dans ce chapitre est basé sur la possibilité de permettre à un grand nombre de données d'appartenir à la même hypersphère de rayon plus grand que 1, puisqu'on a besoin de faire des calculs d'intégrales sur l'hypersphère S_ρ^{d-1} de rayon $\rho \geq 1$. Tout d'abord, on va observer le calcul de la surface d'une hypersphère S_ρ^{d-1} pour différentes dimensions, il est donné par la formule :

$$S = \begin{cases} \frac{2\rho^d \pi^{\frac{d+1}{2}}}{\Gamma(\frac{d+1}{2})}, \text{ si } d \text{ est impaire} \\ \\ \frac{\rho^d 2^{\frac{d}{2}+1} \pi^{\frac{d}{2}}}{\frac{\Gamma(d)}{2\Gamma(\frac{d}{2})}} = \frac{\rho^d \pi^{\frac{d+1}{2}}}{2^{\frac{d}{2}-3}\Gamma(\frac{d+1}{2})}, \text{ si } d \text{ est paire} \end{cases}$$

Pour commencer nous allons fixer $\rho = 1$ et on varie la dimension, les valeurs des surfaces obtenues sont représentées dans le tableau (7.2) :

En lisant les valeurs obtenues (tableau 7.2), nous pouvons voir qu'avec l'augmentation respective de la dimension paire et impaire, la surface augmente, jusqu'à une valeur pic après laquelle les valeurs de la surface diminue. Pour avoir une idée plus claire sur ce

dimension	2	3	4	5	6	7	8	9	10	11
surface	25.13	19.73	26.31	31.00	16.53	32.46	7.42	25.50	2.59	16.02
dimension	12	13	14	15	16	17	18	19	20	21
surface	0.73	8.38	0.17	3.76	0.03	1.47	0.006	0.51	0.001	0.16

TABLE 7.2 – valeurs de la surface de l'hypersphère unitaire pour différentes dimensions

comportement, nous avons fait deux représentations, l'une pour les valeurs de dimensions paire (* bleu) et l'autre pour les valeurs de dimensions impaires (o rouge) dans la figure (7.2).

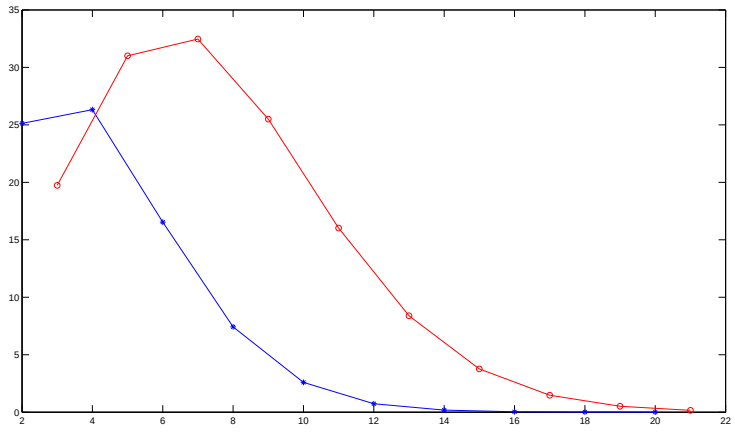

FIGURE 7.2 – Représentation des valeurs de surface de l'hypersphère unitaire pour les dimensions paires et impaires

Signalons que la surface d'une hypersphère de dimension impaire et vérifiant $d \succ 3$ est toujours plus grande que la surface de celle de dimension paire suivante $d+1$, puis dans les deux cas, il y a un pic de valeur de la surface, ceci est vrai quelque soit la valeur du rayon, la surface augmente avec la dimension, jusqu'à un pic de dimension qui dépend du rayon ρ, assisté par :

$$surface = \begin{cases} 2\log(\pi\rho^2) - 1, \ d \text{ est impaire} \\ \\ 2\log(\frac{\pi}{2}\rho^2) - 1, \ d \text{ est paire} \end{cases}$$

A cette fin, nous avons prouvé l'importance de la mise en œuvre des formes sphériques de rayon $\rho \geq 1$ et nous avons à souligné le problème principal du calcul : les intégrales sur toute hypersphère dépendent toujours et fortement de la dimension traitée (paire ou impaire).

2.5 Une version Hard de l'algorithme NEM$_\rho$: NCEM$_\rho$

L'algorithme NCEM$_\rho$ est la version dure de l'algorithme NEM$_\rho$; il est exécuté à chaque itération (c) en remplaçant $Q(\theta, \theta^{(c)})$ par $L_C(\theta)$; une condition forte où les t_{ik} sont convertis en z_{ik}.

$$\begin{aligned} L_C(\theta) &= \sum_{i=1}^{} \sum_{k=1}^{g} z_{ik}^{(c)} \log(\pi_k \varphi_k(\mathbf{x}_i; \alpha_k)) \\ &= \sum_{k=1}^{g} \sum_{\mathbf{x}_i \in P_k} \log\left(\pi_k \gamma_\rho e^{-\|\mathbf{x}_i - \mu_k\|^2}\right). \end{aligned}$$

Les principales modifications apportées à NEM$_\rho$ concernent le calcule et la maximisation de la log-vraisemblance des données complétées $\mathbf{y} = (\mathbf{x}, \mathbf{z})$. Les vecteurs indiquant les composantes manquantes (z_{ik}) de chaque point de l'échantillon, sont inclus dans l'ensemble de données.

Nous proposons ici une version classification de l'algorithme NEM$_\rho$, qu'on appellera NCEM$_\rho$. A partir d'une valeur initiale θ^0, une étape de classification est introduite entre les deux étapes E et M, affectant chaque point $\mathbf{x}_i$ au composant qui maximise la probabilité conditionnelle pour former une partition $P^{(c)} = (P_1^{(c)}, ..., P_g^{(c)})$.

Par conséquent, les estimateurs de l'algorithme NCEM$_\rho$ sont les suivants :

$$\pi_k^{(c)} = \frac{\sum_{i=1}^{n} z_{ik}^{(c)}}{n},$$

et

$$\mu_k^{(c)} = \sqrt{\rho} \frac{\sum_{i=1}^{n} z_{ik}^{(c)} \mathbf{x}_i}{\left\| \sum_{i=1}^{n} z_{ik}^{(c)} \mathbf{x}_i \right\|}.$$

l'expression de $L_C(\theta)$ s'écrit :

$$L_C(\theta) = -\sum_{k=1}^{g} \sum_{\mathbf{x}_i \in P_k} \|\mathbf{x}_i - \mu_k\|^2 + \sum_{k=1}^{g} \#P_k \pi_k + n \log \gamma_\rho,$$

où # dénote la cardinalité.

Il est facile de montrer que si $\rho = 1$ et les proportions du mélange sont égales, la maximisation de $L_C(\theta)$ est équivalente à la maximisation du critère du $k$means sphérique suivant :

$$W = \sum_{k=1}^{g} \sum_{\mathbf{x}_i \in P_k} \mathbf{x}_i^t \mu_k.$$

2.6 Une version Stochastique : NSEM$_\rho$

La version stochastique de l'algorithme NEM$_\rho$ appelée NSEM$_\rho$, intègre une restauration des étiquettes des composants inconnus $(z_{ik}, i = 1, \ldots, n, k = 1, \ldots, g)$ entre les deux étapes E et M, en les tirant au hasard dans leurs distributions conditionnelles courantes.

Algorithme 19: NCEM$_\rho$

1 : L'étape Estimation, calculer les probabilités a posteriori t_{ik}.

2 : L'étape Classification, attribuer chaque point $\mathbf{x}_i$ au composant qui maximise la probabilité conditionnelle et former une partition $P = (P_1, ..., P_g)$.

3 : L'étape Maximisation, estimer les paramètres (ρ, π, μ) maximisant : $L_C(\theta)$.

4 : Répéter les étapes 1 à 3 jusqu'à la convergence.

Comme nous l'avons vu précédemment, dans l'étape stochastique S, l'algorithme attribue chaque point à un composant quelconque du mélange en fonction de la distribution multinomiale e_{ik}, de paramètres : valeurs des probabilités a posteriori t_{ik}.

Algorithme 20: L'algorithme NSEM$_\rho$

1 : L'étape Estimation, Calculer les probabilités conditionnelles t_{ik} $(i = 1, ..., n, k = 1, ..., g)$.

2 : L'étape Stochastique S : Attribuer chaque point x_i au hasard à un composant en fonction de la distribution multinomiale de paramètres t_{ik} pour produire une partition $P = (P_1, ..., P_g)$.

3 : L'étape Maximisation, calculer l'estimateur ML θ en utilisant la partition obtenue à l'étape S.

4 : Répéter les étapes 1 à 3 jusqu'à la convergence.

Cette version stochastique converge en probabilité, elle génère une chaîne de Markov, dont la distribution est plus ou moins concentrée autour des estimations ML.

2.7 Comparaison entre l'algorithme NEM$_\rho$ et l'algorithme EM$_{vMF}$

Les deux algorithmes EM$_{vMF}$ et NEM$_\rho$ sont de forme exponentielle, à l'exception que NEM$_\rho$ s'applique sur des données $\rho-$normalisées, de sorte que $\mathbf{x} = \rho\frac{\mathbf{x}}{\|\mathbf{x}\|}$.

Pour faire une comparaison entre les deux distributions, nous avons jugé nécessaire de prendre $\rho = 1$ dans la distribution (7.1) :

$$\varphi(\mathbf{x}; \alpha) = \gamma_1 \exp\left(-\|\mathbf{x} - \mu\|^2\right) = \gamma_1 \exp(-2) \exp(2\mu^t \mathbf{x})$$

avec :

$$\gamma_1 = \begin{cases} \left[2\exp(-2)\left(\pi^{\frac{d}{2}}\right) I_{\frac{d}{2}-1}(2)\right]^{-1}, \text{ si } d \text{ est impaire} \\ \\ \left[2\exp(-2)\left(\pi^{\frac{d}{2}}(d-2)\right) I_{\frac{d}{2}-1}(2)\right]^{-1}, \text{ si } d \text{ est paire} \end{cases}$$

et dans la distribution vMF, nous avons choisi une concentration $\xi = 2$, alors sa distribu-

tion s'écrit :

$$f(\mathbf{x};\alpha) = c_d(2)\exp(2\mu^t\mathbf{x})$$

Remarquons que l'expression des deux distributions : exponentielle et vMF dans ce cas particulier sont similaires :

$$\begin{cases} \varphi(\mathbf{x};\alpha) = \gamma_1 \exp(-2)\exp(2\mu^t\mathbf{x}) \\ \gamma_1 = \frac{\exp(2)}{2a\pi^{\frac{d}{2}} I_{\frac{d}{2}-1}(2)}. \end{cases} \quad \text{et} \quad \begin{cases} f(\mathbf{x};\alpha) = c_d(2)\exp(2\mu^t\mathbf{x}) \\ c_d(2) = \frac{1}{2\pi^{\frac{d}{2}} I_{\frac{d}{2}-1}(2)}. \end{cases}$$

où

$$\begin{cases} a = d-2, \text{ si } d \text{ est paire} \\ \\ a = 1, \text{ si } d \text{ est impaire} \end{cases}$$

$$\begin{cases} \varphi(\mathbf{x};\alpha) = \frac{\exp(2\mu^t\mathbf{x})}{2a\pi^{\frac{d}{2}} I_{\frac{d-2}{2}}(2)} \\ f(\mathbf{x};\alpha) = \frac{\exp(2\mu^t\mathbf{x})}{2\pi^{\frac{d}{2}} I_{\frac{d}{2}-1}(2)} \end{cases}$$

ce qui signifie :

$$\begin{cases} \varphi(\mathbf{x};\alpha) = f(\mathbf{x};\alpha), \text{ si } d \text{ est impair} \\ \\ \varphi(\mathbf{x};\alpha) = (d-2)f(\mathbf{x};\alpha), \text{ si } d \text{ est pair} \end{cases}$$

D'un autre côté, si nous considérons un rayon ρ différent de 1 puis on applique une distribution de vMF, nous devrions envisager des données normalisées ($\|\mathbf{x}\| = 1$), alors :

$$f(\mathbf{x};\alpha) = \frac{\xi^{\frac{d}{2}-1}\exp(\xi\mu^t\mathbf{x})}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\xi)}$$

on pose $\xi = 2\rho^2$ alors :

$$f(\mathbf{x};\alpha) = \frac{(2\rho^2)^{\frac{d}{2}-1}\exp(2\rho^2\mu^t\mathbf{x})}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(2\rho^2)} = \frac{(2\rho^2)^{\frac{d}{2}-1}\exp(2\rho\mu^t\rho\mathbf{x})}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(2\rho^2)}$$

$\rho\mu$ et $\rho\mathbf{x}$ sont les projections de μ et $\mathbf{x}$ sur l'hypersphère de rayon ρ. alors :

$$f(\mathbf{x};\alpha) = \frac{(2\rho^2)^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(2\rho^2)} \frac{\varphi(\rho\mathbf{x};\alpha)}{\gamma_\rho}.$$

Du calcul précédent, nous avons :

$$\gamma_\rho = \frac{\exp(2\rho^2)}{2a\rho\pi^{\frac{d}{2}} I_{\frac{d}{2}-1}(2\rho^2)},$$

$a = d-2$, si d est pair et $a = 1$ sinon ; alors :

$$\varphi(\mathbf{x};\alpha) = \frac{\exp(2\rho^2)\exp(2\mu^t\mathbf{x})}{2a\rho\pi^{\frac{d}{2}} I_{\frac{d}{2}-1}(2\rho^2)}$$

donc, pour $\rho \geq 1$, on obtient :

$$\varphi(\mathbf{x};\alpha) = \frac{\exp(2\rho^2)}{a\rho^{d-1}} f(\frac{\mathbf{x}}{\rho};(\frac{\mu}{\rho},2\rho^2))$$

Dans le cas où $d = 2$ et pour tout $\rho \geq 1$ on a :

$$\varphi(\mathbf{x};\alpha) = \frac{1}{\rho} f(\frac{\mathbf{x}}{\rho};\alpha)$$

on obtient donc :

$$\varphi(\mathbf{x};\mu) = \begin{cases} \frac{1}{\rho}\ f(\frac{\mathbf{x}}{\rho};(\frac{\mu}{\rho},2\rho^2)), & \text{si } d=2 \\ \frac{\exp(2\rho^2)}{\rho^{d-1}} f(\frac{\mathbf{x}}{\rho};(\frac{\mu}{\rho},2\rho^2)), & \text{si } d\geq 3 \text{ et d impaire} \\ \frac{\exp(2\rho^2)}{(d-2)\rho^{d-1}} f(\frac{\mathbf{x}}{\rho};(\frac{\mu}{\rho},2\rho^2)), & \text{si } d\geq 3 \text{ et d paire} \end{cases}$$

Ce dernier résultat nous a été très utile pour la simulation de certains tableaux à partir d'un modèle exponentiel (paragraphe 2.2), en utilisant le modèle vMF.

3 Expériences numériques

Dans nos expériences, nous nous concentrerons sur la qualité des classes. *Acc* est la précision de la classification, elle discerne une-à-une la relation entre les classes obtenues et les vraies classes. Elle donne la mesure dans laquelle chaque groupe contient des points de données de la classe correspondante ; elle est définie comme suit :

$$Acc = \frac{1}{N} max[\sum_{\mathcal{C}_k,\mathcal{L}_m} T(\mathcal{C}_k,\mathcal{L}_m)],$$

où $\mathcal{C}_k$ est la $k^{\text{ième}}$ classe dans le résultat final, et $\mathcal{L}_m$ est la vraie $m^{\text{ième}}$ classe. $T(\mathcal{C}_k,\mathcal{L}_m)$ est le nombre d'éléments qui appartiennent à la classe m et à la classe k à la fois. La précision calcule la somme maximale de $T(\mathcal{C}_k,\mathcal{L}_m)$ pour toutes les paires : partition-classes, ces paires n'ont pas de chevauchements. La partition de meilleure performance est celle qui est de plus grande précision. Grâce à *Acc*, nous comparons tous les algorithmes décrits sur deux type de données : données obtenues par simulations de Monte Carlo quatre et des ensembles de données d'expression génétique.

3.1 Données simulées

Dans nos expériences, nous avons fait varier la taille des données et le rayon des hypersphères des données. Nos simulations sont issus de mélanges de trois composants de

distributions de von Mises-Fisher. Nous avons évalué et comparé par la suite les algorithmes : SPK-RNS, EM_{vMF} et EM_{NEM_ρ}. On a simulé des tableaux suivant une taille de 60×100 et un rayon $\rho = 4$ et une taille de 60×500 et un rayon $\rho = 5.5$, en générant 20 échantillons chaque fois, pour chacun d'eux on applique les algorithmes précédant en initialisant EM_{vMF} et EM_{NEM_ρ} par la résultat de SPK-RNS. Les 20 $(1 - ACC)$ résultant sont présentés dans la figure 7.3.

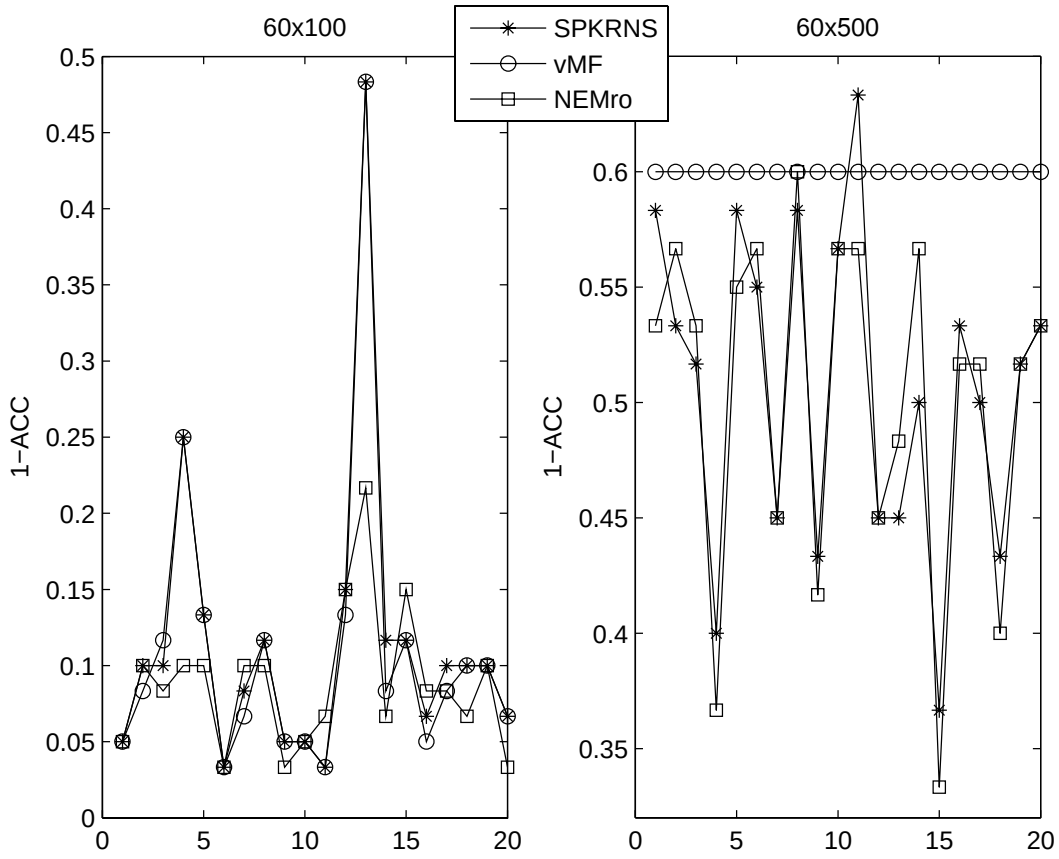

FIGURE 7.3 – Représentation des résultats de $(1 - ACC)$ obtenu par SPK-RNS, EM_{vMF} et EM_{NEM_ρ}.

Ensuite on a calculé les moyennes et écarts types des résultats de $(1 - ACC)$ pour chaque groupe de données et le temps d'exécution des expériences (table 7.3).

dans le tableau 7.3, en observant les moyennes de $(1 - ACC)$, nous pouvons voir que NEM_ρ donne de meilleures partitions. Les écarts-types indique que dans les premières expériences, les résultats de NEM_ρ sont plus proches de leur moyenne que les résultats de vMF. Par ailleurs, pour le deuxième groupe de simulations, où *std* de VMF est nul, nous pouvons voir (figure 7.3) que l'algorithme vMF n'a pas réussi à améliorer ses résultats, comme il a été influencé par les partitions initiales (résultats de SPK-RNS) qui n'étaient

TABLE 7.3 – Moyennes et écarts types des $(1 - Acc)$, rayons estimés ρ, résultats des algorithmes EM_{vMF} et EM_{NEM_ρ} pour les simulations Monte Carlo des deux groupes de données.

$n \times d$	ρ	SPK-RNS	vMF			NEM		
		$1 - ACC$	Time	$1 - ACC$	std	Time	$1 - ACC$	std
60×100	4	0.115	0.051	0.110	0.100	1.103	0.088	0.044
60×500	5.5	0.505	0.245	0.6	0	1.422	0.501	0.074

pas bonnes. A travers de nombreuses expériences, nous avons noté que les rayons estimés peuvent varier dans un grand intervalle de meilleures valeurs possibles, où les partitions qui en résultent sont plus proches de l'original.

3.2 Données génétiques

≪ Les données génétiques sont des données concernant les caractères héréditaires d'un individu ou d'un groupe d'individus apparentés ≫ [1]. L'étude des données génétiques s'avère l'un des sujets scientifiques les plus importants, plusieurs techniques développées pour répondre à cet intérêt. parmi lesquelles, la classification qui est une étape préliminaire et l'un des outils d'exploration de base d'investigation sur les données biopuces, elle peut même faire la découverte de nouveaux sous-types moléculaires. Une large gamme d'algorithmes de classification a été proposée dans ce domaine, y compris le regroupement hiérarchique (Eisen *et al.*, 1998), l'auto-organisation des cartes (SOM) (Tamayo *et al.*, 1999) ; le *k*means et ses variantes (Tseng, 2007) ; les algorithmes basées sur méthodes graphique ((Sharan et Shamir, 2000), (Xu *et al.*, 2001)) ; et les méthodes de regroupement basées sur les modèles de mélange (McLachlan *et al.*, 2002), (Ghosh et Chinnaiyan, 2002).

Nous avons choisi cinq échantillons de données d'expression génétiques, représentant une variété de jeux de données de tailles, dimensions et nombre de classes différents, il s'agit des données : Colon, Pediatric Acute Leukemia, Brain2, lung et Blood2, tous ces tableaux sont composés d'échantillons de tissus biologiques en ligne croisant les gènes en colonnes, tous détaillées ci-après.

- Les données Colon [2] sont un groupe de données qui contient 62 tissus de 2000 gènes, ce groupe est à l'origine catégorisé en deux classes de tissus du côlon, tumorales et normales.
- Les données Pediatric Acute Leukemia se composent de 327 échantillons de $\times 345$

1. http ://portal.unesco.org/fr/ev.php-URL$_ID = 17720\&URL_DO = DO_TOPIC\&URL_SECTION = 201.html$

2. http ://genomics-pubs.princeton.edu/oncology/affydata/ index.html

gènes réduit d'un montant initial de 12625 gènes, sélectionnés en utilisant la méthode de sélection par corrélation (Phuong et Vinh, 2008). Ce groupe est catégorisé en 7 classes : BCR-ABL, E2A-PBX1, Hyperdip50, MLL, T-ALL, TELL-AML1 en plus d'un autre groupe.

- Les données Brain2 [3], contiennent 24 échantillons croisant 1379 gènes, ce groupe est formé de 5 classes de cellule tumorale de cerveau : MD, Mglio, Rhab, Ncer et PNET.
- Les données Lung [4], sont un groupe de 203 échantillons croisant 1543 gènes, à l'origine, il est composé de 5 classes : AD, NL, SCLC, SQ et COID.
- Les données Blood2 [5], sont un groupe de 72 échantillons croisant 2194 gènes, composé de 3 classes : ALL, MLL, AML.

3.3 Stratégies

Pour chaque ensemble de données, nous avons procédé de la manière suivante :

- Chaque vecteur est normalisé pour appartenir à l'hypersphère unité.
- Classer les données en exécutant l'algorithme SPK-RNS (Vinh, 2008). Cet algorithme permet d'obtenir des partitions meilleures que l'algorithme SPK-means, en évitant au maximum les optimums locaux. Le SPK-means est exécuté un certain nombre de fois et initialisé d'une manière aléatoire (200 fois), puis on utilise une recherche à voisinage randomisé, pour affiner les résultats.
- Nous utilisons ensuite les classes obtenues par SPK-RNS comme une initialisation pour les autres algorithmes : EM_{vMF}, NEM avec $\rho = 1, \ldots, 300$ en gardant la meilleure partition comparée à l'origine (NEM_ρ, NCEM_ρ et NSEM_ρ).
  Pour les trois derniers algorithmes, le rayon est estimé en utilisant l'équation (7.1).

3.4 Résultats

Pour chaque ensemble de données, la taille et le nombre de classes sont affichés dans les tableaux 7.4, 7.5 et 7.6. Toutes les valeurs des rayons et les valeurs obtenues de $(1 - Acc)$ sont également signalées pour les six algorithmes. Les principales remarques découlant de nos expériences sont les suivantes :

- Dans le tableau 7.4, nous observons d'abord que l'algorithme NEM donne des résultats meilleurs que ceux de l'algorithme EM_{vMF}. Cependant, il doit balayer un intervalle

3. http ://algorithmics.molgen.mpg.de/Static/Supplements/CompCancer/Affymetrix/pomeroy-2002-v2/
4. http ://algorithmics.molgen.mpg.de/Static/Supplements/CompCancer/Affymetrix/gordon-2002/
5. http ://algorithmics.molgen.mpg.de/Static/Supplements/CompCancer/Affymetrix/armstrong-2002-v2/

considérable de valeurs du rayon, afin de classer un ensemble de données avec une durée de temps importante (dans notre cas 350× temps d'une seule exécution).

– Avec moins d'itérations, en calculant une estimation appropriée du rayon de l'hypersphère, l'algorithme NEM$_\rho$ a réussi le long de ces expériences à améliorer les résultats de l'algorithme de l'initialisation et de trouver des classes meilleures ou de même qualité que les autres algorithmes (voir tableau (7.5)). Signalons également que les algorithmes de type EM, EM$_{vMF}$, NEM et NEM$_\rho$ peuvent être affectés par l'initialisation.
– Sur les quatre ensembles de données, nous observons le bon comportement des différentes variantes de NEM$_\rho$ (Tableau 7.6), en particulier le NCEM$_\rho$.
– En conclusion, on remarque que l'algorithme NEM$_\rho$ a une meilleure performance avec les données ayant un nombre de lignes petit par rapport à la dimension.
NEM$_\rho$ et ses variantes ne donnent pas de bons résultats (voir tableau 7.6), quand le rayon estimé est hors de l'intervalle détecté par NEM, voir les données Leukemia et Lung, ainsi que pour NCEM$_\rho$ (voir tableau 7.6) qui est au minimum d'une qualité équivalente au SPK.

TABLE 7.4 – Résultats de $(1 - ACC)$ des algorithmes SPK-RNS, EM$_{vMF}$ et NEM avec l'intervalle de rayons correspondant.

Data		SPK-RNS	EM$_{vMF}$	NEM
ColData	ρ	1	1	$9 \ldots 17.29$
62×2000				
$g = 2$	$1 - Acc$	0.161290	0.129032	$0,09677$
Leukemia	ρ	1	1	4.35
327×345				
$g = 7$	$1 - Acc$	0.125382	0.1284	0.1193
Brain2	ρ	1	1	2.64
24×1379				
$g = 5$	$1 - Acc$	0.1904	0.1190	0.1428
Lung	ρ	1	1	$13.92 \ldots 18.70$
181×1626				
$g = 2$	$1 - Acc$	0.0497	0.1712	0.03867
Blood2	ρ	1	1	$3.1623 \ldots 18.70$
203×1543				
$g = 3$	$1 - Acc$	0.0277	0.0416	0.0277

TABLE 7.5 – Résultats de $1 - ACC$ des algorithmes NEM et NEM$_\rho$, avec l'intervalle de rayons correspondant.

Data		NEM	NEM$_\rho$
ColData	ρ	$9 \dots 17.29$	23.75
62×2000			
$g = 2$	$1 - Acc$	$0,096$	0.096
Leukemia	ρ	4.35	10.63
327×345			
$g = 7$	$1 - Acc$	0.1193	0.1284
Brain2	ρ	2.64	13.34
24×1379			
$g = 5$	$1 - Acc$	0.1428	0.1428
Lung	ρ	$13.92 \dots 18.70$	20.22
181×1626			
$g = 2$	$1 - Acc$	0.03867	0.0441
Blood2	ρ	$3.1623 \dots 18.70$	19.727
203×1543			
$g = 3$	$1 - Acc$	0.0277	0.0277

TABLE 7.6 – Résultats de $(1 - ACC)$ des algorithmes NEM$_\rho$, NCEM$_\rho$ et NSEM$_\rho$, avec l'intervalle de rayons correspondant.

Data		NEM$_\rho$	NCEM$_\rho$	NSEM$_\rho$
ColData	ρ	23.75	23.75	23.75
62×2000				
$g = 2$	$1 - Acc$	0.096	0.096	0.096
Leukemia	ρ	10.63	10.63	10.63
327×345				
$g = 7$	$1 - Acc$	0.1284	0.1253	0.1284
Brain2	ρ	13.34	13.34	13.34
24×1379				
$g = 5$	$1 - Acc$	0.1428	0.1904	0.1666
Lung	ρ	20.22	20.22	20.22
181×1626				
$g = 2$	$1 - Acc$	0.0441	0.0386	0.0497
Blood2	ρ	19.727	19.727	19.727
203×1543				
$g = 3$	$1 - Acc$	0.0277	0.0277	0.0277

4 Conclusion

Dans ce chapitre, nous avons présenté des preuves montrant qu'il est important pour les données directionnelles d'être normalisées de sorte que les données se trouvent sur une

hypersphère. Nous avons montré aussi qu'il est important de les normaliser avec un rayon plus grand que un, cette idée a permis à un modèle exponentiel simple de devenir un outil efficace de classification de données directionnelles.

Nous avons utilisé l'algorithme d'estimation-maximisation EM pour obtenir une formule de calcul général d'un rayon approprié ; cette formule a été expérimentée avec succès sur des échantillons de données.

Dans une comparaison théorique des deux algorithmes EM_{vMF} et NEM_{ρ}, on a établi un résultat important : à la convergence, les deux modèles fournissent des critères équivalents, dans le cas où $\xi = 2\rho^2$. Le changement de rayon n'influe pas sur un critère géométrique tel que le SPK, par contre pour un critère probabiliste, tel que le modèle exponentiel, la valeur du rayon est un facteur très important, contribuant à l'amélioration de la qualité de la partition.

Conclusions et perspectives

Ces dernières années, le développement général dans tous les domaines de la vie humaine et technologique a engendré un grand nombre de tableaux de données, dont un nombre important sont de grande taille. La collection instantanée des données a poussé les statisticiens et les utilisateurs à chercher à développer des compétences, pour servir au mieux les études et les interprétations des données brutes.

Parmi les domaines les plus demandés : l'analyse des données textuelles, les données génétiques, le traitement d'images et plusieurs autres domaines catalogués sous le répertoire de données directionnelles.

La classification de données directionnelles a imposé aux deux approches géométrique et probabiliste une sélection d'outils adaptés, par exemple le cosinus de Salton comme mesure de similarité et la distribution de von Mises-Fisher comme loi de probabilité, ce qui a motivé un grand nombre de chercheurs.

Notre travail porte sur l'étude de la problématique liée à la classification automatique de données directionnelles, en utilisant le modèle de mélanges de lois de von Mises-Fisher. Le principal apport de ce travail, concerne l'utilisation du modèle vMF, en proposant de nouvelles variantes et un nouveau modèle qui lui est comparable. Nous avons limité nos applications à deux principales types de données : les données textuelles et les données génétiques.

Notre travail se décompose en trois parties :

1. La première partie est consacrée à l'approche probabiliste, après une comparaison des deux algorithmes de type $k$means : le $k$means sphérique (SPK) et le $k$means axiale (KMA) employés dans la classification de données directionnelles, on a favorisé l'utilisation dans nos travaux du $k$means sphérique pour ses avantages par rapport au $k$means axiale.

2. La deuxième partie est un exposé détaillé de l'approche ML en utilisant le modèle de von Mises-Fisher, on a réussi a développer de nouvelles variantes de l'algorithme EM_{vMF}, il s'agit des algorithmes : CEM_{vMF}, SEM_{vMF} et SAEM_{vMF}. L'étude a été complétée par une évaluation du problème de recherche du nombre de classes

dans un mélange de lois de von Mises-Fisher. Une variété de critères d'informations ont été testés sous différentes tailles de données, en considérant différents degrés de mélange. Nous avons observé que certains d'entre eux tels que les critères Aic3, Aic, AICU et Bic ont été les plus intéressants. En outre, nous avons constaté que leur performance s'améliore avec l'augmentation de la taille des données et que les critères Aic3 et AICU apparaissent comme les meilleurs.

3. Dans la troisième partie de cette thèse, on a exploité un modèle exponentiel comparable au modèle von Mises-Fisher, ce dernier nécessite l'appartenance des données à une hypersphère de rayon $\rho \geq 1$. Nous avons présenté des preuves montrant qu'il est important pour les données directionnelles d'être normalisées, de sorte qu'ils se trouvent sur une hypersphère de rayon plus grand que un, cette idée a permis à un modèle exponentiel simple de devenir un outil efficace de classification de données directionnelles. Nous avons alors utilisé l'algorithme d'estimation-maximisation EM pour obtenir une formule de calcul général d'un rayon approprié ; cette formule a été expérimentée avec succès sur des données simulées et réelles.

Dans le domaine de la fouille de données directionnelles qu'on a exploré, les résultats des expériences numériques obtenus par les algorithmes et méthodes exploitées sont relativement encourageants. Notre outil a été implémenté en Matlab. Les futurs travaux pourraient concerner notamment les deux modèles explorés dans notre travail, mais dans un but de recherche des blocs homogènes (classification croisée) qui s'avère très enrichissante dans le contexte data mining.

Concernant le modèle exponentiel, nous pensons qu'il serait intéressant de discuter la possibilité d'estimer des rayons qui dépendent des classes, ce qui nous permettra probablement, de mieux comprendre nos données. Il serait aussi intéressant de s'attaquer à la fois aux problèmes de l'évaluation du nombre de classes combinés au choix de modèles parcimonieux ($[\pi_k, \xi]$, $[\pi_k, \xi]$ et $[\pi, \xi]$).

Publications

1. W. P. Bouberima, M. Nadif, et Y. K. Bencheikh. Clustering using EM and CEM, cluster number selection via the von Mises-Fisher mixture models. Int. J. Open Problems Compt. Math., Vol. 6, No. 1, March 2013

2. W. P. Bouberima, Y. K. Bencheikh et M. Nadif. Different variants of Normalized EM Algorithm for Gene Expression Data. The 2nd International Workshop on Biological Knowledge Discovery and Data Mining - DEXA '11. Toulouse, France, 1-2 Septembre, pp. 418-422, 2011

3. W. P. Bouberima, M. Nadif et Y. K. Bencheikh. Un algorithme EM normalisé pour les données directionnelles. 43éme Journées de Statistique de la Société Française de Statistique, Tunis, Tunisie, 23-27 Mai, pp., 2011.

4. W. P. Bouberima, M. Nadif et Y. K. Bencheikh. Choice of the model and the number of components in von Mises-Fisher mixtures. The 3^{rd} International Conference of the ERCIM (European Research Consortium for Informatics and Mathematics). Working Group on Computing & Statistics, Senate House, University of London, UK. 10-12 December, pp., 2010

5. W. P. Bouberima, M. Nadif et Y. K. Bencheikh. Assessing the Number of Clusters From a Mixture of Von Mises-Fisher. Journal : Lecture Notes in Engineering and Computer Science ISSN 2078-0958, Volume : 2185 ; Issue : 1, p. 2006. Certificat de mérite de meilleur article, world congress on engineering, London, U.K., 30 Juin- 2 Juillet, 2010.

6. W. P. Bouberima, M. Nadif et Y. K. Bencheikh. Classification de données directionnelles. Seizièmes rencontres de la Société francophone de classification (poster), Grenoble, France, 2-4 septembre, 2009.

7. W. Bouberima, M. Nadif et Y. K. Bencheikh. Etude Comparative entre les k-means axiales et le k-means sphérique sur des données textuelles. Journées de statistique

théorique et appliquée, USTHB Bab Ezzouar Alger, Algérie, 22-24 Novembre, 2008.

8. W. Bouberima, M. Nadif et Y. K. Bencheikh. Modèle de mélange Von Mises et classification automatique des données textuelles. Colloque international CISPA 2008, Constantine, Algérie, 18-19 Octobre, 2008.

Bibliographie

M. Abramowitz et I. A. Stegun : *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* New York : Dover, 1972.

H. Akaike : Information theory and an extension of maximum likelihood principle. *In Second International Symposium on Information Theory*, pages pp. 267–281. Akademiai Kiado, 1973.

K. J. Arnold : On spherical probability distributions, dissertation. Rapport technique, Massachusetts Institute of technology, 1941.

F. Ball et P. Blackwell : A finite form for the wrapped poisson distribution. *Adv. Appl. Probab.*, 24(49):pp. 221–222, 1992.

G. H. Ball et D. J.Hall : Isodata a novel method of data analysis and pattern classification. Rapport technique, standford research institute, menbo park calif. U.S.A., 1965.

G. H. Ball et D. J.Hall : A clustering technique for summarizing multivariate data. *Systems Research and Behavorial science*, 12(2):pp. 153–155, March 1967.

A. Banerjee, I. S. Dhillon, J. Ghosh et S. Sra : Clustering on the unit hypersphere using von mises-fisher distributions. *The Journal of Machine Learning Research*, 6:pp. 1345–1382, January 2005.

J. D. Banfield et A. E. Raftery : Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):pp. 803–821, 1993.

E. Batschelet : *Circular statistics in biology.* Academic Press, London, 1981.

R. J. Beran : Exponential models for directional data. *Ann. Statist.*, 7(6):pp. 1162–1178, 1979.

A. Berlinet et Ch. Roland : Parabolic acceleration of the em algorithm. *Stat. Comput.*, 19(1):pp. 35–47, 2009.

D. T. Best et N. I. Fisher : Efficient simulation of the von mises distribution. *Journal of the Royal Statistical Society. Applied Statistics*, 28(2):pp. 152–157, 1979.

J. C. Bezdek : *Pattern Recognition with Fuzzy Objective Function Algorithms.* Kluwer Academic Publishers Norwell, MA, USA, 1981.

C. Biernacki : Choix de modèles en classification. Thése de doctorat, Compiègne University of Technology, 1997.

C. Biernacki, G. Celeux et G. Govaert : Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):pp. 719–725, 2000.

C. Bingham et K. V. Mardia : A small circle distribution on the sphere. *Biometrika*, 65(2):pp. 379–389, 1978.

A. Blake et C. Marinos : Shape from texture estimation, isotopy and moments. *Artificial Intelligence*, 45:pp. 323–380, 1990.

W. P. Bouberima, Y. K. Bencheikh et M. Nadif : Different variants of normalized em algorithm for gene expression. *In Proceedings of the 22nd International Workshop on Database and Expert Systems Applications*, pages pp. 418–422. IEEE, Computer society, 2011.

B. Boulerice et G. R. Ducharme : Decentred directional data. *Ann. Inst. Statist. Math.*, 46:pp. 573–586, 1994.

H. Bozdogan : Model selection and akaikes's information criterion (aic) : The general theory and its analytical extensions. *Psycometrika*, 52(3):pp. 345–370, 1987.

H. Bozdogan : Mixture-model cluster analysis using model selection criteria and a new information measure of complexity. *In Proceedings of the first US/Japan conference on the Frontiers of Statistical Modeling : An Informational Approach*, pages pp. 69–113, Dordrecht, South Africa, 1994. Kluwer Academic Publishers.

M. Broniatowski, G. Celeux et J. Diebolt : *Reconnaissance de mélanges de densités par un algorithme d'apprentissage probabiliste.* In Diday E. et al., Data analysis and Informatics, 1983.

K. Bubna et C. V. Stewart : Model selection techniques and merging rules for range data segmentation algorithms. *Computer Vision and Image Understanding*, 80(2):pp. 215–245, 2000.

K. P. Burnham et D. R. Anderson : *Model Selection and Multimodel Inference : A Practical Information-Theoretic Approach.* Springer-Verlag, 2002.

G. Celeux : Classification et modéles. *Revue de statistique appliquée*, 36(4):pp. 43–58, 1988.

G. Celeux, D. Chauveau et J. Diebolt : A stochastic approximation type em algorithm for the mixture problem. Research report, INRIA, 1991.

G. Celeux, D. Chauveau et J. Diebolt : On stochastique version of th em algorithm. Research report, INRIA, 1995.

G. Celeux et D. Diebolt : *A probabilistic teacher algorithm for iterative maximum likelihood estimation. In Classification and Related Methods of Data Analysis.* H. H. Bock, ed., 1987.

G. Celeux et J. Diebolt : L'algorithme sem : un algorithme d'apprentissage probabiliste pour la reconnaissance de mélange de densités. *Revue de statistique appliquée*, 34(2):pp. 35–52, 1986.

G. Celeux et J. Diebolt : A simulated annealing type em algorithm. Research report, INRIA, 1989.

G. Celeux et J. Diebolt : A stochastic approximation type em algorithme for the mixture problem. *Computational Statistics & Data Analysis - Special issue on optimization*, 41 (1-2):pp. 119–134, 1992.

G. Celeux et G. Govaert : A classification em algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis - Special issue on optimization*, 14 (3):pp. 315–332, 1992.

Y. Chikuse : *Statistics on Special Manifolds.* Springer, 2003.

N. E. Day : Estimating the components of a mixture of normal distributions. *Biometrika*, 56(1):pp. 464–474, 1969.

J. P. Marques de Sá : *Applied Statistics : Using SPSS, STATISTICA, and MATLAB.* Springer, 2003.

G. D. Declan : *Structural Geology and Personal Computers.* Elsevier, 1996.

S. Dégerine : Lois de von mises et lois liées. *Ann. Inst. Henri Poincaré-Probab. Stat*, 15:pp. 63–77, 1979.

B. Delyon, M. Lavielle et E. Moulines : Convergence of a stochastic approximation version of the em algorithm. *The Annals of Statistics*, 27(1):pp. 94–28, 1999.

A. P. Dempster, N. M. Laird et D. B. Rubin : Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39:pp. 1–38, 1977.

I. Dhillon, I. S. Dhillon, Y. Guan et J. Kogan : Iterative clustering of high dimensional text data augmented by local search. *In the 2002 IEEE International Conference on Data Mining*, pages 131–138, Washington, USA, 2002. IEEE Computer Society.

I. S. Dhillon et D. S. Modha : Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):pp. 143–175, 2001.

I. S. Dhillon et S. Sra : Modeling data using directional distributions. Utcs technical report, 2003.

I.S. Dhillon, J. Fan et Y. Guan : *Data mining for scientific and engineering applications.* Kluwer academic publishers, 2001.

E. Diday : Nouvelles méthodes et nouveaux concepts en classification automatique et reconnaissance des formes. Thèse d'état université Paris 6, 1972.

E. Diday : *Optimisation en classification automatique.* INRIA, 1979.

J. L. Dortet-Bernadet et N. Wicker : Model-based clustering on the unit sphere with an illustration using gene expression profiles. *Biostatistics*, 9(1):pp. 66–80, 2007.

T. D. Downs : Orientation statistics. *Biometrika trust*, 59(3):pp. 665–676, 1972.

T. D. Downs et J. Liebman : Statistical methods for vectorcardiographic directions. *IEEE Trans. Bio-med. Eng*, 16:pp. 87–94, 1969.

S. T. Dumais et H. Chen : Hierarchical classification of web content. *In SIGIR '00 Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval*, pages pp. 256–263, New York, NY, USA, 2000. ACM Press.

J. Dunn : Well separated clusters and optimal fuzzy partitions. journal of cybernetics. *Journal of Cybernetics*, 4:pp. 95–104, 1974.

M. B. Eisen, P. T. Spellman P. O. Brown et D. Botstein : Cluster analysis and display of genome-wide expression patterns. *In Proceedings of the National Academy of Sciences of the United States of America*, 1998.

M. A. T. Figueiredo et A. K. Jain : Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and Machine Intelligence*, 24(3):pp. 381–396, 2002.

N. I. Fisher : *Statistical analysis of circular data.* Cambridge University Press, 1995.

N. I. Fisher, T. Lewis et B.J.J. Embleton : *Statistical analysis of spherical data*. Cambridge University Press, 1993.

R. A. Fisher : Dispersion on a sphere. *In Proc. Roy. Soc*, A217, pages pp. 295–305, London, UK, 1953. Roy. Soc.

E. W. Forgy : Cluster analysis of multivariate data : Efficiency vs. interpretability of classifications. *Biometrics*, 21:pp. 768–769, 1965.

J. Ghosh et A. M. Chinnaiyan : Mixture modelling of gene expression from microarray experiments. *Biometrics*, 18(2):pp. 275–286, 2002.

A. D. Gordon, P. E. Jupp et R. W. Byrne : Construction and assessment of metal maps. *British J. Math. Statist. Psych.*, 42:pp.169–182, 1989.

A. L. Gould : A regression technique for angular variates. *Biometrics*, 25:pp. 683–700, 1969.

G. Govaert : Classification avec distance adaptative. Thèse de doctorat de $3^{\text{ème}}$ cycle, Paris 6, 1975.

G. Govaert : Classification croisée. Thése d'état, Université de Paris 6, 1983.

G. Govaert : *Analyse des données. Traitement du signal et de l'image*. Hermes science, 2003.

G. Govaert et M. Nadif : Classification binaire et modèles. *Rev.Statistique Appliquées*, 38(1):pp.67–81, 1990.

I. Gradshteyn et I. Ryzhik : *Table of Integrals*. Elsevier, 2007.

R. J. Hathaway : Another interpretation of the em algorithm for mixture distribution. *Journal of Statistics & Probability Letters*, 4(2):pp. 53–56, 1986.

I. L. Hudson et M. R. Keatley : *Phenological Research*. Springer, 2010.

C. M. Hurvich et C. L. Tsai : Regression and time series model selection in small samples. *Biometrika*, 76(2):pp. 297–307, 1989.

S. R. Jammalamadaka et A. SenGupta : *Topics in Circular Statistics*. World Scientific, 2001.

R. E. Jensen : A dynamical programming algorithm for cluster data analysis. *J Oper Res Soc Amer*, 7:pp. 1034–1057, 1969.

F.-X. Jollois : Contribution de la classification automatique à la fouille de données. Thése de l'Université de Metz, 2003.

F. X. Jollois et M. Nadif : Speed-up for the expectation-maximization algorithm for clustering categorical data. *Journal of Global Optimization*, 37(4):pp. 513–525, 2007.

P. E. Jupp : Some applications of directional statistics to astronomy. *In the 5th Conference on Multivariate Statistics and Matrices in Statistics*, pages pp. 123–133. E. M. Tiit and T. Kollo and H. Niemi, 1995.

M. Kearns, Y. Mansour et A. Ng : An information-theoretic analysis of hard and soft assignment methods for clustering. *In the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages pp. 282–293. Morgan Kaufmann, 1997.

J. T. Kent : The fisher-bingham distribution on the sphere. *J. Royal. Stat. Soc.*, 44 (1):pp. 71–80, 1982.

J. T. Kent et K. V. Mardia : Consistency of procrustes estimators. *J. Royal. Stat. Soc. Ser. B*, 59(1):pp. 281–290, 1997.

W. Kuhn et F. Grün : Beziehungen zwischen elastichen konstanten and dehnungs doppelbrechung hochelastischer stoffe. *Kolloid. Z.*, 101:pp. 248–271, 1942.

A. Kume et S. G. Walker : On the fisher-bingham distribution. *Statistics and Computing*, 19(2):pp. 167–172, 2009.

F. Lagona et M Picone : Maximum likelihood estimation of bivariate circular hidden markov models from incomplete data. *Journal of Statistical Computation and Simulation*, pages pp. 1–15, 2012a.

F. Lagona et M Picone : Model-based clustering of multivariate skew data with circular components and missing values. *Journal of Applied Statistics*, 39:pp. 927–945, 2012b.

G. N. Lance et W. T. Williams : A general theory of classificatory sorting strategies. 1. hierarchical systems. *Computer Journa*, 9(4):pp. 373–380, 1967.

L. Lebart et A. Salem : *Statistique Textuelle*. Dunod, 1994.

A. Lelu : Modéles neuronaux pour l'analyse de données documentaires et textuelles. Thése de l'Universitéé de Paris 6, 1993.

C. Liu, D. B. Rubin et Y. N. Wu : Parameter expansion to accelerate em : The px-em algorithm. *Biometrika*, 85:pp. 755–770, 1998.

K. Mardia, C. Taylor et G. Subramaniam : Protein bioinformatics and mixtures of bivariate von mises distributions for angular data. *Biometrics*, 63:pp. 505–512, 2007.

K. V. Mardia : *Statistics of Directional Data*. Academic Press, New York, 1972.

K. V. Mardia : Characterization of directional distributions. *Statistical Distributions in Scientific Work (Characterizations and Applications)*, 3(2):pp. 365–386, 1974.

K. V. Mardia : Statistics of directional data. *J. Roy. Statist. Soc. Ser. B*, 37(3):pp. 349–393, 1975.

K. V. Mardia, A. Baczkowski, X. Feng et T. J. Hainsworth : Statistical methods for automatic interpretation of digitally scanned finger prints. *Pattern Recognition Lett.*, 18:pp. 1197–1203, 1997.

K. V. Mardia et P. E. Jupp : *Directional Statistics*. John Wiley & Sons, 2009.

K. V. Mardia, J. T. Kent, C. R. Goodall et J. A. Little : Kriging and splines with derivative information. *Biometrika*, 83:pp. 207–221, 1996.

K.V. Mardia, K.V. Hughes, C.C. Taylor et H. Singh : A multivariate von mises distribution with applications to bioinformatics. *Pattern Recognition Lett.*, 36:pp. 99–109, 2008.

G. McLachlan, R. Bean et D. Peel : A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18:pp. 413–422, 2002.

G. McLachlan et T. Krishnan : *The EM algorithm and extensions*. John Wiley & Sons, 1997.

G. J. McLachlan et D. Peel : *Finite mixture models*. Wiley, 2000.

A. McQuarrie, R. Shumway et C. L. Tsai : The model selection criterion aicu. *Statistics & Probability Letters*, 34:pp. 285–292, 1997.

J. B. McQueen : Some methods for classification and analysis of multivariate observations. *In the 5th Berkeley symposium on math. Statistics and probability*, pages 281–297, California, USA, 1967. Univ. of Calif. Press.

M. Nadif et G. Govaert : Clustering for binary data and mixture models : Choice of the model. *Applied Stochastic Models and Data Analysis*, vol. 13:pp. 269–278, 1998.

R. Neal et G. Hinton : A view of the em algorithm that justifies incremental sparse, and other variants. *Kluwer Academic Publishers, Dordrecht*, 1998.

N. M. Phuong et N. X. Vinh : Normalized em algorithm for tumor clustering using gene expression data. *In the 8th IEEE International conference on Bioinformatics and BioEngineering*, pages pp. 315–320, Darlinghurst, Australia, Australia, 2008. Australian Computer Society, Inc.

G. Pólya : Zur statistik der spharischen verteilung der fixsterne. *Ast. Nachr.*, 208:pp. 175–180, 1919.

T. M. Pukkila et C. R. Rao : Pattern recognition based on scale invariant discriminant functions. *Inform. Sci.*, 45:pp. 379–389, 1988.

C. R. Rao : *Linear Statistical Inference and its Applications*. Wiley, 1973.

L. Rayleigh : On the problem of random vibrations, and of randomflights in one, two, or three dimensions. *Phil. Mag.*, 37(6):pp. 321–347, 1919.

S. Régnier : Sur quelques aspects mathématiques des problèmes de classification automatique. *ICC bulletin*, 4:pp. 175–191, 1965.

J. Rissanen : Modelling by shortest data description. *Automatica*, 14:pp. 465–471, 1978.

E. Ronchetti : Robust model selection in regression. *Statistics and Probability Letters*, 3(1):pp. 21–23, 1985.

E. M. Ruspini : A new approch to clustering. *Information and control*, 15(1):pp. 22–32, 1969.

G. Salton et C. Buckley : Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 4(5):pp.513–532, 1988.

G. Salton et M. J. McGil : *Introduction to Modern Retrieval*. McGraw-Hill Book Company, 1983.

G. Saporta : *Probabilité, Analyse des données et Statistique*. Technip, 2006.

A. Schwarz : Reconnaissance des composants d'un mélange. Thèse de doctorat de 3$^{\text{ème}}$ cycle, Paris 6, 1974.

G. Schwarz : Estimating the dimension of a model. *The Annals of Statistics*, 6(2):pp. 461–464, 1978.

R. Sharan et R. Shamir : Click : a clustering algorithm with applications to gene expression analysis. *In Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology(ISMB)*, pages pp.307–316. AAAI Press, 2000.

P. Shi et C. L. Tsai : Random walk on a circle. *Biometrika*, 50:pp. 385–390, 1963.

P. Shi et C. L. Tsai : A note on the unification of the akaike information criterion. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(3):pp. 551–558, 1998.

P. H. Sneath : Computers in taxonomyl. *J. gen. Microbiol.*, 17(1):pp. 201–226, 1957.

R. R. Sokal et P. H. Sneath : *principales of numerical taxonomy.* San Francisco : Freeman, 1963.

M. A. Stephens : The statistics of directions : the von mises and fisher distributions. Thesis, University of toronto, 1962.

N. Stergiou : *Innovative Analyses of Human Movement.* Human Kinetics, 2004.

N. Sugiura : Further analysis of the data by akaike's information criterion and the finite corrections. *Comm. Statist.*, 7:pp. 13–26, 1978.

P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander et T. Golub : Interpreting patterns of gene expression with self-organizing maps : methods and application to hematopoietic differentiation. *In Proceedings of the National Academy of Sciences of the United States of America*, pages pp. 2097–2912, 1999.

B. Thiesson, C. Meek, et D. Heckerman : Accelerating em for large databases. Msr-tr-99-31, microsoft researcht, 2001.

G. Tseng : Penalized and weighted k-means for clustering with scattered objects and prior information in high-throughput biological data. *Bioinformatics*, 23(17):pp. 2247–2255, 2007.

G. Ulrich : Computer generation of distributions on the m-sphere. *Appl. Statist.*, 33:pp. 158–163, 1984.

G. J. G. Upton et B. Fingleton : *Spatial Data Analysis by Example, Volume 2 : Categorical and Directional Data.* John Wiley, 1989.

N. X. Vinh : Gene clustering on the unit hypersphere with the spherical k-means algorithm : coping with extremely large number of local optima. *In The 2008 international conference on bioinformatics and computational biology*, 2008.

R. von Mises : Ueber die "ganzzaligkeit" der atomgewicht und verwandte fragen. *Physikal*, 19:pp. 490–500, 1918.

T. H. Waterman : The analysis of spatial orientation. *Ergeb. Biol.*, 26:pp. 97–117, 1963.

G. S. Watson : Orientation statistics in the earth sciences. *Bul. Geol. Inst. Univ. Uppsale*, 2:pp. 73–89, 1970.

G. S. Watson : *Statistics on Spheres, The University of Arkansas lecture notes in the mathematical sciences*. John Wiley & Sons, 1983.

A. Wintner : On the shape of the angular case of cauchy's distribution curves. *Ann. Math. Statist.*, 18(4):pp. 589–593, 1947.

J. H. Wolf : Pattern clustering by multivariate mixture analysis. *Multivar. Behavior. Res.*, 5:pp. 329–350, 1970.

M. A. Wong : A hybrid clustering method for identifying high-density clusters. *Journal of the American Statistical Association*, 77:pp. 841–847, 1982.

A. T. A. Wood : A bimodal distribution for the sphere. *Appl. Statist.*, 31(1):pp. 52–58, 1982.

A. T. A. Wood : Some notes on fisher-bingham family on the sphere. *Comm. Statist. Theory Methods*, 17(11):pp. 3881–3897, 1988.

A. T. A. Wood : Simulation of the von-mises distribution. communications of statistics. *Comm. Statist. Simulation and Computation*, 23(1):pp. 157–164, 1994.

C. F. Jeff Wu : On the convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):pp. 95–103, 1983.

Y. Xu, V. Olman et D. Xu : Clustering gene expression data using a graphtheoretic approach : an application of minimum spanning trees. *Bioinformatics*, 17(4):pp. 309–318, 2001.

L. A. Zadeh : Fuzzy sets. *Information and Control*, 8:pp. 338–353, 1965.

S. Zhong : Efficient online spherical k-means clustering. *In IEEE Int. Joint Conf. Neural Networks (IJCNN 2005)*, pages 3180–3185, 2005.