Université d'Evry Val d'Essonne

Thèse

présentée par

Camille Brunet

pour obtenir le grade de

Docteur de l'université d'Evry Val-d'Essonne

en Mathématiques Appliquées (Statistiques)

SPARSE AND DISCRIMINATIVE CLUSTERING FOR COMPLEX DATA: Application to cytology

Soutenance prévue le 1er décembre 2011

Composition du jury:

Badran Fouad	Rapporteur
Besse Philippe	Examinateur
Biernacki Christophe	Rapporteur
Comon Pierre	Examinateur
D'Alché-Buc Florence	Examinateur
Peltier Eric	Examinateur
Vigneron Vincent	Directeur de thèse

Contents

Chapter 1

Introduction

1.1 Classification et visualisation des données modernes

Depuis une dizaine d'années, les évolutions de la technique et des technologies ont participé à l'augmentation des capacités de mesures et de stockage de l'information. Cette évolution a touché l'ensemble des domaines applicatifs tels que la biologie, la santé, l'économie ou l'informatique par exemple. Les données générées s'avèrent être de plus en plus complexes et elles présentent, en particulier, la spécificité d'être de grande dimension, du fait du nombre croissant de leurs variables descriptives. Leur traitement nécessite donc l'usage de procédures automatiques qui permettent de fournir une représentation simple des données par partitionnement ou par visualisation. La grande dimension des données peut apparaître comme bénéfique dans la tâche de classification puisqu'elle offre un grand panel de variables détaillant l'objet à analyser. Cependant, il s'avère que, généralement, seul un sous-ensemble de ces variables est nécessaire pour différencier des groupes de données; les variables restantes n'apportant aucune information supplémentaire et pouvant nuire à la classification des données et à leur visualisation. Il apparaît donc nécessaire de réduire la dimension des observations de sorte à faciliter et à contribuer à leur classification et à leur visualisation.

Puisque la dimension des observations est plus grande que leur dimension intrinsèque, il est théoriquement possible de réduire la dimension de l'espace d'origine sans perdre d'information. Il existe moult méthodes de réduction de dimension qui sont traditionnellement exécutées avant une étape de classification. De manière extrêmement classique, on peut penser à l'analyse en composante principale qui reste la méthode la plus utilisée dans le cadre de méthodes linéaires d'extraction de caractéristiques ou encore à des méthodes non-linéaire dont la plupart a été résumée et comparée dans van der Maaten *et al.* [169], mais aussi des méthodes de sélection de variables dont un aperçu global est disponible dans Guyon et Ellisseeff [77]. Cependant, l'une des principales limites de ces approches réside dans le fait qu'elles ne considèrent pas la tâche de partitionnement des données et peuvent donc dégrader les performances du clustering. En effet, les méthodes de réduction de dimension opérées indépendamment d'une procédure de

partitionnement peuvent impliquer une perte d'information qui pourrait être discriminante dans la tache de clustering. Il existe cependant dans la littérature quelques approches qui combinent réduction de dimension et classification automatique. En particulier, des méthodes de sélection de variables ont été développées dans le cadre de la classification automatique par modèle de mélanges tels que, en particulier, les travaux de Law *et al.* [111], Raftery et Dean [148], ainsi que plus récemment ceux de Maugis *et al.* [121]. En outre, il existe d'autres méthodes basées sur le modèle de mélanges gaussiens, et communément appelées méthodes de clustering dans des sous-espaces. En particulier, elles modélisent les groupes de données dans des sous-espaces de petite dimension qui leur est propre. L'ensemble de ces méthodes présentent de bonnes performances de classification mais ne permettent pas cependant de visualiser les données.

Le thème de ce manuscript est la classification automatique et la représentation parcimonieuse de données de grande dimension.

1.2 Contribution

Un premier travail se place dans le contexte de la classification non-supervisée par modèle de mélanges gaussiens. Il est motivé par le fait que le clustering et la visualisation de données de grandes dimensions restent deux enjeux récurrents et actuels de statistiques qui sont confrontés à des problèmes calculatoires ainsi qu'à des difficultés d'interprétation. Il existe différentes manières de gérer ces problèmes et en particulier des approches, combinant la réduction de dimension et la classification non supervisée, basées sur la recherche de sous-espaces propres aux classes, tels que les travaux de Bouveyron *et al.* [22], McLachlan *et al.* [128] ou encore ceux de McNicholas et Murphy [130]. Cependant, malgré les très bonnes performances de ces nouvelles approches, une interprétation de la partition obtenue ainsi qu'une visualisation informative des clusters résultants de la classification restent difficiles.

Pour faire face à ces problèmes, nous proposons une méthode probabiliste et un algorithme de type EM, appelé Fisher-EM, qui permet simultanément de classer et de visualiser des données dans un contexte de classification non supervisée. Cette approche se base sur une modélisation des groupes par modèles de mélange dans un sous-espace latent discriminant de petite dimension lequel est estimé par l'intermédiaire d'un critère basé sur la théorie de Fisher [54]. A cet effet, nous avons introduit le modèle de mélange latent discriminant, appelé le modèle DLM, qui modélise les données de manière parcimonieuse afin de générer une partition et une visualisation discriminantes. Ce modèle se base sur l'idée qu'il existe un sous-espace latent discriminant qui est commun aux groupes, de dimension intrinsèque plus petite que la dimension des observations et pour lequel au plus $K-1$ dimensions sont théoriquement suffisantes pour discriminer K groupes. En imposant des contraintes sur les matrices de variances-covariances des K groupes, nous avons décliné une famille de 12 modèles DLM qui présente l'avantage d'être de faible complexité par rapport à des méthodes comparables telles

que les méthodes de clustering dans les sous-espaces [22, 129, 130] ou encore telles que celles développées dans le cadre des modèles de mélanges factoriels parcimonieux [8, 136, 188, 189]. Nous avons développé un algorithme appelé Fisher-EM qui alterne trois étapes: une étape E calcule à chaque itération l'espérance de la log vraisemblance complétée conditionnellement à la valeur courante du paramètre. Puis, une étape F estime la matrice de projection dont les colonnes engendrent le sous-espace latent discriminant de dimension d bornée à $K-1$ et dans lequel les K groupes sont au mieux séparés. Nous avons pour cela adapté le problème de maximisation du critère traditionnel de Fisher, classiquement utilisé dans un contexte supervisé, à un contexte non supervisé sous la contrainte d'orthogonalité. Pour cette étape, nous avons développé trois manières différentes d'estimation de la matrice de projection: une première approche utilise une procédure de type Gram-Schmidt qui permet de tenir compte de l'orthogonalité des colonnes de la matrice de projection. Une deuxième alternative réécrit le problème d'optimisation de l'étape F comme un problème de régression puis nous l'avons reformulé de telle manière que la solution puisse être approximée par une décomposition en valeurs singulières. Enfin, la troisième étape de l'algorithme Fisher-EM, l'étape M, estime les paramètres du modèle DLM en maximisant l'espérance conditionnelle de la log-vraisemblance complétée. Bien que la convergence de l'algorithme Fisher-EM n'est *a priori* pas garantie, nous montrons que cet algorithme est un algorithme de type EM dans le cas isotropique, lui assurant ainsi la convergence vers un maximum local de la vraisemblance.

Une des principales hypothèses du modèle DLM est la relation linéaire entre l'espace des observations et l'espace latent discriminant. Les variables latentes qui définissent cet espace discriminant de petite dimension, résultent d'une combinaison linéaire des variables d'origine. Ceci pose deux problèmes différents: le premier est lié à l'interprétation des axes discriminants par rapport aux variables d'origine; le second, est relatif à la présence de variables initiales sans intérêts, ou dites de "bruit", dans les axes, pouvant engendrer une détérioration des résultats de clustering d'une part et de la visualisation des données d'autre part. Pour pallier ces problèmes, nous proposons également d'introduire de la parcimonie dans les axes discriminants estimés. La réécriture du problème d'optimisation du critère de Fisher en un problème de type régression, nous permet de considérer le cas de la régression linéaire pénalisée type LASSO [163], rendant alors possible l'introduction de parcimonie dans les composantes des axes estimés. Nous proposons de ce fait trois versions pénalisées de l'algorithme Fisher-EM. En plus de produire des axes parcimonieux, ce terme de pénalité permet de faire de la sélection de variables discriminantes.

Dans un second travail, nous nous sommes intéressés à la détermination du nombre de groupes en utilisant le cadre de la sériation. Un des problèmes récurrents, en analyse de données dans un cadre non probabiliste, est de déterminer des structures, des relations entre les observations. La sériation, qui est une technique d'analyse exploratoire très ancienne, offre cette perspective puisqu'elle se base sur la recherche d'un ordre parmi les observations, de

telle sorte que les éléments adjacents soient les plus similaires entre eux. Elle permet, de plus, une visualisation des groupes, puisqu'elle agit directement sur le tableau de valeurs des observations, par permutations successives des lignes et des colonnes, et permet de révéler une structure par blocs. De nombreux travaux ont été effectués au cours du XXème siècle, et on peut citer en particulier les travaux de Marchotorino [118], VanMichelen [131] et Liiv [112], qui selon leur époque, ont donné un aperçu global des méthodes existantes.

La sériation présente de nombreux avantages de visualisation mais dès lors que les données sont bruitées ou que les groupes se superposent, la visualisation de toute structure devient difficile. Pour remédier à ce problème, nous proposons d'intégrer de la parcimonie dans les données par l'intermédiaire d'une famille de matrices binaires. Ces dernière sont construites à partir d'une mesure de dissimilarité basée sur le nombre de voisins communs entre paires d'observations. En particulier, plus le nombre de voisins communs imposé est important, plus la matrice sera parcimonieuse, *i.e.* remplie de zéros, ce qui permet, à mesure que le seuil de parcimonie augmente, de retirer les valeurs extrêmes et les données bruitées. Cette collection de matrices parcimonieuses est ordonnée selon un algorithme de sériation de type *forward stepwise,* nommé PB-Clus, afin d'obtenir des représentations par blocs des matrices sériées. La sélection du niveau de parcimonie est faite par un critère de compacité calculée à partir de la famille de matrices ordonnées et favorise une représentation diagonale par blocs *i.e.* celle qui révèle la plus distinctement la structure intrinsèque des données.

Enfin, ces deux travaux, bien que très différents, ont été appliqués à des données cytologiques fournies par la société Novacyt. La base de données se compose d'échantillons de cellules issues du col de l'utérus de 13 femmes différentes, et décrites par une quarantaine de caractéristiques morphologiques et texturées. L'objectif de l'application est de sélectionner un sous-ensemble de variables permettant de discriminer les cellules pathologiques des autres objets contenus dans les échantillons. En particulier, la technique de sériation a été mise en oeuvre pour visualiser la structure intrinsèque des données et mettre en exergue une structure principale à 2 groupes, tandis que l'algorithme Fisher-EM pénalisé a été utilisé pour sélectionner les variables discriminantes.

1.3 Organisation de la thèse

Cette thèse est divisée en trois parties distinctes: les deux premières pouvant être lues indépendamment l'une de l'autre et la troisième partie est une application des deux méthodes exposées dans les parties I. et II., au domaine de la cytologie.

Partie I. Subspace clustering in a Gaussian Mixture model

Cette première partie rend compte d'un travail réalisé avec Charles Bouveyron (Université Paris 1 Panthéon-Sorbonne), sur la thématique des modèles de mélanges dans des sous-espaces.

Chapitre 2: Ce premier chapitre introduit de manière générale les modèles de mélanges gaussiens: les approches les plus traditionnelles jusqu'aux plus récentes privilégiant la parcimonie, seront exposées. L'algorithme EM et ses différentes variantes y seront introduits. Par ailleurs, comme le modèle que nous proposons dans cette première partie a la particularité de modéliser les données dans un sous-espace latent discriminant, nous introduirons cette notion dans un contexte supervisé à travers l'analyse discriminante de Fisher. Puisque nombre d'auteurs ont travaillé sur cette approche, certes très ancienne, afin d'étendre et d'améliorer les travaux initiaux de Fisher [54], certains d'entre eux seront exposés plus en détail dans un deuxième paragraphe. Enfin, puisque notre approche s'inscrit dans la famille des méthodes de clustering dans des sous-espaces, nous introduirons succinctement les deux travaux majeurs dans ce domaine, à savoir les familles de modèles à facteurs introduits par Ghahramani [64] et McLachlan [127] puis étendus, en particulier, par les travaux de McNicholas et Murphy [129, 130], d'une part, et la famille de modèles gaussiens introduits par Bouveyron *et al.* [22, 23] d'autre part.

Chapitre 3: Ce deuxième chapitre introduit un modèle de mélanges, appelé modèle de mélanges latent discriminant (*Discriminative Latent Mixture model*), le modèle DLM, qui a comme objectif de modéliser les données de manière parcimonieuse afin d'en générer une partition et une visualisation discriminantes. Plus particulièrement, ce modèle se base sur l'idée qu'il existe un sous-espace latent de dimension intrinsèque plus petite que la dimension des observations, pour lequel au plus $K-1$ dimensions sont théoriquement suffisantes pour discriminer K groupes. Cette approche nous permet d'introduire une famille de 12 modèles en imposant des contraintes sur les matrices de variances covariances. Ces modèles ont l'avantage d'être extrêmement parcimonieux comparativement à la famille de 28 modèles proposée parallèlement par Fraley et Raftery [59] et Govaert et Celeux [36], ou à des approches similaires basées sur des méthodes dans des sous-espaces telles que celles proposées par Bouveyron *et al.* [22] ou McNicholas et Murphy [129, 130] par exemple. Dû au fait que le sous-espace latent discriminant est commun aux classes et de dimension intrinsèque bornée à $K-1$, la complexité des modèles DLM reste faible et de même ordre que les récentes approches de modèles de mélanges factoriels parcimonieux de Yoshida *et al.* [188, 189], Baek et McLachlan [8] ou encore de Montanari et Viroli [136].

Chapitre 4: Dans un contexte non supervisé, la maximisation directe de l'espérance de la log-vraisemblance n'étant pas faisable, nous avons donc utilisé une procédure itérative de type EM. Cependant, contrairement aux approches classiques de modèles de mélanges dans des sous-espaces pour lesquelles le sous-espace est estimé par maximum de vraisemblance, nous cherchons à estimer un espace latent de petite dimension, certes, mais qui est, de plus, discriminant. Afin de tenir compte de cette spécificité, nous avons introduit un algorithme appelé Fisher-EM qui alterne trois étapes: une étape E qui calcule à chaque itération, l'espérance de la log vraisemblance complétée conditionnellement à la valeur courante du paramètre; une

étape F qui détermine la transformation linéaire $U \in \mathbb{R}^{p\times d}$ (avec $d < p$) relative à la base du sous-espace latent de dimension $d \leq K-1$ dans lequel les K groupes sont le mieux séparés. Pour cela, nous avons adapté le problème de maximisation du traditionnel critère de Fisher, habituellement utilisé dans un contexte supervisé, à un contexte non supervisé sous la contrainte d'orthogonalité et conditionnellement à la partition courante des données. De plus, nous proposons trois manières différentes d'estimation de la matrice de projection. Enfin une étape M, traditionnelle à l'algorithme EM, estime les paramètres du modèle DLM en maximisant l'espérance conditionnelle de la log-vraisemblance complétée. Cependant, l'ajout de l'étape F dans l'algorithme EM ne garantit pas *a priori* la convergence de l'algorithme. A cet effet, nous montrons que l'algorithme Fisher-EM, dans le cas isotropique, est un algorithme EM assurant ainsi de bonnes propriétés, notamment de convergence. Enfin, ce chapitre se termine sur quelques considérations pratiques et numériques de l'algorithme Fisher-EM.

Chapitre 5: Ce chapitre met en exergue l'intérêt de notre approche sur des données simulées et sur des données réelles.

La majeure partie des éléments de ces trois premiers chapitres ont fait l'objet d'une publication dans le journal *Statistics and Computing,* référée comme:
Bouveyron C., Brunet C., *Simultaneous model-based clustering and visualization in the Fisher discriminative subspace*, Statistics and Computing, 2011 (in Press).

Une première version de l'algorithme Fisher-EM est disponible sur le site de R CRAN (paquet FisherEM).

Chapitre 6: Ce dernier chapitre concernant le modèle DLM, aborde la question de l'interprétation des axes. En effet, une des principales hypothèses de notre modèle est la relation de linéarité entre l'espace des observations et l'espace latent discriminant. Les variables latentes, qui définissent cet espace discriminant de petite dimension, résultent d'une combinaison linéaire des variables d'origine. Ceci pose deux problèmes: le premier est lié à l'interprétation des axes discriminants par rapport aux variables d'origine puisque l'importance du coefficient, associé à la variable d'origine, ne suffit pas à déterminer son caractère discriminant; le second est relatif à la présence de variables d'origine sans intérêts ou dites de "bruit" dans les axes. En effet, pour cette deuxième situation, le fait que les axes soient des combinaisons linéaires des variables d'origine implique que les variables non-informatives restent présentes dans ceux-ci, pouvant engendrer une détérioration des résultats de clustering d'une part et de la visualisation des données d'autre part. Afin de pallier ce double problème, nous proposons d'introduire de la parcimonie dans l'estimation des axes au moyen d'une pénalité ℓ_1. A partir des trois approches d'estimation de la matrice de projection de l'étape F dans l'algorithme Fisher-EM, nous avons décliné trois procédures pénalisées qui permettent d'estimer avec parcimonie les axes. Cette approche a une double fonction: outre qu'elle facilite l'interprétation et améliore les résultats de clustering pour certaines données, elle joue le rôle

d'une méthode de sélection de variables discriminantes pour la classification. Des expériences sur données simulées et réelles viennent illustrer l'intérêt d'une telle approche.

Partie II. Seriation of a collection of parsimonious matrices

Cette partie est le fruit d'un travail effectué avec Vincent Vigneron (Université d'Evry) et Thomas Villmann (Université de sciences appliquées de Mittweida - Allemagne).

Chapitre 7: Ce chapitre présente, de manière générale, le problème de la sériation. Il se pose dans la définition et l'évaluation de la meilleure permutation possible entre les lignes et les colonnes d'une matrice et se traduit comme un problème d'optimisation. En particulier, l'objectif de la sériation est de trouver la fonction de permutation optimale qui optimise un certain critère de rangement. Les critères traditionnels développés pour la sériation sont présentés dans une première partie ainsi que les différentes approches algorithmiques. Ce critère de rangement pour la sériation repose essentiellement sur la similarité (ou dissimilarité) des paires d'objets à ordonner, c'est pourquoi un paragraphe sera dédié à différentes mesures de dissimilarités. Enfin, le chapitre donnera un rapide aperçu des méthodes de clustering par blocs qui ont été développées dans la littérature et en particulier, celles définies à travers un modèle probabiliste.

Chapitre 8: Dans ce chapitre, nous introduisons une mesure de dissimilarité basée sur la notion de voisinage commun entre paires d'observations. A partir de cette matrice de voisins communs, nous proposons d'y intégrer différents niveaux de parcimonie créant ainsi une collection de matrices binaires. Dans notre approche, le degré de voisinage est défini comme une valeur "seuil" du nombre de voisins communs entre paires d'observations en deçà de laquelle les paires d'observations sont éliminées. Ainsi, plus le nombre de voisins communs imposé est important et plus la matrice sera parcimonieuse, *i.e.* remplie de zéros: à mesure que le seuil de parcimonie augmente, les valeurs extrêmes et les données bruitées sont, de manière structurelle, retirées de l'échantillon considéré. La collection de matrices parcimonieuses est ordonnée selon un algorithme de type *forward-stepwise* nommé PB-Clus et la sélection du niveau de parcimonie, permettant de révéler la structure intrinsèque des données, est faite à partir d'un critère de compacité. Ce critère sélectionne la matrice binaire réordonnée qui permet une visualisation claire, au sens de "diagonale par blocs", de la structure des données. Enfin, différentes considérations numériques et algorithmiques termineront ce chapitre.

Chapitre 9: Ce dernier chapitre met en oeuvre l'algorithme PB-Clus sur des simulations et sur données réelles.

Certains éléments des chapitres de cette deuxième partie ont fait l'objet d'une publication dans la revue française des nouvelles technologies (RNTI), référée comme:

Brunet C., Villman T.,Vigneron V., *Une famille de matrices sparses pour une modélisation multi-échelle par blocs*, Revue des Nouvelles Technologies de l'Information, 2011 (in Press).

Partie III. Application to cervical cancer detection

Cette dernière partie est une application des deux approches introduites précédemment, à savoir la méthode de classification par modèle de mélanges ainsi que celle par sériation, à des données cytologiques fournies par l'entreprise Novacyt.

Chapitre 10: Dans ce chapitre, nous disposons de données fournies par l'entreprise Novacyt qui représentent des échantillons de cellules issues du col de l'utérus provenant de frottis différents.

Chaque cellule est décrite par 42 caractéristiques morphologiques et texturées. L'ambition de cette application est de déterminer un ensemble de caractéristiques qui permettent de discriminer la classe des cellules saines de celle des cellules anormales. En effet, de manière générale, les performances d'un classifieur dépendent de la pertinence des variables sélectionnées. Il apparait donc important de travailler dans un espace de dimension réduite avec des variables discriminantes au regard de la partition connue des données. Pour ce faire, nous avons d'une part utilisé l'algorithme PB-Clus pour visualiser les groupes existants dans les données, et d'autre part nous avons appliqué l'algorithme Fisher-EM pénalisé pour classer les données et sélectionner les variables discriminantes des cellules pathologiques.

Enfin, le chapitre 11 est un chapitre de conclusion qui résume les deux travaux développés dans ce manuscrit ainsi que leurs limites. Par ailleurs, nous exposons brièvement les travaux en cours et les diverses perspectives de recherche. En particulier, dans le cadre de l'algorithme Fisher-EM, nous travaillons actuellement sur une version supervisée et semi-supervisée de cette approche qui permettrait de traiter les problèmes de faux labels et de labels parcimonieux qui apparaissent fréquemment dans le cadre de données biologiques.

Part I

Subspace clustering in a Gaussian mixture model

Chapter 2

State-of-the-art in model-based clustering

Clustering is a data analysis tool which aims to group together data in homogeneous clusters. The clustering problem has been widely studied for years and it usually occurs in all applications for which a partition of data is necessary. In particular, more and more scientific fields require to cluster data in the aim to understand, interpret or make a decision. The earliest approaches which were proposed to cluster data, were based on heuristic, geometric and iterative procedures. They relied on dissimilarity measures between pairs of observations. The most well-known dissimilarity measure is perhaps the distance based on the between groups, previously introduced by Ward [174] for hierarchical clustering. In the same way, the k-means algorithm developed by [115] is perhaps the most popular clustering algorithm among the known iterative procedures. However, even though these methods were extensively studied, they present some disadvantages. In particular, certain statistical properties of these approaches are still unknown and questionings still remain in practice such as the determination of the number of clusters for example.

More recently, clustering was defined in a probabilistic framework allowing thus to formalize the notion of cluster through a probability distribution. One of the main asset of this probabilistic approach remains in the fact that the obtained partition can be statistically interpreted. The first works on finite mixture models were introduced in particular by Wolfe [181], Scott *et al.* [158] and Duda *et al.* [48] and since, many authors keeps on studying its properties and extending clustering in finite mixture model (McLachlan *et al.* [125, 127], Banfield and Raftery [11] or Fraley and Raftery [57],[59]).

In this chapter, the probabilistic framework of clustering will be firstly introduced. In particular, we will focus on the specific but well-known case of the Gaussian mixture models and its associated estimation procedures. Moreover, partially because of the fact that the data storage has become easier and cheaper to get, modern data are very often high dimension. We will see in Section 2.1.2 how the high dimensionality poses problems in the probabilistic

framework of clustering and the so-called well-known *curse of dimensionality* [10] will be introduced. The existing solutions in the Gaussian mixture model will be presented through regularization methods, parsimonious models or dimension reduction methods. In particular, Section 2.2 will focus on linear dimension reduction. However, in the unsupervised context, the dimension reduction methods do not usually consider the classification task and it can occur, therefore, a loss of information which could have been discriminative for the clustering step. The combination of dimension reduction with the classification aim occurs frequently in the supervised context, that is why, we will present in this same section, Fisher discriminant analysis (FDA), an old-fashioned statistical but powerful tool, for classification and dimension reduction, in the supervised classification framework. Finally, the last section of this chapter will detail recent approaches, called the subspace clustering methods, which were proposed in the past few years to model the data of each group in low-dimensional subspaces while avoiding dimension reduction.

2.1 Model-based clustering

2.1.1 Mixture model and the EM algorithm

Model-based clustering, widely studied by [59, 127], aims to partition observed data into several groups which are modeled separately. The overall population is considered as a mixture of these groups and each component is modeled by a probability distribution. Let us consider a given dataset of n observations $\{y_1, \dots, y_n\} \in \mathbb{R}^p$ that one wants to divide into K homogeneous groups *i.e.* adjoin to each observation y_i a value $z_{ik} = 1$, for $k = 1, \dots, K$, if the observation y_i belongs to the kth cluster and $z_{ik} = 0$ otherwise.

Let us also assume that $Y \in \mathbb{R}^p$ is a random vector linked to the observed independent realizations $\{y_1, \dots, y_n\}$ and that $z_i = \{z_{i1}, \dots, z_{iK}\}$ are independent unobserved realizations of a random vector $Z \in \{0, 1\}^K$. The pairs $\{(y_i, z_i)\}_{i=1}^n$ is usually referred to as the complete dataset. By defining f the probabilistic density function of Y, the finite mixture model is written as:

$$f(y) = \sum_{k=1}^{K} \pi_k f_k(y), \tag{2.1.1}$$

where π_k and f_k respectively represent the mixture proportion the conditional density function of the kth cluster. The clusters are often modeled by the same density function in which case the finite mixture model is:

$$f(y) = \sum_{k=1}^{K} \pi_k f(y|\theta_k), \tag{2.1.2}$$

where θ_k is a parameter vector for the kth cluster.

2.1.1.1 The Gaussian mixture model

Most commonly, the density function $f(y|\theta_k)$ is assumed to be a multivariate Gaussian density $\phi(y|\theta_k)$ parametrized by its mean μ_k and its covariance matrix Σ_k, such that the density function of Y can be written in this way,

$$f(y;\theta) = \sum_{k=1}^{K} \pi_k \phi(y|\theta_k), \tag{2.1.3}$$

where $\sum_{k=1}^{K} \pi_k = 1$ and:

$$\phi(y|\theta_k) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2} (y-\mu_k)^t \Sigma_k^{-1} (y-\mu_k)\right)$$

stands for the multivariate Gaussian density with parameters $\theta_k = (\mu_k, \Sigma_k)$. Then, the corresponding log-likelihood is:

$$\log L(\theta; y) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k \phi(y_i; \theta_k)\right). \tag{2.1.4}$$

In order to determine the parameter value θ^*, the log-likelihood function needs to be maximized. However, since the class label z_i of each observation y_i are unknown, the maximization of equation (2.1.4) in a mixture model is untractable. Thus, by considering the pairs $\{(y_1, z_1), \ldots, (y_n, z_n)\}$ introduced previously, the complete log-likelihood of θ is:

$$\ell(\theta; y, z) = \sum_{k=1}^{K} \sum_{i=1}^{n} z_{ik} \log(\pi_k \phi(y_i; \theta_k)),$$

where $z_{ik} = 1$ if y_i comes from the kth component and $z_{ik} = 0$ otherwise. Dempster *et al.* [45] proposed an iterative algorithm called the Expectation-Maximization (EM) algorithm to estimate the unknown parameters μ_k, Σ_k and π_k by the maximum log-likelihood, those was extended by McLachlan *et al.* [126] and Celeux *et al.*[34, 35] in particular.

2.1.1.2 EM algorithm for GMM

Since the maximization of the complete likelihood is untractable, Dempster *et al.* [45] proposed an iterative procedure to find the maximum of likelihood functions in incomplete data problems. The main idea of the EM algorithm remains in the fact that the algorithm is based on the maximization of the conditional expectation of the log-likelihood given current parameters θ. This algorithm consists in forming a sequence $\left(\theta^{(q)}\right)_q$ which satisfies:

$$\theta^{(q)} = \arg\max_{\theta} Q\left(y_1, \ldots, y_n, \theta | \theta^{(q-1)}\right), \tag{2.1.5}$$

where $Q\left(y_1,\ldots,y_n,\theta|\theta^{(q-1)}\right) = \mathbf{E}\left[\log f\left(y,z;\theta\right)|y;\theta^{(q-1)}\right]$ is the conditional expectation of the complete log-likelihood of the observed data $\{y_1,\ldots,y_n\}$ and has the following form:

$$Q\left(\theta\right) = \sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik}\log\left(\pi_k\phi\left(y_i,\theta_k\right)\right), \tag{2.1.6}$$

where $t_{ik} = \mathbf{E}\left(z_{ik}|y_i,\Theta\right)$ and $z_{ik} = 1$ if y_i comes from the kth component and $z_{ik} = 0$ otherwise.

From an initial solution $\theta^{(0)}$, the EM algorithm alternates two steps: first, the expectation step named E-step which computes the expectation of the complete log-likelihood conditionally to the current value of the parameter and the maximization step (M-step) which maximizes the expectation of the complete likelihood subject to the post probabilities.

E-step: this step aims to compute, at iteration (q), the expectation of the complete log-likelihood conditionally to the current value of the parameter $\theta^{(q-1)}$, which, in practice, reduces to the computation of $t_{ik}^{(q)} = E[z_{ik}|y_i,\theta^{(q-1)}]$ where z_{ik} is defined as previously. Let us also recall that $t_{ik}^{(q)}$ is as well the posterior probability that the observation y_i belongs to the kth component of the mixture. According to the mixture model exposed in equation (2.1.3), the posterior probabilities $t_{ik}^{(q)}$, $i = 1,...,n$, $k = 1,...,K$, can be expressed through Bayes' theorem formula as:

$$t_{ik}^{(q)} = \frac{\pi_k^{(q-1)}\phi(y_i,\theta_k^{(q-1)})}{\sum_{l=1}^{K}\pi_l^{(q-1)}\phi(y_i,\theta_l^{(q-1)})}, \tag{2.1.7}$$

where $\phi(.)$ is a Gaussian density function, and π_k and $\theta_k = \{\mu_k,\Sigma_k\}$ are the parameters of the kth mixture component estimated in the previous iteration.

M-step: this step estimates the model parameters by maximizing the conditional expectation of the complete log-likelihood. For the traditional GMM, at iteration (q), the maximization of Q defined by equation (2.1.6) conduces to an estimation of the mixture proportions π_k, the means μ_k and the covariance matrices Σ_k for the K components:

$$\hat{\pi}_k^{(q)} = \frac{n_k^{(q)}}{n}, \tag{2.1.8}$$

$$\hat{\mu}_k^{(q)} = \frac{1}{n_k^{(q)}}\sum_{i=1}^{n} t_{ik}^{(q)} y_i, \tag{2.1.9}$$

$$\Sigma_k^{(q)} = \frac{1}{n_k^{(q)}}\sum_{i=1}^{n} t_{ik}^{(q)}\left(y_i-\hat{\mu}_k^{(q)}\right)\left(y_i-\hat{\mu}_k^{(q)}\right)^t, \tag{2.1.10}$$

where $n_k = \sum_{i=1}^{n} t_{ik}^{(q)}$.

These both steps are iteratively computed until the convergence of the log-likelihood.

One of the most outstanding properties of the EM algorithm is that it guarantees an

improvement of the likelihood function at each iteration: each update to the parameters resulting from an E step followed by an M step is guaranteed to increase the log-likelihood function. In particular, Wu [182] proved that the sequence of $\left(\theta^{(q)}\right)_q$ increases the log-likelihood and converges to a local optimum under certain regularity conditions. For any parameter value θ^* which satisfies:

$$Q\left(\theta^*|\theta^{(q-1)}\right) \geq Q\left(\theta^{(q-1)}|\theta^{(q-1)}\right),$$

the likelihood function increases. However, despite the likelihood-ascent property, additional conditions are required to verify that the final value is not a local minimizing point. A more detailed approach of the EM algorithm can be referred to by [126].

Once the EM algorithm converged, the partition $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ is designed afterward using the maximum *a posteriori* (MAP) rule which implies that the observation y_i is assigned to the group having the highest posterior probability. In order to ease the notation and the interpretation of the decision rule for the rest of this manuscript, let us introduce a classification function Γ_k:

Definition : The classification function Γ_k is defined conditionally to the class k for $k = 1, \ldots, K$:

$$\begin{array}{rcl} \Gamma_k : \mathbb{R}^p & \longrightarrow & \mathbb{R} \\ y_i & \longmapsto & -2\log(\pi_k \phi(y_i, \theta_k)). \end{array}$$

Then, according to this definition, the MAP rule of an observation y_i can be rewritten in this way:

$$\delta^*(y_i) = \arg \max_{k=1\ldots K} \Gamma_k(y_i).$$

2.1.1.3 Limitations of the EM algorithm

Although the EM algorithm is widely used, it is also well-known that, as the EM is a deterministic algorithm, its performances are linked to its initial conditions. Indeed, the solution provided by the EM algorithm is strongly dependent of the starting point which implies that the attained stationary point can be a local optimum or a saddle-point of the log-likelihood function. Thus, to deal with this problem, several strategies were proposed in the literature for initializing the EM algorithm. In an earlier approach, Hathaway *et al.* [86] investigated a constrained version of the EM algorithm for the univariate case by adding a constraint on the parameter space and, simulations of this constrained algorithm suggested that such an algorithm was more robust than the traditional one. However, such constraints are applicable in the univariate case only. A detailed overview of these methods can be seen in Chapter 2 of [125]. A more popular and recent practice [15] executes the EM algorithm several times from a random initialization and keep only the set of parameters associated with the high-

est likelihood. The use of k-means or of a random partition are also standard approaches for initializing the algorithm. Some authors as [11, 44, 57] use also model-based hierarchical clustering for initialize starting values of the EM algorithm for datasets which are not too large. However, such an approach has a heavy computational cost and therefore the computational time increases drastically as soon as the datasets become large. Finally, McLachlan and Peel [127] also proposed an initialization through the parameters by generating the mean and the covariance matrix of each mixture component from a multivariate normal distribution parametrized by the empirical mean and empirical covariance matrix of the data. In practice, the preferred initialization consists to run several mini-EM in first and then to choose the parameters corresponding to the highest log-likelihood.

2.1.1.4 Extensions of the EM algorithm

The CEM algorithm: the classification EM algorithm proposed by Celeux and Govaert [35] can be seen as a classifying version of the EM algorithm and was designed to optimize classification maximum likelihood criteria in the mixture context. A third step, named the C step, is added between the traditional E and M steps using a maximum *a posteriori* rule. This means that each observation is assigned to the cluster for which the posterior probability is the highest. The main advantage of the CEM algorithm is its quickness to converge compared to the traditional EM algorithm. However, this approach provides biased estimates of the mixture parameters and it is theoretically preferable to use the EM algorithm in the context of mixture model. Moreover, in a practical point of view, the provided solution does depend on the starting value of the parameters which is, according to the authors dramatically true when the clusters are not well-separated. However, Celeux et Govaert made an interesting remark on the CEM algorithm since under the assumptions of a Gaussian mixture model with equal proportions and common covariance matrices of the form $\sigma^2\mathbf{I}_p$, the CEM algorithm is equivalent to the k-means algorithm. Indeed, under these assumptions on the covariance matrices in the Gaussian mixture model, the maximization of the log-likelihood criterion and of the k-means criterion are both based on the minimization of the within covariance matrix.

The SEM algorithm: the stochastic EM algorithm was proposed by Celeux and Diebolt [34, 35] as an alternative to the EM algorithm. Indeed, as it is well-known that the solution provided by the EM algorithm is strongly dependent of the starting point or/and can be stucked in a saddle-point of the log-likelihood function, the aim of a stochastic algorithm is to add random perturbations to avoid such local traps. Celeux and Diebolt proposed to add a stochastic step between the E and the M steps of the EM algorithm in order to randomly modify the class label of each observation. At each iteration (q), the S step generates a randomized complete sample $\left\{\left(y_1, z_1^{(q)}\right), \ldots, \left(y_n, z_n^{(q)}\right)\right\}$ from posterior distribution $\left(t_{i1}^{(q)}, \ldots, t_{iK}^{(q)}\right)$ for all observations $i = \{1, \ldots, n\}$ given the observed data $\{y_1, \ldots, y_n\}$. Then, each observation i is assigned to the cluster $\mathcal{C}_k$ with probability t_{ik}. The SEM algorithm can be viewed as a

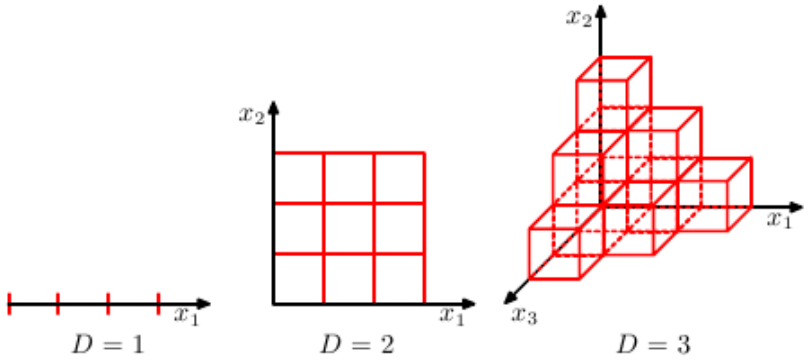


Figure 2.1: Illustration of the curse of dimensionality (Figure borrowed to Bishop in Chapter 1.4 (p. 35) of [17]).

stochastic version of both EM and CEM algorithms. However, the SEM algorithm does not converge point-wise because of the randomized perturbations, but Celeux and Diebolt showed that it converges in distribution. Thus, in practice, the authors propose a hybrid solution which consists of running about ten iterations of the SEM algorithm such as a stationary point is reached in first, and then, running the CEM algorithm from the highest value of the proposed log-likelihood criterion. Other stochastic versions of the EM exist in the literature such as the Monte Carlo EM (MCEM) algorithm introduced by Wei and Tanner [177] which approximates the E-step by using a Monte Carlo average to estimate the expectation of (2.1.7). The stochastic implementation of the EM algorithm is also used to overcome untractable computations at the E-step [124] or to speed up the convergence [30]. A comparison of these different types of stochastic implementation of the EM algorithm is detailed in [96].

2.1.2 The curse of dimensionality

The *curse of dimensionality* introduced by Bellman [10] refers to the exponential growth of an hyper volume as a function of dimensionality. The origin of the problem is illustrated in Figure 2.1 which shows that if we divide a region of a space into regular cells, then the number of such cells grows exponentially with the considered space. This implies the need of the knowledge of an exponentially large quantity of training data in order to ensure that the cells are not empty. In particular, Silverman [160] illustrated this problem from the necessary number of kernels to approximate a dimension-dependent distribution up to a defined precision. An other manifestation of the curse of dimensionality is linked to the geometric properties of high-dimensional spaces which are totally counter intuitive compared to those obtained in low dimensional spaces. The examples of Figures 2.3 illustrate such geometric properties. Indeed, one traditional example depicted in Figure 2.3a stands for the evolution of the volume of a unit-radius sphere with respect to the dimensionality. It can be observed that from the dimension 1 until 5, the volume of the sphere increases and it then decreases until the dimension $p = 20$.

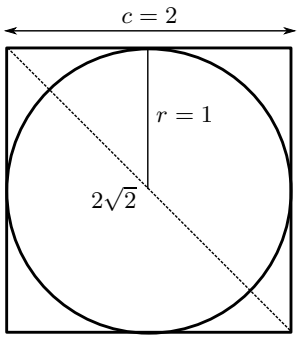


Figure 2.2: Unit-radius circle which is tangent to a square with diagonal length equal to $2\sqrt{2}$.

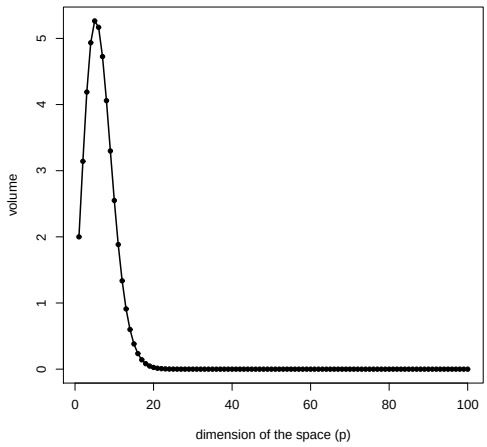
Beyond the dimension 20, the volume is almost equal to zero. The second example illustrated by Figure 2.2 (in 2 dimensions), considers the ratio $\rho(p)$ between the volumes of a unit-radius sphere which is tangent of the cube with diagonal length equal to $2\sqrt{2}$ with respect to the dimensionality p of the space. The ratio of these both volumes is:

$$\rho(p) = \frac{\pi^{\frac{p}{2}}}{2^p \Gamma\left(\frac{p}{2}+1\right)}.$$

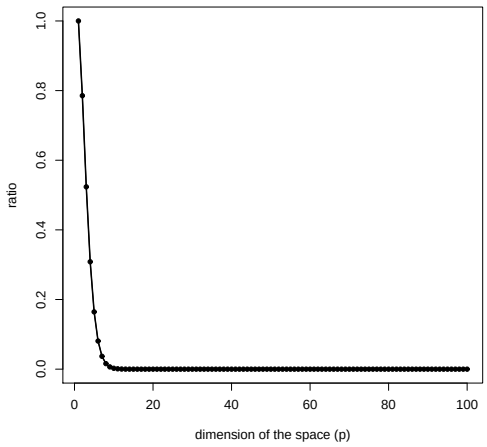
Figure 2.3b stands for the evolution of this ratio $\rho(p)$ with respect to the dimensionality p. Firstly, it can be noted that, in a space of dimension 2, this ratio is equal to $\frac{\pi}{4}$ which means that more than 75% of the surface of the cube is contained in the sphere. Then, this ratio decreases quickly towards 0 when the dimension increases and is equal to zero from the dimension $p = 10$. This simple example means that, if we consider a density and if the samples are drawn randomly and uniformly in a cube, then the probability that the points fall near the corner of the cube is almost equal to 1. Both examples show that the space of dimension p is almost empty since most of points are located near a space of dimension $p-1$ and are related to what it called the *empty space phenomenon*, introduced by Scott and Thomson in [159]. This phenomenon has been widely used to efficiently classify high-dimensional data since it means that high-dimensional data do not fit the whole observation space but live in low-dimensional subspaces. It is well-known indeed that, in high dimension, model-based clustering methods unfortunately can show a disappointing behavior since the quality of the clustering mainly depends of the estimation of the covariance matrices. In particular, the fitted data partition depends on the quantity:

$$\begin{aligned}\Gamma_k(y) &= -2\log(\pi_k \phi(y, \theta_k)) \\ &= (y - m_k)^t S_k^{-1}(y - m_k) + \log(|S_k|) - 2\log(\pi_k) + p\log(2\pi),\end{aligned}$$

for $k = 1, \ldots, K$, which is mainly defined by the inversion of the K covariance matrices. Consequently, when the number of observations n is of the same order than the number of



(a) Volume of a unit-sphere function of the dimension p of the space.



(b) Ratio between the volumes of a unit-sphere tangent with a hyper-cube of length equal to 2.

Figure 2.3: Examples explaining the particular geometric properties of the high-dimensional space.

dimensions p, or when $n \ll p$ which currently occurs in biological data (genetics for example), the covariance matrices are singular and their inversion become impossible. Moreover, it often appears in practice that the covariance matrices are ill-conditioned. This implies biases on the computation of the inverse covariance matrices and, to the end, to the fitted partition. To deal with these problems, different approaches were proposed in the literature and are detailed in the following subsections.

2.1.3 Parsimonious models

Traditional parsimonious model: In the case of Gaussian mixture model for clustering, parsimonious models are introduced to deal with the high-dimensional problem *i.e.* models which need a "reasonable" number of parameters to be estimated. Indeed, as the number of free parameters depends on the number of components of the mixture and on the dimension of the observation space: higher the dimension is, the more the number of parameters to be estimated increases. For example, the unconstrained classical model with full covariance matrices is a highly parametrized model and requires the estimation of 20603 parameters when the number of components is $K = 4$ and the number of variables is $p = 100$. Traditional ways to reduce the number of parameters to estimated is to constraint the covariance matrices to be the same across all mixture components. In general, the multivariate normal density has ellipsoidal contours and the covariance matrices can also be constrained to make the contours spherical or axis-aligned. For the sake of comparison, Table 2.1 presents a comparison between the well-known parsimonious models which can be obtained from a Gaussian mixture model with K components in a p-dimensional space. In this table, the Full-GMM model refers to the

Model	Nb. of parameters	$K = 4$ and $p = 100$
Full-GMM	$(K-1) + Kp + Kp(p+1)/2$	20603
Com-GMM	$(K-1) + Kp + p(p+1)/2$	5453
Diag-GMM	$(K-1) + Kp + Kp$	803
Com-Diag-GMM	$(K-1) + Kp + p$	503
Sphe-GMM	$(K-1) + Kp + K$	407
Com-Sphe-GMM	$(K-1) + Kp + 1$	404

Table 2.1: Number of free parameters to estimate for parsimonious Gaussian mixture models with K components and p variables.

classical Gaussian mixture model introduced by [158] with unconstrained covariance. Moreover, when the covariance matrices are assumed to be equal to a common covariance matrix but not need to be spherical ($\Sigma_k = \Sigma$, $\forall k$), such a Gaussian mixture model first introduced by [60] is referred to the Com-GMM model in the table. The Diag-GMM model refers to the Gaussian mixture model for which the covariance matrices are supposed to be spherical but different to each other then $\Sigma_k = \text{diag}(\sigma_{k1}^2, ..., \sigma_{kp}^2)$ with $\sigma_{k.}^2 \in \mathbb{R}^+$. Finally, Sphe-GMM refers to the Gaussian mixture model for which $\Sigma_k = \sigma_k^2 I_p$ with $\sigma_k^2 \in \mathbb{R}$. Two other intermediate models are added in this table. In particular, the Com-Diag-GMM which supposes diagonal common covariances such as $\Sigma_k = \Sigma = \text{diag}\left(\sigma_1^2, \ldots, \sigma_p^2\right)$ or the most constrained model, the Com-Sphe-GMM model which assumes that the covariance matrices of each class are equal and spherical such that $\Sigma_k = \Sigma = \sigma^2 I_p, \quad \forall k$ with $\sigma^2 \in \mathbb{R}$. The number of free parameters to estimate given in the central column can be decomposed in the number of parameters to estimate for the proportions $(K-1)$, for the means (Kp) and for the covariance matrices (last terms). Whereas the Full-GMM model is a highly parametrized model, the Sphe-GMM and Com-Sphe-GMM models are conversely very parsimonious models since they respectively require the estimation of only 407 and 404 parameters when $K = 4$ and $p = 100$. Finally, the Com-GMM model presents an intermediate level of parsimony since the number of parameters to estimate is 5453.

A family of parsimonious models: In parallel, Banfield and Raftery [9] and Celeux and Govaert [36] proposed a statistical framework, in the case of multivariate Gaussian mixture model, for which the different geometries of the clusters are taken into account. For that, they parametrize the covariance matrices from an eigenvalue decomposition:

$$\Sigma_k = \lambda_k D_k A_k D_k^t,$$

where D_k is the matrix of eigenvectors which determines the orientation of the cluster, A_k is a diagonal matrix proportional to the eigenvalues which explains its shape, and λ_k is a scalar determining its volume. They refer to this model by the $[\lambda_k D_k A_k D_k^t]$ model. This enables us to enumerate 14 different submodels by constraining or not the parameters λ_k, D_k and A_k. This family of 14 models are grouped in Table 2.2 in which the first column stands for the model

Model	Name	Nb. of parameters	$K = 4$ $p = 100$
$[\lambda_k D_k A_k D_k^t]$	VVV	$(K-1) + Kp + Kp(p+1)/2$	20603
$[\lambda D_k A_k D_k^t]$	-	$(K-1) + Kp + Kp(p+1)/2 - (K-1)$	20600
$[\lambda_k D_k A D_k^t]$	VEV	$(K-1) + Kp + Kp(p+1)/2 - (K-1)(p-1)$	20306
$[\lambda D_k A D_k^t]$	EEV	$(K-1) + Kp + Kp(p+1)/2 - (K-1)p$	20303
$[\lambda_k D A_k D^t]$	-	$(K-1) + Kp + p(p+1)/2 + (K-1)p$	5753
$[\lambda D A_k D^t]$	-	$(K-1) + Kp + p(p+1)/2 + (K-1)(p-1)$	5750
$[\lambda_k D A D^t]$	-	$(K-1) + Kp + p(p+1)/2 + (K-1)$	5456
$[\lambda D A D^t]$	EEE	$(K-1) + Kp + p(p+1)/2$	5453
$[\lambda_k B_k]$	VVI	$(K-1) + Kp + Kp$	803
$[\lambda B_k]$	EVI	$(K-1) + Kp + Kp - (K-1)$	800
$[\lambda_k B]$	VEI	$(K-1) + Kp + p + (K-1)$	506
$[\lambda B]$	EEI	$(K-1) + Kp + p$	503
$[\lambda_k \mathbf{I}_p]$	VII	$(K-1) + Kp + K$	407
$[\lambda \mathbf{I}_p]$	EII	$(K-1) + Kp + 1$	404

Table 2.2: Number of free parameters to estimate for parsimonious Gaussian mixture models with K components and p variables.

names used by Celeux and Govaert and the second one represents the nomenclature used by Raftery and Fraley. First of all, we can observe that this family of models can be divided in three levels of parsimony as well as in Table 2.1: 4 models are highly parametrized as the Full-GMM model, 4 models have an intermediate level of parsimony since the number of parameters to estimate is around 5500 as well as the Com-GMM model and finally, the last 6 models are very parsimonious and are in the same order as Diag-GMM or Sphe-GMM. Besides, this reformulation of the covariance matrices enables to rewrite the previous constrained models. For example, the Com-GMM model which can be rewritten as $[\lambda D A D^t]$. There is also the works of [138] which uses the equal shape ($\lambda_k = \lambda, \forall k$) and equal volume ($A_k = A, \forall k$)

such that $\Sigma_k = \lambda D_k A D_k^t$. However, the work of Celeux and Govaert widens the family of parsimonious models since they add unusual models which allow different volumes for the clusters such as the $[\lambda_k DAD^t], [\lambda_k DA_k D^t]$ and $[\lambda_k D_k A D_k^t]$ models. Moreover, by assuming that the covariance matrix Σ_k are diagonal matrices, Celeux and Govaert authors proposed a new parametrization of $\Sigma_k = \lambda_k B_k$ where $|B_k| = 1$. Such a parametrization leads to 4 other submodels detailed in Table 2.2. Finally, by considering the spherical shape, it leads to 2 other models, the $[\lambda_k \mathbf{I}_p]$ and $[\lambda \mathbf{I}_p]$ models which enable to rewrite the Sphe-GMM model for which the covariance matrix of the cluster k is rewritten as $\Sigma_k = \lambda_k \mathbf{I}_p, \forall k$ where $\lambda_k \in \mathbb{R}^+$. In the case of the Com-Sphe-GMM model, the covariance matrices can be noted as $\Sigma_k = \lambda \mathbf{I}_p$ with $\lambda_k \in \mathbb{R}^+$. The reader can refer to [36] for a more detailed approach of these models.

Pseudoinverse or simple regularization To deal with the ill-conditioning or singularity of the K covariance matrices, a very simple approach is to bring a regularized term on the covariance matrix itself. A common method to handle this problem is to use the *pseudoinverse* Σ_k^+ instead of Σ_k^{-1}. An other way to deal with ill-conditioning covariance matrices, is to add a positive term σ^2 to the diagonal of the empirical covariance matrix $\hat{\Sigma}_k$ such as:

$$\tilde{\Sigma}_k = \hat{\Sigma}_k + \sigma^2 \mathbf{I}_p.$$

This type of regularization is comparable to those used in the ridge regression. An other regularization can also be used:

$$\tilde{\Sigma}_k = \hat{\Sigma}_k + \sigma_k \Omega,$$

where Ω is a square matrix of dimension $p \times p$. Such regularized term, introduced by Hastie in [83], looks like the previous one but the difference between both penalizations remains in the fact that PDA penalizes also correlations between the predictors. Thus, the matrix Ω enables to penalize the correlations.

2.2 Dimension reduction

As we have seen previously, several strategies were proposed in the literature for model-based clustering among which parsimonious models and regularization approaches to deal with the dimensionality. Moreover, since the dimension of observed data is usually higher than their intrinsic dimension, it is theoretically possible to reduce the dimension without losing information. Thus, earliest approaches proposed to overcome the problem of high dimension in clustering, by first reducing the dimension before using a traditional clustering method. Among the unsupervised tools of dimension reduction, principal component analysis (PCA) or factor analysis (FA) [101] are traditional and certainly the most used techniques for dimension reduction. They aim to project the data on a lower dimensional subspace in which axes are build either by maximizing the variance of the projected data or by explaining the overall covariance structure. PCA and FA are both a linear tool, which means that non-linear

dependencies are not taken into account. Other non-linear projection tools can be mentioned such as Kohonen's maps [109] or the generative topographic map approach [18]. For a review of these dimension reduction approaches, see [171]. An other way to reduce the dimension in an unsupervised problem was recently considered in [148] and [120] in which the problem of feature selection for model-based clustering is recasted as a model selection problem. However, most of these dimension reduction methods do not consider the clustering task and can provide a suboptimal representation for the clustering step since discriminative information can be not taken into account. Only few approaches combine dimension reduction with the classification aim but unfortunately there are all in the supervised context. In particular, Fisher discriminant analysis (FDA) is one of them in the supervised classification framework.

This Section details the different linear approaches of dimension reduction in both cases of unsupervised and supervised contexts.

2.2.1 The unsupervised case

2.2.1.1 Principal component analysis

Principal component analysis (PCA) is certainly the most popular linear method used for dimension reduction. It was introduced by Pearson [144] in 1901 who defines PCA as a linear projection that minimizes the average projection cost. Later, Hotelling [88] proposed an other definition for PCA which the aim is to reduce the dimension of the data by keeping as much as possible the variation of the dataset. In other words, this method aims to find an orthogonal projection of the dataset in a low-dimensional linear subspace, such that the variance of the projected data is maximized. In this section, the maximum variance formulation is considered but the reader could refer to Chapter 12 of [17] for an explanation of both definitions of PCA.

Formulation in terms of maximization of the variance : Let consider Y a $n \times p$ data matrix of dimension p and $\Sigma = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^t(y_i - \bar{y})$ the total covariance matrix of the dataset $\{y_1, \ldots, y_n\}$ where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is its empirical mean. Firstly, one consider the case of the projection of the dataset onto a single dimension space. This amounts to determine a p-dimensional vector u_1, such that the variance of projected data $u_1^t \Sigma u_1$ is maximized with respect to u_1, under the normalization condition $u_1^t u_1 = 1$. The criterion to maximize can be rewritten using the Lagrange multiplier:

$$u_1^t \Sigma u_1 - \lambda(1 - u_1^t u_1). \tag{2.2.1}$$

By setting the derivative of equation (2.2.1) to zero, there is a stationary point in:

$$\Sigma u_1 = \lambda u_1. \tag{2.2.2}$$

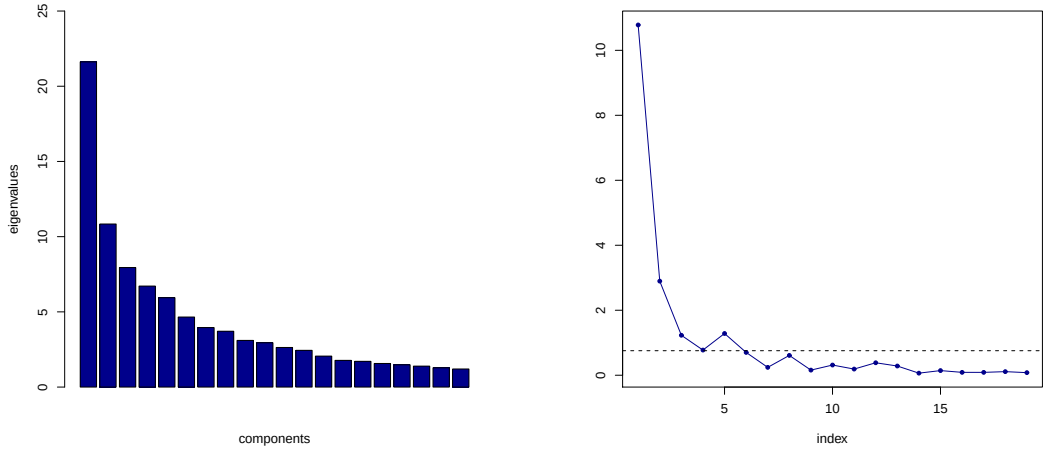
and the very well-known result is obtained: equation (2.2.1) is maximized when u_1 is the eigenvector of Σ associated with the highest eigenvalue λ and this eigenvector is called the

first principal factor. Moreover, the first principal component $X = Yu_1$ is defined as the projected data on this one-dimensional space. By considering the general case for an d-dimensional subspace which aims is to find an optimal linear projection such as the variance of the projected data is maximized, the d principal components correspond then to the d eigenvectors of Σ associated to its d largest eigenvalues. In first, note that the principal factors obtained are orthogonal to each other and also that each principal component is a linear combination of the initial variables which limits the relevance of PCA for non-linear distribution of datapoints. Secondly, one can observe that if the principal components are in the same dimensionality as the observation space, then there is no information loss since the data will just have been rotated. Consequently, there is a loss of information in the case of dimension reduction which implies the question about the choice of the dimensionality of the projection space which is d.

Choice of the dimensionality : Different methods were proposed in the literature, but most of them are based on empirical criteria. In practice, since the eigenvalues stand for the weight of the variance hold by the principal components in the total covariance matrix, the number d of axis can be selected from a certain a proportion of $\sum_{j=1}^{d} \lambda_j$ (90% for example). An other very popular criterion based on a graphical method is to detect an "elbow" in the plot of the eigenvalues and keep the dimensions which are just before this elbow. These both methods remain graphical which can pose some problems ,in particular if the slope gradually becomes less steep, with no clear elbow. Then it is clearly less easy to use such a procedure. A more automatic approach named the *scree-test* was developed by Cattel [33]. He suggests in first to plot the eigenvalues of the covariance matrix and then to find the place where the graph seems to behave randomly. This method is based on the differences between the consecutive eigenvalues and on the detection of an "elbow" in the eigenvalues scree. The number of components corresponds to the number of eigenvalues being above a threshold as in Figure 2.4 : Figure 2.4a stands for the 20 first largest eigenvalues ordering in a decreasing order of the correlation matrix of the USPS dataset and the corresponding scree plot is depicted in Figure 2.4b. In this example, the threshold is fixed to 7% of the largest difference between the eigenvalues and the scree-test of Cattell identifies an "elbow" in the 5th dimension. This choice can also be confirmed in Figure 2.4a.

Other approaches to deal with the choice of d were proposed in the literature and are based on formal tests of hypothesis. In particular, a statistical test known as Bartlett's test seeks from which value of d the last eigenvalues are equal to zero. Thus, by assuming that n observations $\{y_1, \ldots, y_n\}$ are the realizations of a Gaussian random vector then, the test is based on the null hypothesis,

$$\mathbf{H_0} : \lambda_{d+1} = \cdots = \lambda_p, \tag{2.2.3}$$



(a) (b)

Figure 2.4: Choice of the number of retained components according to the scree-test: (a) stands for the 20 first largest eigenvalues ordered in a decreasing order of the covariance matrix on the USPS dataset and (b) represents the corresponding scree plot of Cattell.

against the general alternative H_1 :

$$\mathbf{H_1} : \exists i, j \in \{d+1, \ldots, p\} \quad \text{such as} \quad \lambda_i \neq \lambda_j.$$

This test is used sequentially to find d. In first, the hypothesis $\mathbf{H_{0,p-1}}$:$\lambda_{p-1} = \lambda_p$ is tested and if $\mathbf{H_{0,p-1}}$ is not rejected, then $\mathbf{H_{0,p-2}}$ is tested. This sequence of tests continues until the hypothesis $\mathbf{H_{0,j}}$ is rejected. Then, the number of retained principal components is $d = j$. This criterion is barely used in practice since the distributional assumptions are very often unrealistic and they tend to overestimate the number of necessary variables to retain. The reader can refer to Chapter 6 of [101] for example, for a more detailed description of empirical methods on the choice of the number of retained components. More recently, Tipping and Bishop [19] and independently Roweis [152], rewrote PCA in a probabilistic framework as the maximum likelihood solution of a probabilistic latent variable model for which an EM algorithm can be derived. In particular, it enables a more automatic approach to find the dimensionality of the subspace from the data. Since PCA is formulated in a probabilistic approach, the authors proposed a Bayesian approach to model selection and the reader could refer to Chapter 12.2 of [17] for more details.

Extensions of PCA : PCA has been widely studied since its birth in 1901 and has been improved as one goes along the different statistical challenges. In particular, in modern scientific applications such as genomic or mass spectrometry, the problem of high dimensional low sample size occurs frequently. This $n < p$ problem refers to situations where the number of features p is larger than the number of available observations n. It appears that the

traditional PCA can not be performed on a high-dimensional dataset since the computational cost increases with p^3 and this is computationally infeasible. A traditional way to deal with the $n < p$ problem is to evaluate the cross-product YY^t instead of computing the covariance matrix based on Y^tY. Indeed, the traditional way to do compute the principal component of PCA is to make an eigenvalue decomposition of the covariance matrix Σ, where $\Sigma = 1/n\, Y^tY$ and Y is a $n \times p$ centered data matrix. For a corresponding eigenvector u_j, the problem is such as $1/nY^tYu_j = \lambda_j u_j$. By multiplying both sides of this equation by Y and by posing $v_j = Yu_j$, then the eigenvalue decomposition problem becomes:

$$1/nYY^tv_j = \lambda_j v_j. \tag{2.2.4}$$

Note that this equation corresponds to the eigenvalue decomposition of the cross-product $YY^t \in \mathbb{R}^{n\times n}$ instead of those of $Y^tY \in \mathbb{R}^{p\times p}$ and implies a lower computational cost in the case $n < p$. Moreover, the $n-1$ eigenvalues associated to equation (2.2.4) are in common with the $p-n+1$ eigenvalues of the original problem. Therefore, the jth eigenvectors which corresponds to the eigenvalue λ_j of the covariance matrix Σ are finally obtained by Y^tv_j. Consequently, to tackle the problem of high-dimensional low sample size, only the eigenvectors and eigenvalues of YY^t are needed to be computed and finally the eigenvectors of the original space are obtained by computing $u_j = Y^tv_j/\sqrt{n\lambda_j}$.

An other extension of PCA is based on the introduction of sparsity in the loadings of the factor components in the aim of interpreting the projected axis. In particular, Zou *et al.* [193] suggested a sparse approach of PCA by proposing a criterion penalized by a ℓ_1-penalty. Besides, even though PCA is the most popular technique for processing and visualization, its effectiveness is limited by its global linearity. Therefore, other alternatives were proposed and here is a non-exhaustive list of works: Kambhatla and Leen [104], for example, developed a method which uses PCA but locally, in restricted parts of the space; Scholkopf and Smola [156] proposed a method called Kernel PCA which transforms the original data in a higher dimensional space before applying PCA in these transformed data; Hastie *et al.* [84] and Girard [65] also proposed non-linear versions of PCA ; from the probabilistic framework designed for PCA, Tipping and Bishop [165] derived a mixture of probabilistic PCA which can be considered for dimensionality reduction and data compression in local linear modeling as well as a way to control the number of parameters for the estimation of covariance structures in high dimensions.

2.2.1.2 Factor analysis

Factor analysis is an other way to deal with dimension reduction. This approach is as old as PCA since its origins are relative to Spearman in 1904 [161] and there is an important literature on this subject (see for example in Chapter 12 [17]). The basic idea of factor analysis is to both reduce the dimensionality of the space and to keep the observed covariance structure of

the data. The factor analysis model can be expressed as a latent variable model: let consider a random vector $Y \in \mathbb{R}^p$ for which $\{y_1, \ldots, y_n\}$ are its independent observed realizations. Let also consider that Y can be expressed by an unobserved random vector $X \in \mathbb{R}^d$ described in a lower dimensional space with dimension d ($d < p$) and that the relationship between these two spaces is linear such that:

$$Y = \Lambda X + \mu + \varepsilon, \tag{2.2.5}$$

where Λ is a $p \times d$ matrix, $\mu \in \mathbb{R}^p$ is the mean vector of Y and $\varepsilon \in \mathbb{R}^p$ is a centered Gaussian noise term with a diagonal covariance matrix Ψ,

$$\varepsilon \sim \mathcal{N}(0, \Psi). \tag{2.2.6}$$

Moreover, in the latent space, the random vector $X \in \mathbb{R}^d$ is assumed to be distributed according to a Gaussian density function such as:

$$X \sim \mathcal{N}(0, \mathbf{I}_d). \tag{2.2.7}$$

The use of assumptions (2.2.5), (2.2.6) and (2.2.7) implies that the marginal distribution of Y is also Gaussian and:

$$Y \sim \mathcal{N}(\mu, \Lambda\Lambda^t + \Psi). \tag{2.2.8}$$

The first remark concerns the columns of Λ, called the factor loadings, which capture the correlation between variables whereas the diagonal matrix Ψ stands for the independent noise variance of each variable. Moreover, such a modeling enables to determine the model parameters by maximum likelihood and the estimation is executed through an iterative procedure since there is no closed form maximum likelihood solution for Λ. The key assumption of the factor analysis model is the constraint on the error covariance Ψ to be a diagonal matrix, then the observed variables Y are conditionally independent given the values of the latent variables X. These latent variables are thus intended to explain the correlations between observation variables while ε represents the variability of each variable. This is where factor analysis fundamentally differs from standard PCA which treats covariance and variance identically. Finally, an other difference remains between factor analysis and PCA presented previously, since factor analysis attempts to achieve a dimension reduction by invoking a model whereas PCA does not. However, in the case of probabilistic PCA (PPCA) developed by Tipping and Bishop [19], factor analysis and PPCA have common assumptions. They however mainly differ from the shape of the covariance matrix of the noise term which is supposed isotropic in the PPCA case ($\Psi = \sigma^2\mathbf{I_p}$).

Finally, Bishop extended the latent variable framework of FA to a non-linear latent variable model called the Generative Topographic Mapping (GTM) [18] which mainly aims to visualize the data. As the factor analysis model, the latent subspace has its dimension d lower than the dimension of the observation space p but it is generally fixed equal to $d = 2$. Moreover,

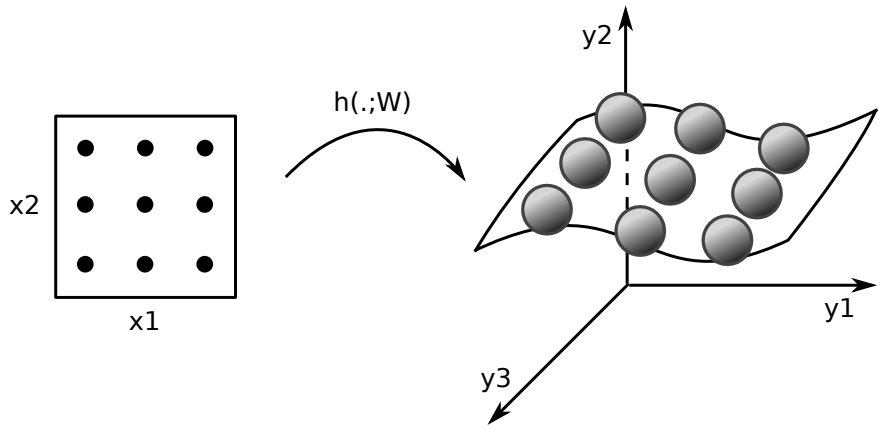


Figure 2.5: In the latent space (right), the

in the general context, the distribution of the latent variables $x = (x_1, \ldots, x_d)$ is supposed to be a sum of delta functions centered on the nodes of a regular grid in the latent space. Moreover, each node of the regular grid is mapped in the data space and forms the centers of the corresponding Gaussian density functions. His main idea is then based on the search for a function $h(.; W)$ governed by a matrix of parameters W which defines a d-dimensional manifold $\mathcal{S}$ embedded within the data space given the latent variable x. Figure 2.5 illustrates schematically for the case $d = 2$ and $p = 3$. Under these assumptions, GTM can be viewed as the probabilistic counterpart of the self-organizing map (SOM) [109]. However, by considering the prior probability of x to be Gaussian, the noise distribution to be Gaussian with a diagonal covariance matrix, then the standard factor analysis model is re-found.

2.2.2 The supervised case: Fisher discriminant analysis

These previous unsupervised approaches of dimension reduction do not consider the classification task which can provide sometimes a sub-optimal data representation for the clustering step. Indeed, dimension reduction methods imply an information loss which could be discriminative. In particular, Chang [38] demonstrated theoretically and practically that the principal components with the largest eigenvalues do not necessarily contain the most information about the cluster structure. Thus, taking a subset of principal components can lead to a major loss of information about the groups in the data. Only few approaches combine dimension reduction with the classification aim but, unfortunately, those approaches are all supervised methods. Fisher discriminant analysis (FDA) is one of the dimension reduction approach in the supervised classification framework which combines dimension reduction with the classification task. It is a powerful tool for finding the subspace which best discriminates

the classes and reveals the structure of the data.

2.2.2.1 Fisher discriminant analysis

In 1936, in the case of supervised classification, Fisher poses the problem of the discrimination of three species of iris described by four measurements in these terms : *What linear function of the four measurements will maximize the ratio of the difference between the specific means to the standard deviation within species?* [54]. Such a remark has led to useful approaches in supervised classification and in dimension reduction.

The main goal of Fisher is to find a linear subspace that separates two class patterns according to a criterion based on a separability measure between both classes (see [48]). Its work was extended to the multi-class problem (see chapter 10 in [62]) and in this subsection, we consider directly the case of multiple classes. From a statistical point of view, let us consider $\{y_1, \ldots, y_n\}$ n-realizations of a random vector Y of dimension p and $\{z_1, \ldots, z_n\}$ n-independent observed realizations of a random variable $Z \in \{1, \ldots, K\}$ which is equal to k when the observation belongs to $\mathcal{C}_k$, the class k. We assume also that the dimensionality p of the original space is greater than the number K of classes. Moreover, let us define the projection of a p-dimensional input vector in a subspace $\mathbb{E} \subset \mathbb{R}^p$ of dimension d lower that the dimensionality of the observed space $d < p$:

$$x = U^t y, \tag{2.2.9}$$

where U is the projection matrix of dimension $p \times d$ and $\{x_1, \ldots, x_n\}$ are n-realizations of a d-dimensional random vector $X \in \mathbb{E}$. The subspace $\mathbb{E}$ is defined to be discriminant which supposes that the Fisher's criterion is large when the between scatter matrix in this subspace s_B is large and when the within scatter matrix s_W is small.

Let consider an introductory example from the USPS datasets.The data comes from a sample of the USPS handwritten image data [94] collected by the Center of Excellence in Document Analysis and Recognition (CEDAR) at SUNY Buffalo. The overall dataset consists of digital numbers $0, 1, 2, \ldots, 9$ described in 256 dimensions, but in this example only the numbers 3, 5 and 8 are considered since they are difficult to discriminate. Figures 2.6a. and 2.6b. illustrate the histogram of the three classes of the USPS358 datasets resulting from projections on the two Fisher discriminant axes and on the two first components of PCA. It can be seen that the discriminative axes greatly improve class separation since the first Fisher's axis allows to distinct three different classes and the second axis improves the class separability: the projected mean of each class on the first axis is well-separated and the variance of each class is smaller in the projected space than in the PCA case where the classes overlap. This figure illustrates the original idea of Fisher to determine a criterion based on the maximization of a function giving a large separation between the projected classes while also giving a small variance within each class. Four different criteria [62] can be found in the literature which

satisfy such a constraint but one criterion is traditionally used:

$$J_1(U) \quad = \quad \mathrm{tr}\left(s_W^{-1} s_B\right) \tag{2.2.10}$$

with $s_B = \frac{1}{n}\sum_{k=1}^{K} n_k(\mu_k - \bar{x})(\mu_k - \bar{x})^t$ and $s_W = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k}(x_i - \mu_k)(x_i - \mu_k)^t$ where $\mu_k = \frac{1}{n_k}\sum_{i\in\mathcal{C}_k}^{K} x_i$ and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ are respectively the mean of the observations x_i in the class k and the mean on the whole dataset. This criterion can be rewritten as an explicit function of the projection matrix U, since $s_B = U^t S_B U$ and $s_W = U^t S_W U$:

$$J_1(U) \quad = \quad \mathrm{tr}\left(\left(U^t S_W U\right)^{-1}\left(U^t S_B U\right)\right) \tag{2.2.11}$$

where

$$S_W = \frac{1}{n}\sum_{k=1}^{K} n_k C_k \tag{2.2.12}$$

is the within-covariance in the input space $\mathbb{R}^p$ with $C_k = \frac{1}{n_k}\sum_{i\in C_k}(y_i - m_k)(y_i - m_k)^t$ and n_k, the number of observations which belongs to the class k;

$$m_k = \frac{1}{n_k}\sum_{i\in C_k}^{K} y_i \tag{2.2.13}$$

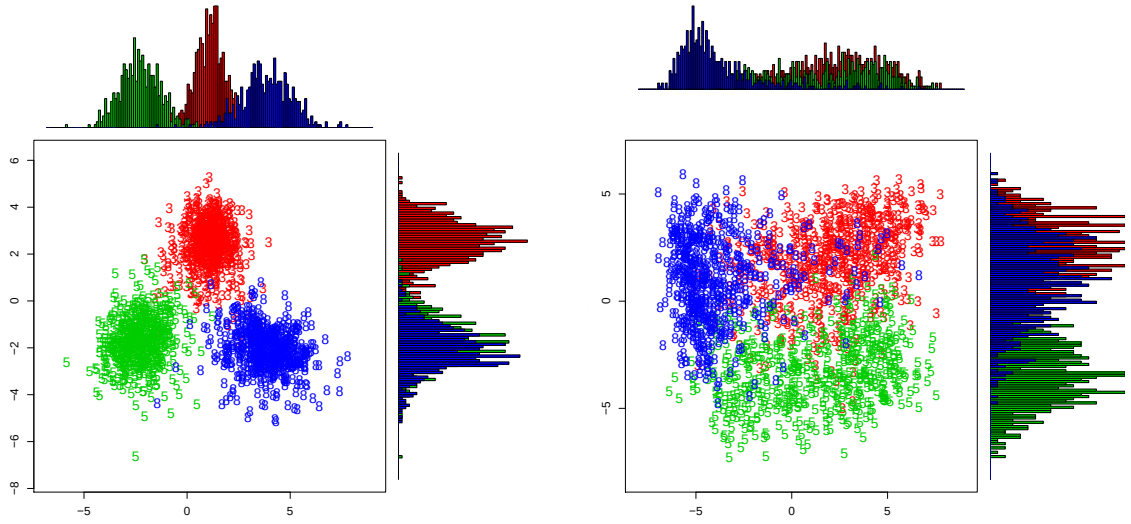
is the mean of the observations y_i in the class k, and:

$$S_B = \frac{1}{n}\sum_{k=1}^{K} n_k(m_k - \bar{y})(m_k - \bar{y})^t \tag{2.2.14}$$

is the between-covariance where $\bar{y} = \frac{1}{n}\sum_{k=1}^{K} n_k m_k$ is the mean of the observations. Consequently, Fisher discriminant subspace looks for a projection matrix U which projects the observations in a discriminant and low-dimensional subspace of dimension d. This subspace is defined such that the linear transformation U of dimension $p \times d$ aims to maximize a criterion which is large when the between-class covariance matrix (S_B) is large and when the within-covariance matrix (S_W) is small. Besides, since the Huygens relation:

$$S = S_W + S_B \tag{2.2.15}$$

between the covariance matrix $S = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^t$ and the within and between covariance matrices holds, the Fisher criterion defined in equation (2.2.11) can be rewritten and optimizes differently from different combinations of S, S_W and S_B. Typical examples are the pairs $\{S_B, S_W\}$, $\{S, S_W\}$ and $\{S_B, S\}$. The equivalence on optimizing the Fisher's criterion with these different pairs is exposed in the next section.



(a) (b)

Figure 2.6: Projection of the USPS358 dataset with the corresponding empirical distribution of each class in the Fisher discriminative subspace (a) and in the 2 first principal components of PCA (b).

2.2.2.2 Optimization of the projection matrix U

The solution linked to the maximization of the Fisher's criterion defined in equation (2.2.10) is the eigenvectors associated to the $K-1$ largest eigenvalues of the matrix $S_W^{-1}S_B$ when S_W is assumed to be non singular. Indeed, according to results from the symmetric-definite generalized eigenvalue problem [66], there exists a non singular matrix $Z \in \mathbb{R}^{p\times p}$ such as both S_W and S_B are diagonalized:

$$Z^t S_W Z = I_p \quad \text{and} \quad Z^t S_B Z = \Lambda = \text{diag}\left(\lambda_1, \ldots, \lambda_p\right). \tag{2.2.16}$$

By stating $\mathbf{z}_j$ a jth column of Z:

$$S_B \mathbf{z}_j = \lambda_j S_W \mathbf{z}_j, \tag{2.2.17}$$

which means that $\mathbf{z}_j$ and λ_j are respectively the jth eigenvector and eigenvalue of $S_W^{-1}S_B$. Besides, the between covariance matrix S_B is positive semi definite which means that the p eigenvalues of $S_W^{-1}S_B$ are positive or nul. Moreover, since S_B is composed of the sum of K matrices based on the term $(m_k - \bar{y})$ and by noting that $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{k=1}^{K} n_k m_k$ with m_k defined by equation (2.2.13), it can be seen that only $K-1$ matrices are independent. Consequently, S_B has rank at most equal to $K-1$ and then only $K-1$ eigenvalues are non-zeros. Thus, the criterion (2.2.10) is maximized by those eigenvectors of $S_W^{-1}S_B$ that correspond to the $K-1$ largest eigenvalues.

Let us consider Z and $\Lambda = \text{diag}\left(\lambda_1, \ldots, \lambda_p\right)$, the eigenvector and eigenvalue matrices of $S_W^{-1}S_B$ and the Huygens relation defined in (2.2.15), then the eigendecomposition problem

$S_B Z = S_W Z\Lambda$ can be rewritten as:

$$S_B Z = (S - S_B) Z\Lambda = SZ\Lambda - S_B Z\Lambda \quad (2.2.18)$$

$$S_B Z = SZ\Lambda (I_p + \Lambda)^{-1}. \quad (2.2.19)$$

Then, it can be observed that Z stands for also the eigenvectors of $S^{-1}S_B$ associated with the eigenvalue matrix $\Lambda (I_p + \Lambda)^{-1}$. In the same manner, by considering $S_B = S - S_W$, the eigendecomposition problem becomes $S_W Z = SZ (\Lambda + I_p)^{-1}$ and the eigenvectors of $S^{-1}S_W$, that correspond to the eigenvalue matrix $(\Lambda + I_p)^{-1}$, are the same as those obtained by the eigendecomposition of $S_W^{-1}S_B$. Since the eigenvalues of Λ are ordered in decreasing order $\lambda_1 \geq \cdots \geq \lambda_d \geq \cdots \geq \lambda_p \geq 0$, then the eigenvalues associated with $(\Lambda + I_p)^{-1}$ are in increasing order:

$$0 \leq \frac{1}{1+\lambda_1} \leq \cdots \leq \frac{1}{1+\lambda_d} \leq \cdots \leq \frac{1}{1+\lambda_p}.$$

and consequently, according to the criterion to optimize, the projection matrix U is formed by the $K-1$ eigenvectors corresponding to the $K-1$ largest eigenvalues of $S_W^{-1}S_B$ or of $S^{-1}S_B$, or to the $K-1$ smallest eigenvalues of $S^{-1}S_W$. However, the J_1 criterion has limitations since the matrix S_W (or S according the pairwise covariance matrices chosen to optimize the criterion) must be nonsingular.

2.2.2.3 Regularization of Fisher discriminant analysis

The optimization of a generalized Fisher criterion $S_1^{-1}S_2$ supposes the non-singularity of the matrix S_1 and it appears that this singularity occurs frequently, particularly in the case of very high-dimensional space or in the case of undersampled problems. In the literature, different solutions are proposed to deal with such a problem in a supervised classification framework: the regularized discriminant analysis (RDA) proposed by Friedman [61], the use of the generalized singular value decomposition (GSVD) developed by Howland [89] and improved by Zhang [192] or a combination of several solutions as Jin *et al.* [99] proposed in the case of small observations but large set of variables.

Standard regularization The eigenvalue problem can be rewritten as $S_1^{-1}S_2 Z = Z\Lambda$ which supposes that the matrix S_1 is non-singular: it fails in practice since S_1 is usually ill-conditioned. As previously in Section 2.1, a common method to handle this problem is to use the *pseudoinverse* S_1^+ instead of S_1^{-1} as it is suggested by [62] and solving the following problem:

$$S_1^+ S_2 Z = Z\Lambda. \quad (2.2.20)$$

A second variant of regularization adds a positive term σ^2 to the diagonal of S_1 such as $\tilde{S}_1 = S_1 + \sigma^2 \mathbf{I}_p$. Therefore, the eigendecomposition problem becomes:

$$\left(S_1 + \sigma^2 \mathbf{I}_p\right)^{-1} S_2 Z = Z\Lambda. \tag{2.2.21}$$

This last regularization can be extended by using the penalized discriminant analysis (PDA) [83] which deals with high correlated variables or undersampled cases. In this case, the scatter matrix S_1 can be regularized by $\tilde{S}_1 = S_1 + \sigma^2\Omega$ where the $p \times p$ matrix Ω penalizes the correlations between variables. Finally, a third variant can be borrowed to Friedman [61] who proposed to regularize the covariance matrices Σ_k of each class in the case of linear discriminant analysis. He suggested that the K covariance matrices $\Sigma_k = \frac{1}{n_k}\sum_{i\in\mathcal{C}_k}(y_i - \mu_k)(y_i - \mu_k)^t$ for $k = \{1, \ldots, K\}$ depends on two regularized parameters λ and γ such that they can be approximated by:

$$\tilde{\Sigma}_k = (1-\gamma)\,\hat{\Sigma}_k(\lambda) + \frac{\gamma}{p}\text{trace}\left(\hat{\Sigma}_k(\lambda)\right)\mathbf{I}_p$$

where:

$$\hat{\Sigma}_k(\lambda) = \frac{(1-\lambda)\,(n_k - 1)\,\Sigma_k + \lambda\,(n-k)\,S}{(1-\lambda)\,(n_k-1) + \lambda\,(n-k)},$$

Note that the parameter λ controls the contribution between S and Σ_k whereas the parameter γ controls the estimation of the eigenvalues of Σ_k. This regularization can be incorporated in the Fisher's criterion from the within covariance matrix S_W , by denoting that $S_W = \frac{1}{n}\sum_{k=1}^{K} n_k\Sigma_k$.

Generalized singular value decomposition Recently, after the reformulation of the FDA problem in terms of generalized singular value decomposition (GSVD) [66, 141], Howland and Park [89] extended this approach to the singularity case of the pooled scatter matrix (S_W). Their approach is developed in different steps. First, they diagonalize simultaneously S_W and S_B according to the symmetric-definite generalized eigenvalue problem defined by [66]:

$$Z^t S_W Z = \mathbf{I}_p \quad \text{and} \quad Z^t S_B Z = \Lambda,$$

where $Z \in \mathbb{R}^{p\times p}$ and Λ is a $p \times p$ diagonal matrix which contains the eigenvalues of $S_W^{-1}S_B$. Once these both matrices diagonalized, Howland and Park showed that the maximum of J_1 can be achieved for:

$$U = Z\begin{pmatrix} \mathbf{I}_d \\ \mathbf{0} \end{pmatrix}, \tag{2.2.22}$$

whenever U contains the $d = \text{rank}\,(S_B)$ eigenvectors of $S_W^{-1}S_B$ corresponding to its d largest eigenvalues. Then, to deal with the singularity case, the authors redefined three matrices

which recall the partitioning of the whole dataset into K clusters:

$$H_W = \frac{1}{\sqrt{n}}\left[Y_1 - m_1\mathbf{1}^t_{n_1}, \ldots, Y_K - m_K\mathbf{1}^t_{n_K}\right] \in \mathbb{R}^{p\times n} \quad (2.2.23)$$

$$H_B = \frac{1}{\sqrt{n}}\left[\sqrt{n_1}\,(m_1 - \bar{y}), \ldots, \sqrt{n_K}\,(m_K - \bar{y})\right] \in \mathbb{R}^{p\times K} \quad (2.2.24)$$

$$H_M = \frac{1}{\sqrt{n}}\left[y_1 - \bar{y}, \ldots, y_K - \bar{y}\right] = \frac{1}{\sqrt{n}}\left[Y - \bar{y}\mathbf{1}^t_n\right] \in \mathbb{R}^{p\times n} \quad (2.2.25)$$

where Y_k stands for the block data which belongs to the class k, $\mathbf{1}_n$ is a n-dimensional column vector containing ones, $m_k = \frac{1}{n_k}\sum_{i\in\mathcal{C}_k} y_i$ and n_k are respectively the mean and the number of patterns in the class k and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Thus, the scatter matrices can be re-expressed as:

$$S_W = H_W H_W^t, \quad S_B = H_B H_B^t \quad \text{and} \quad S = H_M H_M^t. \quad (2.2.26)$$

By considering the matrices defined by the equations (2.2.23),(2.2.24) and (2.2.25) and according to a general theorem developed by Paige and Saunders [141], Howland and Park reformulated the eigendecomposition problem in (2.2.17) as a problem that can be solved by a generalized singular value decomposition (GSVD):

$$\beta_j^2 H_B H_B^t \mathbf{z}_j = \alpha_j^2 H_W H_W^t \mathbf{z}_j, \quad (2.2.27)$$

where the α_i's and β_i's satisfy several conditions which are detailed in [89]. Howland and Park concluded that only the d first columns of Z which correspond to the d largest $\lambda_i = \frac{\alpha_i^2}{\beta_i^2}$ are needed and those form the projection matrix U. Moreover, they proved that the rule which computes the projection matrix U is the same in the non-singular case as in the singular case. To that end, they proposed a LDA/GSVD algorithm based on a partial GSVD of the matrix pair $\left(H_B^t, H_W^t\right)$ which can be applied even though S_W is singular. However, one limitation of this method is the high computational cost of GSVD particularly for large and high-dimensional data.

Dealing with the $n < p$: In the same aim to deal with the case of $n < p$, Zhang *et al.* [192] modified and improved Howland and Park's approach. Indeed, they first add a regularized term in the covariance matrix S which transforms the Fisher's criterion in $\left(S + \sigma^2\mathbf{I}_p\right)^{-1} S_B Z = Z\Lambda$. Secondly, by using the Howland's representations of S and S_B, Zhang *et al.* express the eigendecomposition problem as:

$$\left(H_M H_M^t + \sigma^2\mathbf{I}_p\right)^{-1} H_B H_B^t Z = Z\Lambda, \quad (2.2.28)$$

which enable them to deal with the $n \ll p$ problem. Indeed, Zhang *et al.* showed that the following equality $\left(H_M H_M^t + \sigma^2\mathbf{I}_p\right)^{-1} H_M = H_M\left(H_M^t H_M + \sigma^2\mathbf{I}_n\right)^{-1}$holds which is very interesting in terms of computations since the size of the inverse matrix $\left(H_M H_M^t + \sigma^2\mathbf{I}_p\right)^{-1}$ is $p \times p$ whereas those of $\left(H_M^t H_M + \sigma^2\mathbf{I}_n\right)^{-1}$ is $n \times n$. Consequently, when $n < p$ the compu-

tation cost of their method is reduced compared to Howland's one. furthermore, Zhang *et al.* proposed an efficient algorithm to deal with their approach and have extend it for a kernel approach of Fisher discriminant analysis.

An other recent approach was suggested by Ye [186] which proposed a new optimization criterion for discriminant analysis. This new approach extends the traditional FDA to the undersampled problem by combining tools used by Howland and Park in [89] and the pseudoinverse regularization. The proposed criterion J_2 is based on a regularization of J_1, such that:

$$J_2 = \text{trace}\left(\left(U^t S U\right)^+ U^t S_B U\right), \tag{2.2.29}$$

where $\left(U^t S U\right)^+$ denotes the pseudoinverse of the total class matrix in the lower dimensional subspace. His main idea is based on the simultaneous diagonalization of the three covariance matrices S, S_W and S_B which conduces him to an equivalent solution but more general to (2.2.22). This generalization leads Ye, to propose two very simple and efficient algorithms: on the one hand, he declines the uncorrelated LDA (ULDA) algorithm which has the property that the features in the reduced space are uncorrelated. The proposed algorithm is a natural extension of the Fisher discriminant analysis by replacing the inverse by the pseudoinverse. On the other hand, Ye proposes an alternative to Fisher discriminant analysis with orthogonal discriminant vectors yielding to the orthogonal linear discriminant analysis (OLDA) algorithm.

2.2.2.4 Fisher criterion as a regression-type problem

Qiao *et al.* [147] recently transformed the eigendecomposition problem defined in equation (2.2.11), as a ridge regression-type problem. Indeed, by considering the matrices H_W and H_B, defined previously in equations (2.2.23) and (2.2.24), and by defining the Cholesky decomposition of the within covariance matrix S_W such as $S_W = R_W^t R_W$ where R_W is an upper triangular matrix of dimension $p \times p$, Qiao *et al.* proposes the following theorem which allows to rewrite the eigendecomposition problem as a regression-type problem:

Theorem 2.2.1. Consider the Cholesky decomposition of the within covariance matrix $S_W = R_W^t R_W$ where $R_W \in \mathbb{R}^{p\times p}$ is a upper triangular matrix. Let $H_B \in \mathbb{R}^{p\times K}$ be defined as 2.2.24. Let $U_1, \ldots, U_d$ d column vectors of dimension p for which $d \leq \min(p, K-1)$ denote the eigenvectors linked to d largest values of the eigendecomposition of $S_W^{-1} S_B$. Let consider the $p \times d$ matrices $A = [\alpha_1, \ldots, \alpha_d]$ and $B = [\beta_1, \ldots, \beta_d]$. For $\rho > 0$, let $\hat{A}$ and $\hat{B}$ be the solutions of the following problem:

$$\min_{A,V} \sum_{k=1}^{K} \left\| R_W^{-t} H_{B,k} - A B^t H_{B,k} \right\|_F^2 + \rho \sum_{j=1}^{d} \beta_j^t S_W \beta_j \text{ w.r.t. } A^t A = \mathbf{I}_d, \tag{2.2.30}$$

where $H_{B,k} = \sqrt{n_k/n}\,(m_k - \bar{y})$ is the kth column of H_B and $\|.\|_F$ stands for the Frobenius norm. Then, the d column vectors $\hat{\beta}_j$ span the same linear space as those of the projection

matrix U.

Qiao *et al.* proved this theorem by computing alternatively the derivatives of expression (2.2.30) with respect to B given A and conversely, with respect to A given B. By considering the eigendecomposition of the matrix $R_W^{-t} S_B R_W^{-1} = E\Lambda E^t$ with E, the matrix containing the d associated eigenvectors and Λ the diagonal matrix containing its eigenvalues, then the optimal loadings matrix $\hat{A}$ satisfies:

$$\hat{A} = EP, \tag{2.2.31}$$

where P is an arbitrary $d \times d$ orthogonal matrix. By using such estimation for A, the optimal loadings matrix $\hat{B}$ is therefore:

$$\hat{B} = R_W^{-1} E(\Lambda + \rho \mathbf{I})^{-1} \Lambda P,$$

with $\rho > 0$. By remarking that the d column vectors of $U = R_W^{-1} E$ is solution of the generalized eigenvalue problem defined in (2.2.17) with $S_W = R_W^t R_W$, it allows the authors to conclude: the d column vectors of the fitted matrix $\hat{B}$ span the same linear subspace as the column vectors of U, solution of the eigendecomposition problem.

Note that the positive constant term ρ in the formula (2.2.30) of the Qiao's theorem stands for the ridge penalty term: when $n > p$, this theorem does not require a positive ρ. However, if $p > n$ and $\rho = 0$, ordinary multiple regression has no unique solution. This discrepancy is then removed by an additional positive ridge penalty term by posing $\rho > 0$.

2.2.2.5 Extension to unsupervised classification

Since clustering approaches are sensitive to high-dimensional and noisy data, recent works focused on combining low dimensional discriminative subspace with one of the most used clustering algorithm: k-means. This method iteratively computes a discriminative subspace based on Fisher criterion given the previous partition and obtains a new partition by k-means subject to this subspace. The first basic algorithm was proposed by Xu *et al.* [40] and then extended by De la Torre [110] who develops a discriminative cluster analysis (DCA) method in the case of non invertible scatter matrix. A theoretical framework is suggested by Ding in [47] when both tasks perform simultaneously since Fisher discriminant analysis and k-means clustering optimize the same objective function. More recently, Ye *et al.* [187] reformulate the iterative problem of clustering in discriminative subspace and show that the iterative subspace selection and k-means clustering is equivalent to kernel k-means task with a specific kernel Gram matrix. However, these approaches do not really compute the discriminant subspace and are not interested in the visualization of the data.

2.3 The subspace clustering

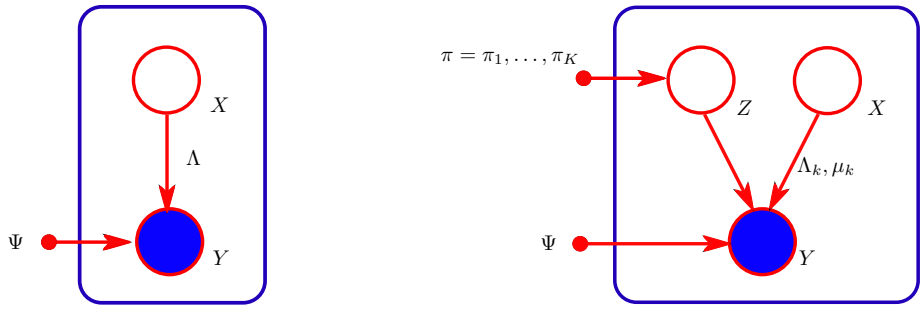
Finally, an other way which combines dimension reduction and clustering is the use of subspace clustering methods, proposed in the past few years. These methods exploit the "empty space" phenomenon since they consist of modeling the data of each group in specific subspaces and introduce some restrictions to ease the discrimination of the groups while keeping all dimensions.

Subspace clustering methods can be split into two categories: heuristic and probabilistic methods. Heuristic methods use algorithms to search for subspaces of high density within the original space. On the one hand, bottom-up algorithms use histograms for selecting the variables which best discriminate the groups. The Clique algorithm [1] was one of the first bottom-up algorithms and remains a reference in this family of methods. On the other hand, top-down algorithms use iterative techniques which start with all original variables and remove at each iteration the dimensions without groups. A review on heuristic methods is available in [143]. However, in this Section, we are going to focus on the second category of the subspace clustering methods which are based on a probabilistic framework. These methods assume that the data of each group live in a low-dimensional latent space and usually model the data with a generative model. Earlier strategies [153] are based on the factor analysis model which assumes that the latent space is related with the observation space through a linear relationship. This model was recently extended in [8, 128] and yields in particular the well known mixture of probabilistic principal component analyzers [19]. Recent works [22, 129] propose two families of parsimonious and regularized Gaussian models which partially encompass previous approaches. All these techniques turn out to be very efficient in practice, to cluster high-dimensional data.

2.3.1 Mixture of factor analyzers (MFA)

Mixture of factor analyzers [64, 127] is one of the subspace clustering method which both clusters the data and reduces locally the dimensionality of each cluster. Even though many authors extended the MFA model, the main idea was firstly introduced by Ghahramani and Hinton [64], and then extended by McLachlan *et al.* [127]. In this first paragraph, the original work of Ghahramani and Hinton on MFA is introduced before developing the works of Baek and McLachlan on this subject and other recent works [8, 136, 188, 189]. To distinct these both approaches, we recall G-MFA and M-MFA, the MFA approach developed by Guahramani and Hinton and respectively by McLachlan *et al.*.

The G-MFA model is an extension of the factor analysis (FA) model introduced in Section 2.2.1.2 to a mixture of K factor analyzers. Let $\{y_1, \ldots, y_n\}$ be independent observed realizations of a random vector $Y \in \mathbb{R}^p$. Let us also consider that Y can be expressed by an unobserved random vector $X \in \mathbb{R}^d$ named the factor and described in a lower dimensional space of dimension $d < p$. Moreover, $z_i = \{z_{i1}, \ldots, z_{iK}\}$ are assumed to be independent unobserved realizations of a random vector $Z \in \{0,1\}^K$ where $z_{ik} = 1$ if the data point is



(a) FA model (b) G-MFA model.

Figure 2.7: Graphical summary of factor analysis (FA) model (a) and mixture of factor analyzers (G-MFA) model of Ghahramani and Hinton (b).

generated by the kth factor analyzer and 0 otherwise. The relationship between these two spaces is assumed to be linear and, conditionally to the kth factor analyzer $Z = k$, it follows that:

$$Y_{|Z=k} = \Lambda_k X + \mu_k + \varepsilon, \tag{2.3.1}$$

where Λ_k is a $p \times d$ matrix and depicts the kth factor analyzer matrix and $\mu_k \in \mathbb{R}^p$ is the mean vector of the kth factor analyzer. Moreover $\varepsilon \in \mathbb{R}^p$ is a centered Gaussian noise term with a diagonal covariance matrix Ψ which is common to all factor analyzers:

$$\varepsilon \sim \mathcal{N}(0, \Psi). \tag{2.3.2}$$

Besides, as in standard FA, the factors $X \in \mathbb{R}^d$ are assumed to be distributed according to a Gaussian density function such as $X \sim \mathcal{N}(0, \mathbf{I}_d)$. This implies that the conditional distribution of Y is also Gaussian:

$$Y_{|X,Z=k} \sim \mathcal{N}(\Lambda_k X + \mu_k, \Psi), \tag{2.3.3}$$

The marginal density of Y is a Gaussian mixture model such as $f(y) = \sum_{k=1}^{K} \pi_k \phi(y; \theta_k)$, where π_k stands for the mixture proportion, $\phi(.)$ is a Gaussian density with parameters $\theta_k = \{\mu_k, \Lambda_k \Lambda_k^t + \Psi\}$. At this point, two main differences can be stated between G-MFA and the standard factor analysis. On the one hand, the G-MFA model considers a mixture of factor analyzers: it allows to have different local factor models, in different regions of the input space, compared to the standard FA which assumes a common factor model. On the other hand, conversely to the FA model, for which the mean of the data has no interest and the model is fitted on $Y - \mu$, in the G-MFA model, each factor analyzer has different means μ_k which allows each of them to model the data covariance structure in a different part of the observation space. Figures 2.7.a and 2.7.b summarize respectively the FA and G-MFA models. The complexity of the G-MFA model can be computed according to the number of parameters to estimate. Since the G-MFA model is in a Gaussian mixture model of K components, there are $(K-1)$ parameters for the proportions and Kp for the means. Moreover,

$Kd(p-(d-1)/2)+p$ parameters are required to estimate the component-covariance matrices, since these covariances matrices are defined in a factor representation such as $S_k = \Lambda_k \Lambda_k^t + \Psi$. The model complexity is then $\gamma_{G-MFA} = (K-1) + Kp + Kd(p-(d-1)/2) + p$ and by considering the following numerical example $p = 100$, $K = 4$ and $d = 3$, then 1691 parameters have to be estimated for this G-MFA model.

This approach introduced by Ghahramani and Hinton was generalized by McLachlan *et al.* [128] by removing the constraint on the variance of the noise. Therefore, the conditional distribution of the noise term becomes $\varepsilon_{|Z=k} \sim \mathcal{N}(0, \Psi_k)$ where Ψ_k stands for the diagonal matrix of the cluster k. The conditional distribution of Y is then: $Y_{|X,Z=k} \sim \mathcal{N}(\Lambda_k X + \mu_k, \Psi_k)$. In this case, since there are K covariance matrices of noise to compute in comparison to the Ghahramani's MFA, the model complexity increases and takes the following expression: $\gamma_{M-MFA} = (K-1) + Kp + Kd(p-(d-1)/2) + Kp$.

More recently, McLachlan and Baek [8] provided an alternative approach which aims to improve the complexity of the model by proposing a more parsimonious model. To that end, they re-parametrized the mixture model with restrictions on the means, such as $\mu_k = A\rho_k$ where A is a $p \times d$ orthonormal matrix ($A^t A = \mathbf{I}_d$) and ρ_k is a d-dimensional vector, and on the covariance matrix $S_k = A\Omega_k A^t + \Psi$, where Ω_k is a $d \times d$ positive definite symmetric matrix and Ψ a diagonal $p \times p$ matrix. This model is referred to by the mixture of factor analyzers with common factor loadings (MCFA) by its authors, as the matrix A is common to the factors. According to the MCFA assumptions, there are only Kd means parameters to estimate instead of Kp in the MFA model. Moreover, since the matrix A is constrained to have orthonormal columns and to be common to all classes, then only $pd - d(d+1)/2$ loadings are required to estimates it. Finally, according to the restriction on the matrices Ω_k, the number of parameters to estimate these K matrices is $Kd(d+1)/2$. Consequently, the complexity of the MCFA model is: $\gamma_{MCFA} = (K-1) + Kd + p + (pd - d(d+1)/2)) + Kd(d+1)/2$ and for the numerical example, this complexity is equal to 433 which is much more parsimonious than the previous MFA models. Besides, this MCFA approach is a special case of the MFA model but has the main advantage to allow the data to be displayed in a common low-dimensional plot. The MCFA approach is also a generalization of the works of Yoshida *et al.* [188, 189] since these authors, in their approach, constrained the covariance of the noise term to be spherical ($\Psi = \lambda \mathbf{I}_p$) and the component-covariance matrices of the factors to be diagonal ($\Omega_k = \Delta_k = \text{diag}(\sigma_{k1}^2, \ldots, \sigma_{kd}^2)$). This approach, called mixtures of common uncorrelated factor spherical-error analyzers (MCUFSA), is therefore more parsimonious than MCFA according to the additional assumptions done on the parameters of the MFA model. More recently, Montanari and Viroli [136] presented an approach called heteroscedastic mixture factor model (HMFA) which is very similar to the model described in MCFA. Their model differs from the MCFA approach only on the definition of the common loadings matrix A which does not need to have orthonormal columns. However, to obtain a unique solution for the matrix A, Montanari and Viroli added restrictions on this matrix such as $A^t \Psi^{-1} A$ is

Model name	Cov. structure	Nb. of parameters	$K=4$ $d=3$ $p=100$
M-MFA	$S_k = \Lambda_k \Lambda_k^t + \Psi_k$	$(K-1) + Kp + Kd[p-(d-1)/2] + Kp$	1991
G-MFA	$S_k = \Lambda_k \Lambda_k^t + \Psi$	$(K-1) + Kp + Kd[p-(d-1)/2] + p$	1691
MCFA	$S_k = A\Omega_k A^t + \Psi$	$(K-1) + Kd + p + d[p-(d+1)/2] + Kd(d+1)/2$	433
HFMA	$S_k = V\Omega_k V^t + \Psi$	$(K-1) + (K-1)d + p + d[p-(d-1)/2] + (K-1)d(d+1)/2$	427
MCUFSA	$S_k = A\Delta_k A^t + \lambda \mathbf{I}_p$	$(K-1) + Kd + 1 + d[p-(d+1)/2] + Kd$	322

A is defined such as $A^t A = \mathbf{I}_d$, V such as $V\Psi^{-1}V^t$ is diagonal with decreasing order and Δ_k is a diagonal matrix.

Table 2.3: Nomenclature of the MFA models developed by Ghahramani and Hinton (G-MFA), MacLachlan *et al.* (M-MFA), and MCFA models with their corresponding covariance structure.

diagonal with elements in decreasing order.

The differences between these MFA models are summarized in Table 2.3 which presents both the covariance structure and the model complexity of each approach.

2.3.2 Parsimonious Gaussian Mixture Model (PGMM)

More recently, a general framework for the MFA model was proposed by McNicholas and Murphy [129] which, in particular, included the previous works of Ghahramani and Hinton and of McLachlan *et al.*[128].

By considering the previous framework defined by the assumptions (2.3.1,2.3.3), McNicholas and Murphy[130] proposed a family of 12 models known as the expanded parsimonious Gaussian mixture model (EPGMM) family. by constraining the terms of the covariance matrix to be equal or not, by considering an isotropic variance for the noise term, or by reparametrizing the factor analysis covariance structure, they decline 12 EPGMM models. The nomenclature of both PGMM and EPGMM is illustrated in Table 2.4 in which the covariance structure of each model is detailed. In particular, the terminology of the PGMM family is as following: the first letter stands for the loading matrix which is constrained to be equal between groups (C..) or not (U..), and the last terms are linked to the error variance. This variance can be common between factors (.C.) or not (.U.) corresponding to the second term and the last term denotes the covariance structure which can be either isotropic (..C) or not (..U). Thus, the CCC model refers to by a model with common factors ($\Lambda_k = \Lambda$, $\forall k \in \{1, \ldots, K\}$) and a common and isotropic noise variance ($\Psi_k = \psi \mathbf{I}_p$). In the terminology of the EPGMM family, the main difference remains in the 3 last terms which correspond to the noise variance structure and are a combination of the following constraints: Δ_k can be common (.C..) or not (.U..),

Model name	Cov. structure	Nb. of parameters	$K=4, d=3$ $p=100$
UUUU - UUU	$S_k = \Lambda_k \Lambda_k^t + \Psi_k$	$(K-1) + Kp + Kd[p-(d-1)/2] + Kp$	1991
UUCU -	$S_k = \Lambda_k \Lambda_k^t + \omega_k \Delta_k$	$(K-1) + Kp + Kd[p-(d-1)/2] + [1 + K(p-1)]$	1988
UCUU -	$S_k = \Lambda_k \Lambda_k^t + \omega_k \Delta$	$(K-1) + Kp + Kd[p-(d-1)/2] + [K + (p-1)]$	1694
UCCU - UCU	$S_k = \Lambda_k \Lambda_k^t + \Psi$	$(K-1) + Kp + Kd[p-(d-1)/2] + p$	1691
UCUC - UUC	$S_k = \Lambda_k \Lambda_k^t + \psi_k \mathbf{I}_p$	$(K-1) + Kp + Kd[p-(d-1)/2] + K$	1595
UCCC - UCC	$S_k = \Lambda_k \Lambda_k^t + \psi \mathbf{I}_p$	$(K-1) + Kp + Kd[p-(d-1)/2] + 1$	1592
CUUU - CUU	$S_k = \Lambda \Lambda^t + \Psi_k$	$(K-1) + Kp + d[p-(d-1)/2] + Kp$	1100
CUCU -	$S_k = \Lambda \Lambda^t + \omega \Delta_k$	$(K-1) + Kp + d[p-(d-1)/2] + [1 + K(p-1)]$	1097
CCUU -	$S_k = \Lambda \Lambda^t + \omega_k \Delta$	$(K-1) + Kp + d[p-(d-1)/2] + [K + (p-1)]$	803
CCCU - CCU	$S_k = \Lambda \Lambda^t + \Psi$	$(K-1) + Kp + d[p-(d-1)/2] + p$	800
CCUC - CUC	$S_k = \Lambda \Lambda^t + \psi_k \mathbf{I}_p$	$(K-1) + Kp + d[p-(d-1)/2] + K$	704
CCCC - CCC	$S_k = \Lambda \Lambda^t + \psi \mathbf{I}_p$	$(K-1) + Kp + d[p-(d-1)/2] + 1$	701

where $\omega_k \in \mathbb{R}^+$and $|\Delta_k| = 1$.

Table 2.4: Nomenclature of the members of the PGMM and EPGMM families and the corresponding covariance structure.

$\omega_k = \omega \ \forall k \in \{1, \ldots, K\}$ (..C.) or not (..U.) and finally $\Delta_k = \mathbf{I}_p$ (...C) or not (...U). The table also gives the maximum number of free parameters to estimate according to K, p and d for the 12 models. In particular, this number of free parameters to estimate can be decomposed in the number of parameters to estimate for the proportions $(K-1)$, for the means (Kp) and for the covariance matrices (last terms).

According to this family of 12 models, the previous approaches developed by [64, 165, 128, 129, 8] become then submodels of the EPGMM approach. For example, by constraining only the noise variance to be isotropic on each class ($\Psi_k = \sigma_k^2 \mathbf{I}_p$) which corresponds to the CUC and CCUC models, it produces Mixt-PPCA. In the same way, by considering the covariance structure of the UCU and UCCU models such that $\Psi_k = \Psi$ and Λ_k, then we obtain the mixture of factor analyzers model developed by Ghahramani and Hinton and the UUUU model is equivalent to the MFA model proposed by McLachlan *et al.* in [128]. Finally, by parametrizing the factor analysis covariance structure by writing $\Psi_k = \omega_k \Delta_k$ where Δ_k is a diagonal matrix and $|\Delta_k| = 1$, McNicholas and Murphy proposed four additional models to their previous work [129] according to the restrictions on the covariance structure and also to the following constraint $\omega_k = \omega, \forall k$.

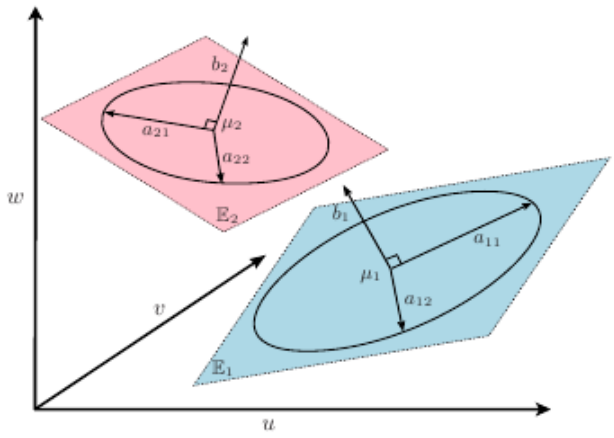


Figure 2.8: Parametrization of the model $[a_{kj}b_kQ_kd_k]$ for the HDDC approach.

In the case of EPGMM models, the parameter estimation is done according to alternating expectation-conditional maximization algorithms (AECM) [133] wherein the M-step is replaced by few conditional maximization steps since the group membership and the latent factors are both unknown.

2.3.3 High-dimensional GMM (Hd-GMM)

In a slightly different context, Bouveyron *et al.* [22, 23] proposed a family of 28 parsimonious and flexible Gaussian models to deal with high-dimensional data. Conversely to the previous approaches, this family of GMM was proposed in both supervised and unsupervised classification context. However in this subsection, we consider only the unsupervised context and to ease the designation of this family, we propose to recall these models: the high-dimensional Gaussian mixture models (Hd-GMM). By considering the model-based clustering framework presented in Section 2.1.1.1, Bouveyron *et al.* [22] proposed to rewrite the covariance matrix S_k of the class k by using an eigendecomposition of S_k as it was firstly proposed by Banfield and Raftery [9]:

$$S_k = Q_k \Lambda_k Q_k^t,$$

where Q_k is a $p \times p$ orthogonal matrix which contains the eigenvectors of S_k and Λ_k is a $p \times p$ diagonal matrix containing the eigenvalues of S_k. The key idea of the work of Bouveyron *et al.* is to reparametrize the matrix Λ_k, such as Λ_k models each group in a subspace of lower dimension than the dimension of the observed space such that:

$$\Lambda_k = \text{diag}\left(a_{k1}, \ldots, a_{kd_k}, b_k, \ldots, b_k\right),$$

where the d_k first values $a_{k1}, \ldots, a_{kd_k}$ parametrize the variance in the group-specific subspace and the $p - d_k$ last terms, the b_k's, model the noise. With this parametrization, these parsimo-

Model name	Nb. of parameters	$p = 100$ $K = 4$, $d = 3$
$[a_{kj}b_kQ_kd_k]$	$(K-1) + Kp + \sum_{k=1}^{K} d_k[p - (d_k+1)/2] + \sum_{k=1}^{K} d_k + 2K$	1599
$[a_{kj}bQ_kd_k]$	$(K-1) + Kp + \sum_{k=1}^{K} d_k[p - (d_k+1)/2] + \sum_{k=1}^{K} d_k + 1 + K$	1596
$[a_kb_kQ_kd_k]$	$(K-1) + Kp + \sum_{k=1}^{K} d_k[p - (d_k+1)/2] + 3K$	1591
$[ab_kQ_kd_k]$	$(K-1) + Kp + \sum_{k=1}^{K} d_k[p - (d_k+1)/2] + 1 + 2K$	1588
$[abQ_kd_k]$	$(K-1) + Kp + \sum_{k=1}^{K} d_k[p - (d_k+1)/2] + 2 + K$	1585
$[a_{kj}b_kQ_kd]$	$(K-1) + Kp + Kd[p - (d+1)/2] + Kd + K + 1$	1596
$[a_jb_kQ_kd]$	$(K-1) + Kp + Kd[p - (d+1)/2] + d + K + 1$	1587
$[a_{kj}bQ_kd]$	$(K-1) + Kp + Kd[p - (d+1)/2] + Kd + 2$	1593
$[a_jbQ_kd]$	$(K-1) + Kp + Kd[p - (d+1)/2] + d + 2$	1584
$[a_kb_kQ_kd]$	$(K-1) + Kp + Kd[p - (d+1)/2] + 2K + 1$	1588
$[ab_kQ_kd]$	$(K-1) + Kp + Kd[p - (d+1)/2] + K + 2$	1585
$[a_kbQ_kd]$	$(K-1) + Kp + Kd[p - (d+1)/2] + K + 2$	1585
$[abQ_kd]$	$(K-1) + Kp + Kd[p - (d+1)/2] + 3$	1582
$[a_jbQd]$	$(K-1) + Kp + d[p - (d+1)/2] + d + 2$	702
$[abQd]$	$(K-1) + Kp + d[p - (d+1)/2] + 3$	700

Table 2.5: Nomenclature for the members of the Hd-GMM family and the number of parameters to estimate. For the numerical example, the intrinsic dimension of the clusters has been fixed to $d_k = \bar{d} = 3$, $\forall k = 1, \ldots, K$.

nious models assume that conditionally to the groups, the noise variance of each cluster k is isotropic and is contained in a subspace which is orthogonal to the subspace of the kth group. The authors proposed a family of parsimonious models from a very general model, referred to as $[a_{kj}b_kQ_kd_k]$ to very simple models. Figure 2.8 illustrates the parametrization of the model $[a_{kj}b_kQ_kd_k]$ and Table 2.5 stands for the nomenclature of the main Hd-GMM models and their complexity. In particular, the first quantity $(K-1)+Kp$ stands for the number of parameters for the means and the mixture proportions of K clusters. Then, there are $\sum_{k=1}^{K} d_k[p(d_k+1)/2]$ loadings to estimate for the K orientation matrices Q_k and finally the last terms represent the parameters for the covariance matrices in the latent and in the noise subspaces of K clusters and their intrinsic dimension. Such approach can be viewed in two different ways: on the one hand, these models enable to regularize the models in high-dimension. In particular, by

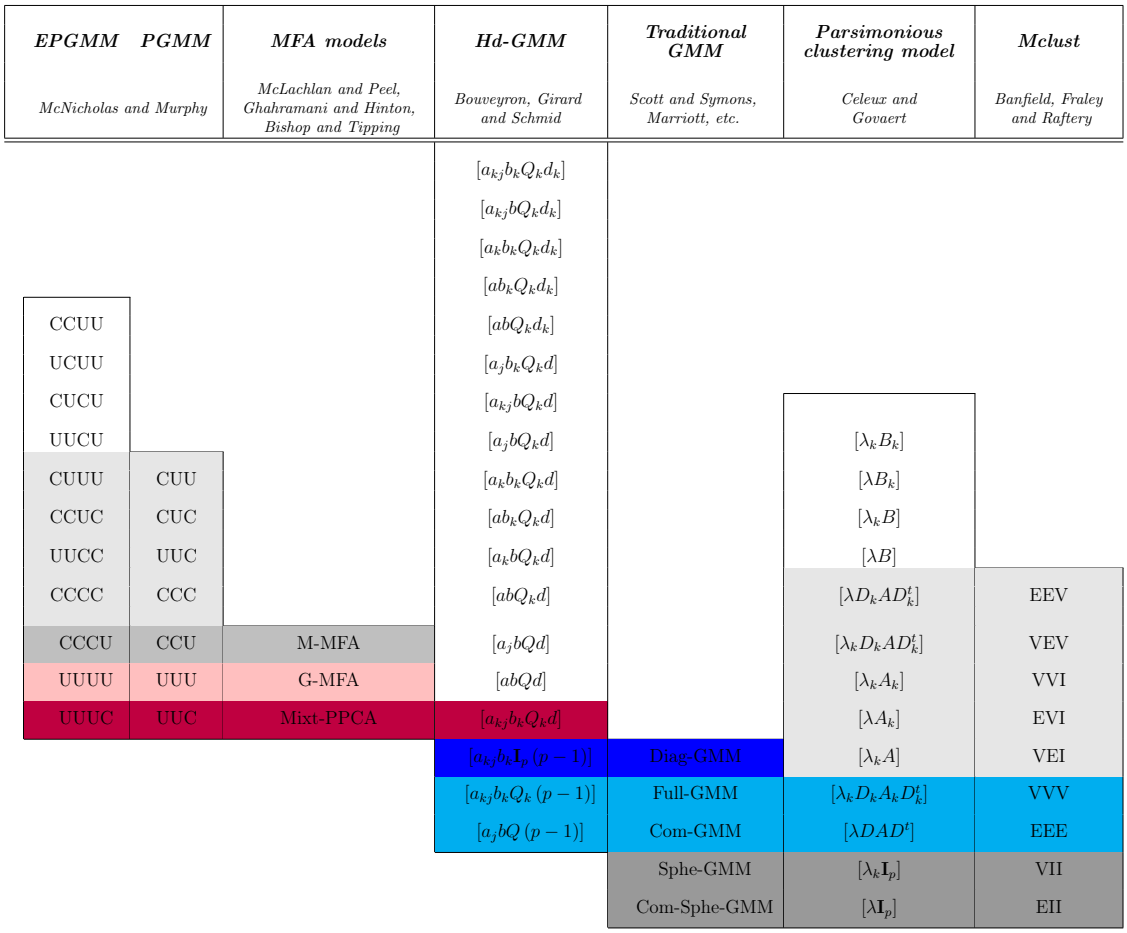


Figure 2.9: Links between existing subspace clustering and parsimonious models.

imposing a constraint on $d_k = p - 1$, the proposed approach can be viewed as an extension of the works of [36, 59]. Indeed, the model $[a_{kj}b_kQ_k\,(p-1)]$ stand for the Full-GMM model or the $[\lambda_kD_kA_kD_k]$ model of [36]. In the same manner, the model $[a_{kj}b_kQ\,(p-1)]$ is equivalent to the Diag-GMM or the $[a_jbQ\,(p-1)]$ is also the Com-Diag-GMM. On the other hand, this approach can also be viewed as an extension of the mixture of principal component analyzer (Mixt-PPCA) model [165] since it relaxes the equality assumption on d and the Hd-GMM model referred to by $[a_{kj}b_kQ_kd]$ stands for the Mixt-PPCA model.

2.3.4 Comparison and limitations of the subspace clustering methods

The subspace clustering methods which have been presented until now belong to a huge family of Gaussian mixture models and several links exist between these approaches. Indeed, in the last paragraph, we saw that some constrained Hd-GMM models were equivalent to traditional parsimonious models such that Com-Diag-GMM or Diag-GMM and consequently to the models proposed by Raftery and Fraley or by Celeux and Govaert. Besides, it appears

that the model $[a_{kj}b_kQ_kd]$ of the Hd-GMM models stands for the Mixt-PPCA model. In the same manner, few models which belong to the EPGMM family of [130] are also included in the HDDC family. In particular, the $UCUC$ model of [130] which corresponds to the model $[a_{kj}b_kQ_kd]$. Moreover, the EPGMM family proposed by McNicholas and Murphy included individual works on MFA particularly the works of Guahramani [64], Tipping and Bishop [165], McLachlan [128], McNicholas and Murphy [130] and Baek *et al.*[8] which become submodels. However, it seems difficult to compare these methods since the hypothesis of Hd-GMM models are stronger than the MFA models. Indeed, the subspace of each class is spanned by orthogonal vectors, whereas it is not a necessary condition in MFA, even if such a situation can occur sometimes as in the case of the model UCUC (CUC). The different links between the different families of parsimonious models are presented in Figure 2.9.

However, despite their efficiency to cluster high-dimensional data, these probabilistic methods based on subspace clustering present several limitations. Indeed, these approaches is based on the fact that the clustering results do not provide a simple understanding neither a global visualization of the clusters. Since each cluster lives in a different subspace, then the visualization of all the clusters in the same subspace seems difficult or even impossible. Only MCFA, MCUFSA or HMFA for which the factor loadings are formulated *via* common matrix for the component factor loadings can provide low-dimensional plots of the data structure. Finally, in the case where the subspace of each group is constrained to be common, the models choose the orientation such as the variance of the projected data is maximum which can be not sufficient to catch discriminative information. To overcome these limitations, we propose in the next Section a family of parsimonious models.

Chapter 3

Model-based clustering in a discriminative subspace

Among the previous approaches dealing with high-dimensional data, each of them presents certain limitations. In particular, when the dimension reduction is operated before the clustering task, the discriminative information can be ousted from the classification task. Chang [38] showed earlier, that the principal components linked to the largest eigenvalues do not necessary contain the most relevant information about the group structure of the dataset. Therefore, the selection of a subset of principal components can lead to a loss of discriminative information about the groups, in the data. Moreover, the main methods, which reduce the dimensionality by taking into account information about the group structure, occur often in a supervised classification context which seems to be useless in our approach. Finally, in the case of subspace clustering, since these methods model each group in a specific subspace, they are not able to provide a global visualization of the clustered data, which could be helpful for the practitioner.

Thus, in this section we propose a new statistical framework which aims to simultaneously cluster and reduce the dimension, such as the new axes well discriminate the groups. The main purpose is to model and cluster the data into a common latent subspace which both best discriminates the groups according to the current fuzzy partition of the data and has an intrinsic dimension which is lower than the dimension of the observation space.

3.1 The discriminative latent mixture model

This section introduces a mixture model, called the discriminative latent mixture model, which tries to find both a parsimonious and discriminative fit for the data, in order to generate a clustering and a visualization of the data. The proposed modeling is mainly based on two key ideas: firstly, actual data are assumed to live in a latent subspace with an intrinsic dimension lower than the dimension of the observed data; secondly, a subspace of $K-1$ dimensions is theoretically sufficient to discriminate K groups.

3.1.1 The $\text{DLM}_{[\Sigma_k\beta_k]}$ model

Let $\{y_1, \ldots, y_n\} \in \mathbb{R}^p$ denote a dataset of n observations that one wants to cluster into K homogeneous groups, *i.e.* adjoin to each observation y_i a value $z_i \in \{1, \ldots, K\}$ where $z_i = k$ indicates that the observation y_i belongs to the kth group. On the one hand, let us assume that $\{y_1, \ldots, y_n\}$ are independent observed realizations of a random vector $Y \in \mathbb{R}^p$ and that $\{z_1, \ldots, z_n\}$ are also independent realizations of a random variable $Z \in \{1, \ldots, K\}$. On the other hand, let $\mathbb{E} \subset \mathbb{R}^p$ denote a latent space assumed to be the most discriminative subspace of dimension $d \leq K - 1$ such that $\mathbf{0} \in \mathbb{E}$ and where d is strictly lower than the dimension p of the observed space. Moreover, let $\{x_1, \ldots, x_n\} \in \mathbb{E}$ denote the actual data, described in the latent space $\mathbb{E}$ of dimension d, which are in addition presumed to be independent unobserved realizations of a random vector $X \in \mathbb{E}$. Finally, for each group, the observed random vector $Y \in \mathbb{R}^p$ and the latent random vector $X \in \mathbb{E}$ are assumed to be linked through a linear transformation:

$$Y = UX + \varepsilon, \tag{3.1.1}$$

where $d < p$, U is the $p \times d$ orthogonal matrix common to the K groups, such as $U^t U = I_d$, and $\varepsilon \in \mathbb{R}^p$, conditionally to Z, is a centered Gaussian noise term with covariance matrix Ψ_k, for $k = 1, ..., K$:

$$\varepsilon_{|Z=k} \sim \mathcal{N}(\mathbf{0}, \Psi_k).$$

Following the classical framework of model-based clustering, each group is in addition assumed to be distributed according to a Gaussian density function within the latent space $\mathbb{E}$. Hence, the random vector $X \in \mathbb{E}$ has the following conditional density function:

$$X_{|Z=k} \sim \mathcal{N}(\mu_k, \Sigma_k),$$

where $\mu_k \in \mathbb{R}^d$ and $\Sigma_k \in \mathbb{R}^{d \times d}$ are respectively the mean and the covariance matrix of the kth group. Conditionally to X and Z, the random vector $Y \in \mathbb{R}^p$ has the following conditional distribution:

$$Y_{|X,Z=k} \sim \mathcal{N}(UX, \Psi_k),$$

and its marginal distribution is therefore a mixture of Gaussians:

$$f(y) = \sum_{k=1}^{K} \pi_k \phi(y; m_k, S_k),$$

where π_k is the mixture proportion of the kth group, $\phi(.)$ the Gaussian density function parametrized by:

$$\begin{aligned} m_k &= U\mu_k, \\ S_k &= U\Sigma_k U^t + \Psi_k, \end{aligned}$$

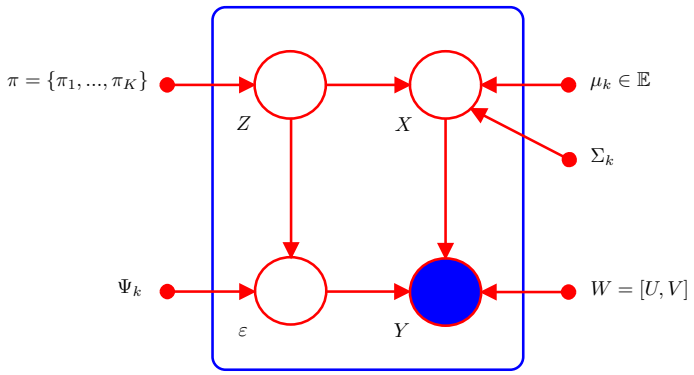


Figure 3.1: Graphical summary of the $\text{DLM}_{[\Sigma_k \beta_k]}$ model

which are respectively the mean and the covariance matrix of the kth group in the observation space. Let us also define $W = [U, V]$ a $p \times p$ matrix which satisfies $W^t W = W W^t = I_p$ and for which the $p \times (p-d)$ matrix V, is the orthonormal complement of U defined above. We finally assume that the non discriminative information covariance matrix Ψ_k satisfies the conditions $V \Psi_k V^t = \beta_k I_{d-p}$ and $U \Psi_k U^t = 0_d$, such that a $p \times p$ matrix Δ_k can be defined as:

$$\Delta_k = W^t S_k W \tag{3.1.2}$$

and has the following form:

$$\Delta_k = \begin{pmatrix} \boxed{\Sigma_k} & \mathbf{0} \\ \mathbf{0} & \boxed{\begin{matrix} \beta_k & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & \beta_k \end{matrix}} \end{pmatrix} \begin{matrix} \Big\} \\ \\ \Big\} \end{matrix} \begin{matrix} d \leq K-1 \\ \\ (p-d) \end{matrix}$$

This last assumption implies that the discriminative latent subspace and the non discriminative one are othogonal meaning that all the relevant clustering information remains in the latent subspace.

This model, called the discriminative latent mixture (DLM) model and referred to by $\text{DLM}_{[\Sigma_k \beta_k]}$ in the sequel, is summarized by Figure 3.1. The $\text{DLM}_{[\Sigma_k \beta_k]}$ model is therefore parametrized by the parameters π_k, μ_k, U, Σ_k and β_k, for $k = 1, ..., K$ and $j = 1, ..., d$. On the one hand, the mixture proportions $\pi_1, ..., \pi_K$ and the means $\mu_1, ..., \mu_K$ parametrize in a classical way the prior probability and the average latent position of each group respectively. On the other hand, U defines the latent subspace $\mathbb{E}$ by parametrizing its orientation according to the basis of the original space. Finally, Σ_k parametrizes the variance of the kth group within

the latent subspace $\mathbb{E}$ whereas β_k parametrizes the variance of this group outside $\mathbb{E}$. With these notations and from a practical point of view, one can say that the variance of the discriminative information is therefore modeled by Σ_k and the variance of the non discriminative information is modeled by β_k.

3.1.2 Complete log-likelihood of the $\text{DLM}_{[\Sigma_k\beta_k]}$ model

The complete log-likelihood of the $\text{DLM}_{[\Sigma_k\beta_k]}$ model is defined in the following proposition:

Proposition 3.1.1. *In the case of the model* $\text{DLM}_{[\Sigma_k\beta_k]}$*, the complete log-likelihood* $\ell(y_1,\ldots,y_n,\theta)$ *has the following expression:*

$$\begin{aligned}\ell(\theta) = -\frac{1}{2}\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + \text{trace}(\Sigma_k^{-1}U^tC_kU) + \log(|\Sigma_k|) \\ + (p-d)\log(\beta_k) + \frac{1}{\beta_k}(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^tC_ku_j) + \gamma\Big].\end{aligned} \tag{3.1.3}$$

where C_k *is the empirical covariance matrix of the* k*th group,* u_j *is the* j*th column vector of* U*,* $n_k = \sum_{i=1}^{n} z_{ik}$ *and* $\gamma = p\log(2\pi)$ *is a constant term.*

Proof. By considering $\{(y_1,z_1),\ldots,(y_n,z_n)\}$ the complete dataset, then the complete log-likelihood $\ell(y_1,\ldots,y_n,\theta)$ is:

$$\ell(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log\left(\pi_k\phi(y_i,\theta_k)\right),$$

where z_{ik} stands for the class membership and $z_{ik} = \mathbf{1}_{\{y_i\in\mathcal{C}_k\}}$. Then, in the case of the $\text{DLM}_{[\Sigma_k\beta_k]}$ model, the complete log-likelihood of the observed data can be rewritten in this way:

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log(\pi_k\phi(y_i,\theta_k)) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\Big[-\frac{1}{2}\log(|S_k|) - \frac{1}{2}(y_i-m_k)^tS_k^{-1}(y_i-m_k) + \log(\pi_k) - \frac{p}{2}\log(2\pi)\Big],\end{aligned} \tag{3.1.4}$$

where $z_{ik} = 1$ if the observation y_i belongs to the class k and $z_{ik} = 0$ otherwise. According to the definitions of the diagonal matrix $\Delta_k = W^tS_kW$ and of the orientation matrix W for which $W^{-1} = W^t$, the inverse covariance matrix S_k^{-1} of Y can be written as $S_k^{-1} = (W\Delta_kW^t)^{-1} = W^{-t}\Delta_k^{-1}W^{-1} = W\Delta_k^{-1}W^t$ and the determinant of S_k can be also reformulated in the following way:

$$|S_k| = |\Delta_k| = |\Sigma_k|\,\beta_k^{p-d}. \tag{3.1.5}$$

Consequently, the complete log-likelihood $\ell(\theta)$ can be rewritten as:

$$\begin{aligned}\ell(\theta) = &-\frac{1}{2}\sum_{k=1}^{K} n_k \Big[-2\log(\pi_k) + \log(|\Sigma_k|) + (p-d)\ \log(\beta_k) \\ &+ \frac{1}{n_k}\sum_{i=1}^{n} z_{ik}(y_i - m_k)^t W \Delta_k^{-1} W^t (y_i - m_k) + \gamma\Big].\end{aligned} \tag{3.1.6}$$

where $n_k = \sum_{i=1}^{n} z_{ik}$ and $\gamma = p\log(2\pi)$ is a constant term. At this point, two remarks can be done on the quantity $\sum_{i=1}^{n} z_{ik}(y_i - m_k)^t W \Delta_k^{-1} W^t (y_i - m_k)$. First, as this quantity is a scalar, it is equal to its trace. Secondly, this quantity can be divided in two parts since $W = [U, V]$ and $W = \tilde{W} + \bar{W}$, with $\tilde{W} = [U, \mathbf{0_{p-d}}]$ and $\bar{W} = [\mathbf{0_d}, V]$. Then, the relation $W\Delta_k^{-1}W^t = \tilde{W}\Delta_k^{-1}\tilde{W}^t + \bar{W}\Delta_k^{-1}\bar{W}^t$ is stated and we can write:

$$\begin{aligned}(y_i - m_k)^t W \Delta_k^{-1} W^t (y_i - m_k) \quad = \quad & \text{trace}\left((y_i - m_k)^t \tilde{W} \Delta_k^{-1} \tilde{W}^t (y_i - m_k)\right) \\ + \quad & \text{trace}\left((y_i - m_k)^t \bar{W} \Delta_k^{-1} \bar{W}^t (y_i - m_k)\right).\end{aligned}$$

Moreover, pointing out that $C_k = \frac{1}{n_k}\sum_{i=1}^{n} z_{ik}(y_i - m_k)(y_i - m_k)^t$ is the empirical covariance matrix of the kth group, the previous quantity can be rewritten as:

$$\frac{1}{n_k}\sum_{i=1}^{n} z_{ik}(y_i - m_k)^t W \Delta_k^{-1} W^t (y_i - m_k) = \text{trace}(\Delta_k^{-1}\tilde{W}^t C_k \tilde{W}) + \text{trace}(\Delta_k^{-1}\bar{W}^t C_k \bar{W})$$

and finally:

$$\frac{1}{n_k}\sum_{i=1}^{n} z_{ik}(y_i - m_k)^t W \Delta_k^{-1} W^t (y_i - m_k) = \quad \text{trace}(\Sigma_k^{-1} U^t C_k U) \quad + \quad \sum_{j=1}^{p-d} \frac{v_j^t C_k v_j}{\beta_k},$$

where v_j, is the jth column vector of V. However, since $\bar{W} = W - \tilde{W}$ and $W = [U, V]$, it is also possible to write:

$$\begin{aligned}\frac{1}{\beta_k}\sum_{j=1}^{p-d} v_j^t C_k v_j =& \frac{1}{\beta_k}\left(\sum_{j=1}^{p} w_j^t C_k w_j - \sum_{j=1}^{d} u_j^t C_k u_j\right) \\ =& \frac{1}{\beta_k}\left(\sum_{j=1}^{p} \text{trace}(w_j w_j^t C_k) - \sum_{j=1}^{d} u_j^t C_k u_j\right) \\ =& \frac{1}{\beta_k}\Big[\text{trace}(C_k) - \sum_{j=1}^{d} u_j^t C_k u_j\Big].\end{aligned}$$

Consequently, replacing this quantity in (3.1.6) provides the final expression of $\ell(\theta)$. □

3.1.3 Classification function of the $\text{DLM}_{[\Sigma_k\beta_k]}$ model

As the MAP rule is entirely defined by the classification function $\Gamma_k = -2\log(\pi_k\phi(y,\theta_k))$ defined in Chapter 2, then, in this Subsection, the classification function corresponding to the $\text{DLM}_{[\Sigma_k\beta_k]}$ is given below.

Proposition 3.1.2. *With the assumptions of the model* $\text{DLM}_{[\Sigma_k\beta_k]}$*, the classification function* $\Gamma_k(y_i)$ *for* $i = 1, ..., n$ *and* $k = 1, ..., K$*, can be expressed as :*

$$\begin{aligned}\Gamma_k(y) = \quad & ||\mu_k - P(y)||^2_{\mathcal{D}_k} + \frac{1}{\beta_k}||\,(y - P(y)) - (m_k - \mu_k)\,||^2 \\ & + \log(|\Sigma_k|) + (p-d)\log(\beta_k) - 2\log(\pi_k) + \gamma,\end{aligned} \tag{3.1.7}$$

where $||.||^2_{\mathcal{D}_k}$ *is a norm on the latent space* $\mathbb{E}$ *defined by* $||y||^2_{\mathcal{D}_k} = y^t\mathcal{D}_k y$*,* $\mathcal{D}_k = \tilde{W}\Delta_k^{-1}\tilde{W}^t$*,* $\tilde{W}$ *is a* $p \times p$ *matrix containing the* d *vectors of* U *completed by zeros such as* $\tilde{W} = [U, 0_{p-d}]$*,* P *is the projection operator on the latent space* $\mathbb{E}$*, i.e.* $P(y) = UU^t y$*, and* $\gamma = p\log(2\pi)$ *is a constant term.*

Proof. Let consider the expression of the classification function $\Gamma_k(y) = -2\log(\pi_k\phi(y,\theta_k))$ which is:

$$\Gamma_k(y) = (y - m_k)^t S_k^{-1}(y - m_k) + \log(|S_k|) - 2\log(\pi_k) + p\ \log(2\pi),$$

where $|.|$ represents the determinant. Since the assumption (3.1.2) and according to the definitions of the diagonal matrix Δ_k, it implies that the inverse covariance matrix S_k^{-1} of Y can be written as $S_k^{-1} = W\Delta_k^{-1}W^t$. Moreover, since the determinant of S_k can also be reformulated as in equation (3.1.5), then the cost function can be rewritten in this way:

$$\Gamma_k(y) = (y - m_k)^t W\Delta_k^{-1}W^t(y - m_k) + \log(|\Delta_k|) - 2\log(\pi_k) + p\ \log(2\pi).$$

According to the assumptions of the model $\text{DLM}_{[\Sigma_k\beta_k]}$ and given that $W = \tilde{W} + \bar{W}$ where $\tilde{W} = [U, 0_{p-d}]$ and $\bar{W} = [0_d, V]$ where the relation $W\Delta_k^{-1}W^t = \tilde{W}\Delta\tilde{W}^t + \bar{W}\Delta_k^{-1}\bar{W}^t$ is stated, then Γ_k can be reformulated as:

$$\begin{aligned}\Gamma_k(y) = \quad & (y - m_k)^t\tilde{W}\Delta_k^{-1}\tilde{W}^t(y - m_k) + (y - m_k)^t\bar{W}\Delta_k^{-1}\bar{W}^t(y - m_k) \\ & + \log(|\Delta_k|) - 2\log(\pi_k) + p\ \log(2\pi),\end{aligned}$$

Moreover, since the relations $\tilde{W}(\tilde{W}^t\tilde{W}) = \tilde{W}$ and $\bar{W}(\bar{W}^t\bar{W}) = \bar{W}$ hold due to the construction

of $\tilde{W}$ and $\bar{W}$, then:

$$\begin{aligned}\Gamma_k(y) = & \left(\tilde{W}\tilde{W}^t(y-m_k)\right)^t \tilde{W}\Delta_k^{-1}\tilde{W}^t\left(\tilde{W}\tilde{W}^t(y-m_k)\right) \\ & + \frac{1}{\beta_k}\left(\bar{W}\bar{W}^t(y-m_k)\right)^t\left(\bar{W}\bar{W}^t(y-m_k)\right) \\ & + \log(|\Delta_k|) - 2\log(\pi_k) + p\ \log(2\pi).\end{aligned}$$

Let us now define $\mathcal{D}_k = \tilde{W}\Delta_k^{-1}\tilde{W}^t$ and $||.||_{\mathcal{D}_k}$, a norm on the latent space spanned by $\tilde{W}$, such that $||y||^2_{\mathcal{D}_k} = y^t\mathcal{D}_k y$. With these notations, and according to the definition of Δ_k, Γ_k can be rewritten as:

$$\begin{aligned}\Gamma_k(y) = & ||\tilde{W}\tilde{W}^t(y-m_k)||^2_{\mathcal{D}_k} + \frac{1}{\beta_k}||\bar{W}\bar{W}^t(y-m_k)||^2 \\ & + \log(|\Sigma_k|) + (p-d)\ \log(\beta_k) - 2\log(\pi_k) + p\ \log(2\pi).\end{aligned}$$

Let us also define the projection operators P and $P^\perp$ on the subspaces $\mathbb{E}$ and $\mathbb{E}^\perp$ respectively:

- $P(y) = \tilde{W}\tilde{W}^t y$ is the projection of y on the discriminative space $\mathbb{E}$,
- $P^\perp(y) = \bar{W}\bar{W}^t y$ is the projection of y on the complementary space $\mathbb{E}^\perp$.

Consequently, the cost function Γ_k can be finally reformulated as:

$$\begin{aligned}\Gamma_k(y) = & \|P(y-m_k)\|^2_{\mathcal{D}_k} + \frac{1}{\beta_k}\left\|P^\perp(y-m_k)\right\|^2 \\ & + \log(|\Sigma_k|) + (p-d)\ \log(\beta_k) - 2\log(\pi_k) + p\ \log(2\pi).\end{aligned}$$

Since $P^\perp(y) = y - P(y)$, then the distance associated with the complementary subspace can be rewritten as $||P^\perp(y_i - m_k)||^2 = ||(y_i - m_k) - P(y_i - m_k)||^2$ such as:

$$\begin{aligned}\Gamma_k(y) = & ||P(y-m_k)||^2_{\mathcal{D}_k} + \frac{1}{\beta_k}||(y-m_k) - P(y-m_k)||^2 \\ & + \log\left(|\Sigma_k|\right) + (p-d)\log(\beta_k) - 2\log(\pi_k) + \gamma,\end{aligned}$$

and this allows to conclude. □

Firstly, Proposition 3.1.2 provides a comprehensive interpretation of the classification function Γ_k which mainly governs the MAP rule. Indeed, it appears that Γ_k mainly depends on two distances: firstly, the distance $\|P(y-m_k)\|^2_{\mathcal{D}_k}$ associated with the matrix $\mathcal{D}_k = \tilde{W}\Delta_k^{-1}\tilde{W}^t$ where $\tilde{W} = [U, \mathbf{0}_{p-d}]$, and stands for the distance between the projections on the discriminant subspace $\mathbb{E}$ of the observation y_i and the mean m_k. Secondly, the Euclidean distance $\frac{1}{\beta_k}\left\|P^\perp(y-m_k)\right\|^2$ corresponds to the distance between the projections on the complementary subspace $\mathbb{E}^\perp$ of y and m_k and is weighted by the variance of non discriminative information β_k. Therefore, the classification function Γ_k facilitates the affectation of an observation y to the cluster k if the projection of this observation, in the subspace $\mathbb{E}$, is close to the projected

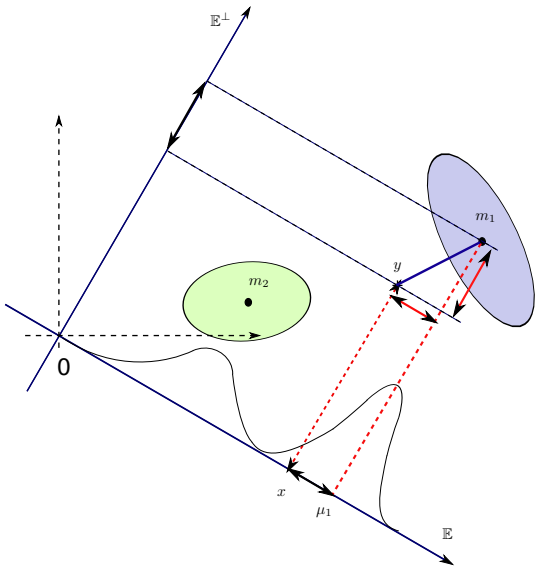


Figure 3.2: Two groups and their 1-dimensional discriminative subspace $\mathbb{E}$.

center of the cluster k, and if its projection in $\mathbb{E}^{\perp}$ is also close to the center, which seems quite natural. Obviously, these distances are also balanced by the variances in $\mathbb{E}$ and $\mathbb{E}^{\perp}$ and by the mixture proportions. For example, if the data are very noisy meaning that β_k is large then, the distance $\left\|P^{\perp}(y_i - m_k)\right\|$ from the point to the subspace $\mathbb{E}^{\perp}$ weighted by $1/\beta_k$ becomes smaller and consequently the MAP rule is mainly defined by what it happens in the discriminative subspace. Remark that the latter distance $\left\|P^{\perp}(y - m_k)\right\|$ can be reformulated in order to avoid the use of the projection on $\mathbb{E}^{\perp}$. Indeed, as Figure 3.2 illustrates, this distance can be re-expressed according projections on $\mathbb{E}$. Besides these geometrical aspects, there is as well a computational interest since the cost function does not require the use of the projection on the complementary subspace $\mathbb{E}^{\perp}$. Thus, there is no need to get Γ_k to compute the $p - d$ last terms of W *i.e.* the matrix V. This will provide the stability of the algorithm and will allow its use when $n < p$ (*cf.* paragraph 4.3.2).

3.1.4 Complexity of the $\text{DLM}_{[\Sigma_k \beta_k]}$ model

For the $\text{DLM}_{[\Sigma_k \beta_k]}$, it is necessary to estimate only one subspace of dimension $d < p$ since firstly, the classes are assumed to live in a common subspace and secondly, we have just seen, in the previous paragraph, that there is no need to estimate the $p - d$ columns of the matrix W to obtain the cost function Γ_k. Consequently, the complexity of the model mainly depends on the dimensionality of the subspace d and smaller is the dimensionality, the more the model will be parsimonious. The number of parameters to estimate in the $\text{DLM}_{[\Sigma_k \beta_k]}$ model depends on the number of classes K since the dimensionality of the subspace d is assumed to be strictly less than K. Indeed, since the discriminative subspace is spanned by the column vectors of

the matrix U, there are Kd parameters for the means and $K-1$ proportions to estimate. The estimation of the K covariance matrices in the latent space requires $Kd(d+1)/2$ parameters to estimate and only K parameters for the orthogonal subspace. Finally, only d columns have to be estimated to obtain the projection matrix U and the columns of this matrix are constrained to be orthogonal. Hence, the matrix U needs the estimation of $d[p-(d+1)/2]$ loadings. Therefore, the complexity γ of the DLM$_{[\Sigma_k\beta_k]}$ is:

$$\gamma = d(\frac{3K-1}{2} + p) + d^2\frac{K-1}{2} + 2K - 1.$$

Contrary to the traditional Gaussian models in which the complexity increases with p^2, the complexity of the DLM$_{[\Sigma_k\beta_k]}$ grows linearly with p. In particular, if we consider the case with $p = 100$, $K = 4$ and $d = 3$, then the complexity of the DLM$_{[\Sigma_k\beta_k]}$ is $\gamma = 337$ which is drastically less than the number of parameters to estimate in the case of the Full-GMM ($\gamma = 20603$) and remains very competitive with the most parsimonious model, Sphe-GMM, which requires the estimation of 407 parameters in the same case. On top of this parsimony, the DLM model offers also a flexible modelling structure for high-dimensional data as it explains in the next section.

3.2 The submodels of the DLM$_{[\Sigma_k\beta_k]}$ model

By applying constraints on parameters of the matrix Δ_k in the DLM$_{[\Sigma_k\beta_k]}$ model, several submodels can be generated. They can be separated into two categories according to the shape of Δ_k: the models with variable shapes of covariance matrices and those having a common covariance matrix accross the groups.

3.2.1 Characterization of the submodels

In the first category, the covariance matrices $\Sigma_1, \ldots, \Sigma_K$ in the latent space can be assumed to be common across groups and this submodel will be referred to by DLM$_{[\Sigma\beta_k]}$. Similarly, in each group, Σ_k can be assumed to be diagonal, *i.e.* $\Sigma_k = \text{diag}(\alpha_{k1}, \ldots, \alpha_{kd})$. This submodel will be referred to by DLM$_{[\alpha_{kj}\beta_k]}$. In the same manner, the $p-d$ last values of Δ_k can be assumed to be common for the k classes, *i.e.* $\beta_k = \beta$, $\forall k = 1, ..., K$, meaning that the variance outside the discriminant subspace is common to all groups. This assumption can be viewed as modeling the non discriminative information with a unique parameter which seems natural for data obtained in a common acquisition process. Following the notation system introduced above, this submodel will be referred to by DLM$_{[\alpha_{kj}\beta]}$. The variance within the latent subspace $\mathbb{E}$ can also be assumed to be isotropic for each group and the associated submodel is DLM$_{[\alpha_k\beta_k]}$. In this case, the variance of the data is assumed to be isotropic both within $\mathbb{E}$ and outside $\mathbb{E}$. Similarly, it is possible to compel the previous model to have the parameters β_k common between classes and this gives rise to the model DLM$_{[\alpha_k\beta]}$. Finally, the variance within the

subspace $\mathbb{E}$ can be assumed to be independent from the mixture component which corresponds to the DLM$_{[\alpha_j\beta_k]}$ model and can also be constrained to be spherical suggesting the model DLM$_{[\alpha\beta_k]}$.

In the second category, there remain 3 models which are the models DLM$_{[\Sigma\beta]}$, DLM$_{[\alpha_j\beta]}$, and DLM$_{[\alpha\beta]}$. These models all assume that the variance outside the latent space is isotropic meaning that $\beta_k = \beta$, $\forall k \in \{1, \ldots, K\}$. Moreover, either the covariance matrices Σ_k in the latent space are assumed to be common across groups and then we obtain the model DLM$_{[\Sigma\beta]}$ or, they are supposed to be common and diagonal, $\Sigma_k = \text{diag}(\alpha_1, \ldots, \alpha_d)$ for all k, which corresponds to the model DLM$_{[\alpha_j\beta]}$. Finally, the most parsimonious DLM model assumes that the covariance matrix is isotropic in both subspaces which suggested in particular that $\Sigma_k = \alpha\mathbf{I}_d$, $\forall k \in \{1, \ldots, K\}$ in the latent subspace.

We therefore enumerate 12 different DLM models and an overview of them is proposed in Table 3.1.

3.2.2 Complexity of the submodels

Table 3.1 also gives the maximum number of free parameters to estimate (case of $d = K - 1$) according to K and p for the 12 DLM models and for some classical models. We recall that the Full-GMM model refers to the classical Gaussian mixture model with full covariance matrices, the Com-GMM model refers to the Gaussian mixture model for which the covariance matrices are assumed to be equal to a common covariance matrix ($S_k = S$, $\forall k$), Diag-GMM refers to the Gaussian mixture model for which $S_k = \text{diag}(s_{k1}^2, ..., s_{kp}^2)$ with $s_{kj}^2 \in \mathbb{R}$ and Sphe-GMM refers to the Gaussian mixture model for which $S_k = s_k^2 I_p$ with $s_k^2 \in \mathbb{R}$. Finally, Mixt-PPCA denotes the subspace clustering model proposed by Tipping and Bishop in [165]. In addition, Table 3.1 gives the number of free parameters to estimate for specific values of K and p in the right column. The number of free parameters to estimate given in the central column can be decomposed in the number of parameters to estimate for the proportions ($K - 1$), for the means (Kp or Kd) and for the covariance matrices (last terms). Among the classical models, the Full-GMM model is a highly parametrized model and requires the estimation of 20603 parameters when $K = 4$ and $p = 100$. Conversely, the Diag-GMM and Sphe-GMM model are parsimonious models since they respectively require the estimation of only 803 and 407 parameters when $K = 4$ and $p = 100$. The Com-GMM and Mixt-PPCA models appear to both have an intermediate complexity. However, the Mixt-PPCA model is a less constrained model compared to the Diag-GMM model and should be preferred for clustering high-dimensional data.

In this table, it appears that the DLM models are very parsimonious models compared to the classical GMM (Full-GMM, Com-GMM, Diag-GMM, Sphe-GMM) or to some subspace clustering methods (Mixt-PPCA). Only the model (MCUFSA) developed by Yoshida *et al.* [188, 189] (see Chapter 2) has a number of free parameters which is comparable to those of the DLM family. However, whereas in that model the authors assume some very strong

Model	Nb. of parameters	$K = 4, d = 3$ $p = 100$
$\text{DLM}_{[\Sigma_k\beta_k]}$	$(K-1) + Kd + d[p-(d+1)/2] + Kd(d+1)/2 + K$	337
$\text{DLM}_{[\Sigma_k\beta]}$	$(K-1) + Kd + d[p-(d+1)/2] + Kd(d+1)/2 + 1$	334
$\text{DLM}_{[\Sigma\beta_k]}$	$(K-1) + Kd + d[p-(d+1)/2] + d(d+1)/2 + K$	319
$\text{DLM}_{[\alpha_{kj}\beta_k]}$	$(K-1) + Kd + d[p-(d+1)/2] + K(d+1)$	325
$\text{DLM}_{[\alpha_{kj}\beta]}$	$(K-1) + Kd + d[p-(d+1)/2] + Kd + 1$	322
$\text{DLM}_{[\alpha_k\beta_k]}$	$(K-1) + Kd + d[p-(d+1)/2] + 2K$	317
$\text{DLM}_{[\alpha_k\beta]}$	$(K-1) + Kd + d[p-(d+1)/2] + K + 1$	314
$\text{DLM}_{[\alpha_j\beta_k]}$	$(K-1) + Kd + d[p-(d+1)/2] + d + K$	316
$\text{DLM}_{[\alpha\beta_k]}$	$(K-1) + Kd + d[p-(d+1)/2] + 1 + K$	314
$\text{DLM}_{[\Sigma\beta]}$	$(K-1) + Kd + d[p-(d+1)/2] + d(d+1)/2 + 1$	316
$\text{DLM}_{[\alpha_j\beta]}$	$(K-1) + Kd + d[p-(d+1)/2] + d + 1$	313
$\text{DLM}_{[\alpha\beta]}$	$(K-1) + Kd + d[p-(d+1)/2] + 2$	311
Full-GMM	$(K-1) + Kp + Kp(p+1)/2$	20603
Com-GMM	$(K-1) + Kp + p(p+1)/2$	5453
Mixt-PPCA	$(K-1) + Kp + K(d(p-(d+1)/2) + d + 1) + 1$	1198
Diag-GMM	$(K-1) + Kp + Kp$	803
Sphe-GMM	$(K-1) + Kp + K$	407
MCUFSA	$(K-1) + Kd + 1 + d[p-(d+1)/2] + Kd$	322

Table 3.1: Number of free parameters to estimate when $d = K - 1$ for the DLM models and some classical models (see text for details). The numerical examples have been done with parameters $p = 100$, $K = 4$ and $d = 3$ for all models.

conditions on the covariance matrices of the component factors and of the specific factors which can appear too restrictive in certain situations, the family of DLM models proposes 12 different models more or less constraints while remaining very parsimonious. The DLM models turn out to have a low complexity whereas their modeling capacity is comparable to the one of the Mixt-PPCA model. Moreover, according to the fact that the dimension d is linked to the number of clusters K (this remark is developed in the next Chapter) then the complexity of each DLM model depends only on K and p contrary to Mixt-PPCA or MCUFSA which depend also on an hyper-parameter d independent of the number of clusters.

To conclude, the DLM models drastically reduce the complexity of models compared to the other approaches while allowing to modelize structures of the covariance matrices, more or less unconstrained, in the latent space.

3.2.3 Complete log-likelihood of the DLM submodels

The complete log-likelihoods of the 11 DLM submodels are defined in the following propositions:

Proposition 3.2.1. *The complete log-likelihood $\ell(y_1,\dots,y_n,\theta)$ of the model* DLM$_{[\Sigma_k\beta]}$ *has the following expression:*

$$\begin{aligned}\ell(\theta) = &-\frac{1}{2}\Big(\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + \operatorname{trace}(\Sigma_k^{-1}U^tC_kU) + \log(|\Sigma_k|) + \gamma\Big] \\ &+ n(p-d)\ \log(\beta) + \frac{1}{\beta}\Big[n\ \operatorname{trace}(C) - n\sum_{j=1}^{d} u_j^tCu_j\Big]\Big).\end{aligned} \tag{3.2.1}$$

where $C_k = \frac{1}{n_k}\sum_{i=1}^{n} z_{ik}(y_i-m_k)(y_i-m_k)^t$ is the empirical covariance matrix of the kth group, $C = \frac{1}{n}\sum_{k=1}^{K} n_kC_k$ is the empirical within covariance matrix, u_j is the jth column vector of U, $n_k = \sum_{i=1}^{n} z_{ik}$ and $\gamma = p\log(2\pi)$ is a constant term.

Proof. According to the expression obtained in equation (3.1.6), the complete log-likelihood of the DLM$_{[\Sigma_k\beta]}$ for which the non discriminative information term is assumed to be common to all classes, such that $\beta_k = \beta\ \forall k$, has the following form:

$$\begin{aligned}\ell(\theta) = &-\frac{1}{2}\Big(\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + \operatorname{trace}(\Sigma_k^{-1}U^tC_kU) + \log(|\Sigma_k|)\Big] \\ &+ \sum_{k=1}^{K} n_k(p-d)\ \log(\beta) + \sum_{k=1}^{K}\frac{n_k}{\beta}\Big[\operatorname{trace}(C_k) - \sum_{j=1}^{d} u_j^tC_ku_j\Big]\Big),\end{aligned}$$

then by noting that the terms $(p-d)\ \log(\beta)$ is independent of the class k:

$$\sum_{k=1}^{K}\frac{n_k}{\beta}\text{trace}(C_k)=\frac{1}{\beta}\text{trace}(C),$$

where $C=\frac{1}{n}\sum_{k=1}^{K}n_kC_k$ stands for the empirical within covariance matrix of the whole dataset, and:

$$\begin{aligned}\sum_{k=1}^{K}\frac{n_k}{\beta}\sum_{j=1}^{d}u_j^tC_ku_j &= \frac{1}{\beta}\sum_{j=1}^{d}u_j^t\left(\sum_{k=1}^{K}n_kC_k\right)u_j\\ &= \frac{n}{\beta}\sum_{j=1}^{d}u_j^tCu_j,\end{aligned}$$

then, the complete log-likelihood can be written as:

$$\begin{aligned}\ell(\theta)=&-\frac{1}{2}\Big(\sum_{k=1}^{K}n_k\Big[-2\log(\pi_k)+\text{trace}(\Sigma_k^{-1}U^tC_kU)+\log(|\Sigma_k|)+\gamma\Big]\\ &+n(p-d)\ \log(\beta)+\frac{1}{\beta}\Big[n\ \text{trace}(C)-n\sum_{j=1}^{d}u_j^tCu_j\Big]\Big),\end{aligned}$$

□

Proposition 3.2.2. *The complete log-likelihood $\ell(y_1,\ldots,y_n,\theta)$ of the model* DLM$_{[\Sigma\beta_k]}$ *has the following expression:*

$$\begin{aligned}\ell(\theta)=&-\frac{1}{2}\Big(\sum_{k=1}^{K}n_k\Big[-2\log(\pi_k)\Big]+n\ \log(|\Sigma|)+n\ \text{trace}(\Sigma^{-1}U^tCU)\Big]\\ &+\sum_{k=1}^{K}n_k\Big[(p-d)\ \log(\beta_k)+\frac{1}{\beta_k}(\text{trace}(C_k)-\sum_{j=1}^{d}u_j^tC_ku_j)+\gamma\Big]\Big).\end{aligned}\tag{3.2.2}$$

where C_k and C have been already defined, $n_k=\sum_{i=1}^{n}z_{ik}$ and $\gamma=p\log(2\pi)$ is a constant term.

The proof is obvious, by considering $\Sigma_k=\Sigma$, $\forall k=1,\ldots,K$ in the complete log-likelihood defined in equation (3.1.6). In the same manner by considering Proposition (3.1.6) and Proposition (3.2.1), the complete log-likelihood of the DLM$_{[\Sigma\beta]}$ is obtained.

Proposition 3.2.3. *The complete log-likelihood $\ell(y_1,\ldots,y_n,\theta)$ of the model* DLM$_{[\alpha_{kj}\beta_k]}$ *has the following expression:*

$$\begin{aligned}\ell(\theta) &= -\frac{1}{2}\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + \sum_{j=1}^{d}\left(\log(\alpha_{kj}) + \frac{u_j^t C_k u_j}{\alpha_{jk}}\right)\\ &\quad +(p-d)\ \log(\beta_k) + \frac{1}{\beta_k}(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^t C_k u_j) + \gamma\Big],\end{aligned}$$

where C_k and C have been already defined, $n_k = \sum_{i=1}^{n} z_{ik}$ and $\gamma = p\log(2\pi)$ is a constant term. The complete log-likelihood of model DLM$_{[\alpha_{kj}\beta]}$ *is derived from this proposition and Proposition 3.2.1.*

Proof. By considering the complete log-likelihood obtained in Proposition 3.2.1 and by considering that $\Sigma_k = \text{diag}(\alpha_{k1}, \ldots, \alpha_{kd})$, then the quantity $\log(|\Sigma_k|)$ can be rewritten in this way:

$$\begin{aligned}\log(|\Sigma_k|) &= \log(\prod_{j=1}^{d}\alpha_{kj})\\ &= \sum_{j=1}^{d}\log(\alpha_{kj}).\end{aligned}$$

Moreover, the quantity $\text{trace}(\Sigma_k^{-1}U^t C_k U)$ becomes:

$$\begin{aligned}\text{trace}(\Sigma_k^{-1}U^t C_k U) &= \text{trace}(\text{diag}(\frac{1}{\alpha_{k1}}, \ldots, \frac{1}{\alpha_{kd}})U^t C_k U)\\ &= \sum_{\text{j}=1}^{\text{d}}\text{trace}(\frac{1}{\alpha_{kj}}u_j^t C_k u_j).\end{aligned}$$

Since $u_j^t C_k u_j$ is a scalar, the trace is equal to this scalar and this enables us to conclude. Besides, by replacing β_k with $\beta\ \forall k \in \{1\ldots, K\}$ in the expression obtained in Proposition 3.2.3, then the complete log-likelihood of the submodel DLM$_{[\alpha_{kj}\beta]}$ is obtained.

□

Proposition 3.2.4. *For the* DLM$_{[\alpha_k\beta_k]}$ *model, the complete log-likelihood has the following form:*

$$\begin{aligned}\ell(\theta) = &-\frac{1}{2}\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + d\ \log(\alpha_k) + \frac{1}{\alpha_k}\sum_{j=1}^{d} u_j^t C_k u_j\\ &+ (p-d)\ \log(\beta_k) + \frac{1}{\beta_k}(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^t C_k u_j) + \gamma\Big].\end{aligned}$$

where C_k and C have been already defined, $n_k = \sum_{i=1}^{n} z_{ik}$ where $z_{ik} = \mathbf{1}_{\{z_{ik}\in\mathcal{C}_k\}}$ and $\gamma =$

$p\log(2\pi)$ *is a constant term. The complete log-likelihood of the* DLM$_{[\alpha_k\beta]}$ *model derives from this proposition with* $\beta_k = \beta\ \forall k \in \{1,\dots,K\}$.

Proof. By constraining the covariance matrix of the latent space to be isotropic, meaning that $\Sigma_k = \alpha_k \mathbf{I}_d$, the expression of the log-likelihood obtained in Proposition 3.2.3 becomes:

$$\begin{aligned}\ell(\theta) = &-\frac{1}{2}\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + \sum_{j=1}^{d}\left(\log(\alpha_k) + \frac{u_j^t C_k u_j}{\alpha_k}\right)\\ &+ (p-d)\ \log(\beta_k) + \frac{1}{\beta_k}(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^t C_k u_j) + \gamma\Big],\end{aligned}$$

which directly enables us to conclude.

□

Proposition 3.2.5. *The complete log-likelihood of the* DLM$_{[\alpha_j\beta_k]}$ *model has the following form:*

$$\begin{aligned}\ell(\theta) = &-\frac{1}{2}\Big(\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k)\Big] + n\sum_{j=1}^{d}\log(\alpha_j) + n\sum_{j=1}^{d}\frac{u_j^t C u_j}{\alpha_j}\\ &+ \sum_{k=1}^{K} n_k\Big[(p-d)\ \log(\beta_k) + \frac{1}{\beta_k}(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^t C_k u_j) + \gamma\Big]\Big).\end{aligned}$$

where C_k *and* C *have been already defined,* $n_k = \sum_{i=1}^{n} z_{ik}$ *where* $z_{ik} = \mathbf{1}_{\{z_{ik}\in\mathcal{C}_k\}}$ *and* $\gamma = p\log(2\pi)$ *is a constant term. The complete log-likelihood of the models* DLM$_{[\alpha_j\beta]}$, DLM$_{[\alpha\beta_k]}$ *and* DLM$_{[\alpha\beta]}$ *derive directly from this proposition with* $\beta_k = \beta\ \forall k \in \{1,\dots,K\}$ *or/and* $\alpha_j = \alpha$ $\forall j \in \{1,\dots,d\}$.

Proof. By replacing the terms α_{kj} by α_j in Proposition 3.2.3, then the complete log-likelihood can be reexpressed as:

$$\begin{aligned}\ell(\theta) = &-\frac{1}{2}\sum_{k=1}^{K} n_k\Big(-2\log(\pi_k) + \sum_{j=1}^{d}\left(\log(\alpha_j) + \frac{u_j^t C_k u_j}{\alpha_j}\right)\\ &+ (p-d)\log(\beta_k) + \frac{1}{\beta_k}\sum_{j=1}^{p-d} v_j^t C_k v_j + \gamma\Big),\\ = &-\frac{1}{2}\Big(\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k)\Big] + n\sum_{j=1}^{d}\log(\alpha_j) + n\sum_{j=1}^{d}\frac{u_j^t C u_j}{\alpha_j}\\ &+ \sum_{k=1}^{K} n_k\Big[(p-d)\ \log(\beta_k) + \frac{1}{\beta_k}(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^t C_k u_j) + \gamma\Big]\Big).\end{aligned}$$

By replacing β_k with β, $\forall k \in \{1, \ldots, K\}$ in this expression, the complete log-likelihood of the submodel DLM$_{[\alpha_j \beta]}$ is obtained. Moreover, by constraining the terms α_j to be common on all dimensions $j \in \{1, \ldots, d\}$, then the log-likelihood becomes:

$$\begin{aligned}
\ell(\theta) = & -\frac{1}{2}\sum_{k=1}^{K} n_k \Big(-2\log(\pi_k) + d\ \log(\alpha) + \frac{1}{\alpha}\sum_{j=1}^{d} u_j^t C_k u_j + \\
& + (p-d)\log(\beta_k) + \frac{1}{\beta_k}(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^t C_k u_j) + \gamma \Big), \\
\ell(\theta) = & -\frac{1}{2}\Big(\sum_{k=1}^{K} n_k[-2\log(\pi_k)] + n\ d\ \log(\alpha) + \frac{n}{\alpha}\sum_{j=1}^{d} u_j^t C u_j \\
& + \sum_{k=1}^{K} n_k \Big[(p-d)\log(\beta_k) + \frac{1}{\beta_k}(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^t C_k u_j) + \gamma \Big] \Big),
\end{aligned}$$

which is the expression of the complete log-likelihood of the DLM$_{[\alpha\beta_k]}$ and of the DLM$_{[\alpha\beta_k]}$ with $\beta_k = \beta\ \forall k \in \{1, \ldots, K\}$. □

3.2.4 Classification functions of the DLM submodels

DLM$_{[\Sigma_k \beta]}$ model: According to the classification function obtained for the general model in equation (3.1.2), the classification function Γ_k of the submodel DLM$_{[\Sigma_k\beta]}$ is:

$$\begin{aligned}
\Gamma_k(y) \quad = \quad & ||P(y) - \mu_k||^2_{\mathcal{D}_k} + \frac{1}{\beta}||\,(y - P(y)) - (m_k - \mu_k)\,||^2 \\
& + \log\left(|\Sigma_k|\right) + (p-d)\log(\beta) - 2\log(\pi_k) + \gamma, \qquad (3.2.3)
\end{aligned}$$

where $||.||^2_{\mathcal{D}_k}$ is a norm on the latent space $\mathbb{E}$ defined by $||y||^2_{\mathcal{D}_k} = y^t \mathcal{D}_k y$, $\mathcal{D}_k = \tilde{W}\Delta_k^{-1}\tilde{W}^t$, $\tilde{W}$ is a $p \times p$ matrix containing the d vectors of U completed by zeros such as $\tilde{W} = [U, 0_{p-d}]$, P is the projection operator on the latent space $\mathbb{E}$, *i.e.* $P(y) = UU^t y$, and $\gamma = p\log(2\pi)$ is a constant term.

DLM$_{[\Sigma\beta_k]}$ model: The classification function Γ_k of the submodel DLM$_{[\Sigma\beta]}$ is:

$$\begin{aligned}
\Gamma_k(y) \quad = \quad & ||P(y) - \mu_k||^2_{\mathcal{D}_k} + \frac{1}{\beta_k}||\,(y - P(y)) - (m_k - \mu_k)\,||^2 \\
& + \log\left(|\Sigma|\right) + (p-d)\log(\beta_k) - 2\log(\pi_k) + \gamma, \qquad (3.2.4)
\end{aligned}$$

The classification function of the DLM$_{[\Sigma\beta]}$ model can be obtained by reparametrizing β_k and $\mathcal{D}_k$ in $\beta_k = \beta$ and also $\mathcal{D}_k = \mathcal{D}$ such that $\mathcal{D} = \tilde{W}\Delta^{-1}\tilde{W}^t$.

DLM$_{[\alpha_{kj}\beta_k]}$ model: Since $\Sigma_k = \text{diag}(\alpha_{k1}, \ldots, \alpha_{kd})$, then the logarithm of the determinant of Σ_k becomes:

$$\log\left(|\Sigma_k|\right) = \log \prod_{j=1}^{d} \alpha_{kj} = \sum_{j=1}^{d} \log\left(\alpha_{kj}\right). \tag{3.2.5}$$

Consequently, by considering this formulation with the classification function obtained in equation (3.1.7) then, the classification function Γ_k of the submodel DLM$_{[\alpha_{kj}\beta]}$ can be reformulated as:

$$\begin{aligned}\Gamma_k(y) = & \quad ||P(y) - \mu_k||^2_{\mathcal{D}_k} + \frac{1}{\beta_k}||\left(y - P(y)\right) - \left(m_k - \mu_k\right)||^2 \\ & + \sum_{j=1}^{d} \log(\alpha_{kj}) + (p-d)\log(\beta_k) - 2\log(\pi_k) + \gamma. \end{aligned} \tag{3.2.6}$$

By considering the same remark as previously, the classification function for the DLM$_{[\alpha_{kj}\beta]}$ is directly obtained by reparametrizing the scalar β_k in β and the matrix $\mathcal{D}_k$ in $\mathcal{D}$ such that $\mathcal{D} = \tilde{W}\Delta^{-1}\tilde{W}^t$.

DLM$_{[\alpha_k\beta]}$ model: According to equation (3.2.5) and by considering that $\Sigma_k = \alpha_k \mathbf{I}_d$ then, the classification function of the DLM$_{[\alpha_k\beta]}$ model is:

$$\begin{aligned}\Gamma_k(y) = & \quad \frac{1}{\alpha_k}||P(y) - \mu_k||^2 + \frac{1}{\beta}||\left(y - P(y)\right) - \left(m_k - \mu_k\right)||^2 \\ & + d\log(\alpha_k) + (p-d)\log(\beta) - 2\log(\pi_k) + \gamma. \end{aligned} \tag{3.2.7}$$

In this model, it can be observed that the distance $||\mu_k - P(y)||^2_{\mathcal{D}_k}$ between the projection of the observation y_i and the mean of the cluster k in the latent space is here a Euclidean distance weighted by the parameter α_k which stands for the variance term of the cluster k in the latent space $\mathbb{E}$.

DLM$_{[\alpha\beta]}$ model: The classification function of the DLM$_{[\alpha\beta]}$ model is:

$$\Gamma_k(y) = \frac{1}{\alpha}||P(y) - \mu_k||^2 + \frac{1}{\beta}||\left(y - P(y)\right) - \left(m_k - \mu_k\right)||^2 - 2\log(\pi_k) + \gamma_1, \tag{3.2.8}$$

where $\gamma_1 = d\log(\alpha) + (p-d)\log(\beta) + p\log(2\pi)$ is a constant term. In the same manner as previously, the distance $||P(y - m_k)||^2_{\mathcal{D}_k} = ||P(y - m_k)||^2$ is also a Euclidean distance.

3.3 Comparison with existing methods

At this point, some links can be established with models existing in the clustering literature. The closest models were proposed in [8], [22] and [129] and are linked to the mixture of factor analyzer (MFA) model [128, 153]. According to the assumptions of the DLM model, the comparison has to be done on a subset of these families. Particularly, we have to consider the models which assume that the subspace of each group is common, the variance of the non discriminative information is isotropic and also that the dimension of the subspace is $d \leq K - 1$.

Firstly, the DLM and Hd-GMM models are very similar since they share a common assumption on the decomposition of the covariance matrix of the observation space. In particular, the 8 DLM models which suppose a diagonal matrix in the latent subspace ($\Sigma_k = \text{diag}(\alpha_{k1}, \ldots, \alpha_{kd})$, $\Sigma_k = \alpha_j \mathbf{I}_d$ and also $\Sigma_k = \alpha \mathbf{I}_d$, $\forall k = 1, \ldots, K$) are comparable to those belonging to the family of $[a_{kj} b_k Q d]$ models. Indeed, for these models, they both assume that the covariance matrice of each group S_k in the observation space can be decomposed such that $\Delta_k = Q S_k Q^t$, where Δ_k is a diagonal matrix and Q is an orthogonal matrix. The Hd-GMM model can be rewritten through a local latent mixture model which seeks to relate a p-dimensional observation vector y to a corresponding d-dimensional vector of latent variables X where $d < p$.

In the same manner, it is also possible to compare the PGMM and the extended PGMM families with the DLM models, in the structure of their covariance matrix in particular. For example, in the more general case, the covariance structure in the observation space of the DLM model is $S_k = U \Sigma_k U^t + \Psi_k$ and those of the PGMM family is $S_k = \Lambda_k \Lambda_k^t + \Psi$ with Ψ a diagonal matrix. However, the main difference with the EPGMM approach remains in the fact that the DLM model defines a common subspace for all clusters which is not the case in EPGMM. Hence, the approaches like MCFA, MCUFSA and HMFA which defines a common matrix for the component factor loadings propose closer models of the DLM model than those obtained by EPGMM. Moreover, the models complexity of these 3 approaches are in a same order that the ones of the DLM models. In particular, the MCUFSA approach has a number of free parameters which is equivalent to the $\text{DLM}_{[\alpha_{kj}\beta]}$. These both models assume diagonal covariance matrices in the latent space and an isotropic variance for the non discriminative information.

However, despite the fact that all these models share some assumptions on the covariance structure in the observation space or on the latent and the non discriminative subspaces, the DLM model remains very different. Indeed, the main difference between those models remains in both the definition and the estimation of the latent subspace. Indeed, in the case of Hd-GMM, the projection matrix is estimated by maximum log-likelihood through an EM procedure and its columns are the eigenvectors corresponding to the largest eigenvalues of a weighted within covariance matrix. In the same manner, the factor loadings of the MFA models are obtained by maximum log-likelihood but through an extension of the EM algorithm named AECM since the maximization step is divided in substeps. In both cases, the loadings

estimated are chosen such as the variance or the covariance structure of the projected data is maximum. Differently, the DLM models chooses the latent subspace orientation such as it best discriminates the groups. In particular, our approach aims to estimate a subspace which best discriminates the clusters meaning that in such a subspace, the centroids of each cluster are well separated between them but the covariance of each cluster remains compact. This specific feature of the DLM models should therefore improve in most cases both the clustering and the visualization of the results. In particular, the DLM models should be able to better model situations where the axes carrying the greatest variance are not parallel to the discriminative axes than the other approaches (Figure 10.1 of [62] illustrates such a situation). Consequently, the next Chapter introduces an algorithm which enables to both cluster the data and estimate a discriminative latent subspace.

Chapter 4

Parameter estimation: the Fisher-EM algorithm

This chapter proposes an estimation procedure for the parameters of the DLM model. Since this work focuses on clustering of unlabeled data by Gaussian mixture model, the estimation procedure that we propose is based on an EM-type algorithm for estimating the parameters of DLM models. Due to the nature and to the goal of the models described above, the algorithm we propose, alternates three-steps:

- an E-step in which posterior probabilities that observations belong to the K groups are computed,
- a F-step which estimates the orientation matrix U of the discriminative latent space conditionally to the posterior probabilities,
- an M-step in which parameters of the mixture model are estimated in the latent subspace by maximizing the conditional expectation of the complete likelihood.

Due to the additional step introduced in the traditional EM algorithm, the F-step, which is based on the work of Sir R.A. Fisher, we have named the proposed algorithm, the Fisher-EM algorithm.

In this Chapter, the three-steps of the Fisher-EM algorithm will be described. In particular, we detail three different ways to compute the projection matrix U of the discriminative subspace in the F-step. Moreover, as the projection matrix U is not obtained in maximizing the conditional expectation of the log-likelihood, the convergence of our algorithm is not guaranteed. Consequently, we will show that, for the isotropic case of the DLM model, the convergence of the Fisher-EM algorithm is satisfied. This aspect will be discussed in Section 4.2. Some computational aspects of the Fisher-EM algorithm concerning initialization, stopping criterion and model selection problems will be also discussed. Finally, this chapter will end with some practical aspects, such as the choice of the dimension of the latent subspace d or the case of high dimension and low sample size dataset.

4.1 The Fisher-EM algorithm

Since the DLM model is inscribed in a Gaussian mixture model context, then a very common way to estimate the parameters of the mixture model is the standard EM algorithm. However, an additional step named the F-step, is introduced between the E and the M-steps to compute the projection matrix whose columns span the discriminative latent subspace.

4.1.1 The E-step

This step aims to compute, at iteration q, the expectation of the complete log-likelihood conditionally to the current value of the parameter $\theta^{(q-1)}$, which, in practice, reduces to the computation of $t_{ik}^{(q)} = E[z_{ik}|y_i, \theta^{(q-1)}]$ where $z_{ik} = 1$ if y_i comes from the kth component and $z_{ik} = 0$ otherwise. Let us also recall that $t_{ik}^{(q)}$ is, as well, the posterior probability that the observation y_i belongs to the kth component of the mixture. The Bayes formula enables us to write the posterior probability $t_{ik}^{(q)}$ as:

$$t_{ik}^{(q)} = \frac{\pi_k \phi(y_i, \theta_k)}{\sum_{\ell=1}^{K} \pi_\ell \phi(y_i, \theta_\ell)},$$

where π_k is the mixture proportion of the cluster k. Then, by considering the classification function $\Gamma_k(y_i) = -2\log(y_i, \theta_k)$ defined in Chapter 3, then the explicit form of $t_{ik}^{(q)}$, for $i = 1, ..., n,\ k = 1, ..., K$, in the case of the model $\text{DLM}_{[\Sigma_k \beta_k]}$ is provided by the following proposition:

Proposition 4.1.1. *With the assumptions of the model* $\text{DLM}_{[\Sigma_k \beta_k]}$*, the posterior probabilities* $t_{ik}^{(q)}$*,* $i = 1, ..., n,\ k = 1, ..., K$*, can be expressed as :*

$$t_{ik}^{(q)} = \frac{1}{\sum_{l=1}^{K} \exp\left(\frac{1}{2}(\Gamma_k^{(q-1)}(y) - \Gamma_l^{(q-1)}(y))\right)},$$

with:

$$\begin{aligned} \Gamma_k^{(q-1)}(y_i) = \quad & ||P(y_i - m_k^{(q-1)})||_{\mathcal{D}_k}^2 + \frac{1}{\beta_k^{(q-1)}} ||(y_i - m_k^{(q-1)}) - P(y_i - m_k^{(q-1)})||^2 \\ & + \log\left(\left|\Sigma_k^{(q-1)}\right|\right) + (p-d)\log(\beta_k^{(q-1)}) - 2\log(\pi_k^{(q-1)}) + \gamma, \end{aligned} \tag{4.1.1}$$

where $||.||_{\mathcal{D}_k}^2$ *is a norm on the latent space* $\mathbb{E}$ *defined by* $||y||_{\mathcal{D}_k}^2 = y^t \mathcal{D}_k y$*,* $\mathcal{D}_k = \tilde{W} \Delta_k^{-1} \tilde{W}^t$*,* $\tilde{W}$ *is a* $p \times p$ *matrix containing the* d *vectors of* $U^{(q-1)}$ *completed by zeros such as* $\tilde{W} = [U^{(q-1)}, 0_{p-d}]$*,* P *is the projection operator on the latent space* $\mathbb{E}$*, i.e.* $P(y) = U^{(q-1)} U^{(q-1)t} y$*, and* $\gamma = p\log(2\pi)$ *is a constant term.*

The proof is direct by considering the classification function established in Proposition (3.1.2). It can be noticed that the posterior probability is mainly defined by the classi-

fication function Γ_k which implies that the geometric interpretations and the computational remarks given in the previous chapter remain valid.

4.1.2 The F-step

This step aims to determine, at iteration q, the discriminative latent subspace of dimension $d \leq K-1$ in which the K groups are best separated. Naturally, the estimation of this latent subspace has to be done conditionally to the current values of posterior probabilities $t_{ik}^{(q)}$ which indicates the current soft partition of the data. Estimating the discriminative latent subspace $\mathbb{E}^{(q)}$ reduces to the computation of a projection matrix $U^{(q)} \in \mathbb{R}^{p\times q}$ consisting of d discriminative axes.

A subspace is qualified to be discriminative in the sense described by Fisher [54]. In particular, the projection matrix $U^{(q)}$ is chosen such as it maximizes a criterion which is large when the between covariance matrix S_B is large and when the within covariance matrix S_W is small. Following the original idea of Fisher [54], the d axes which best discriminate the K groups are those which maximize the criterion $J(U) = \text{trace}((U^t S_W U)^{-1} U^t S_B U)$. However, as the traditional criterion $J(U)$ is defined in a supervised classification framework, it assumes that the data are complete. Unfortunately, the situation of interest here is that of unsupervised classification and the matrices S_B and S_W have therefore to be defined conditionally to the current soft partition. Furthermore, the DLM models assume that the discriminative latent subspace must have an orthonormal basis and, sadly, the traditional Fisher's approach provides non-orthogonal discriminative axes.

Let us introduce the optimization problem by defining, in first, the soft between-covariance matrix $S_B^{(q)}$.

Definition 4.1.1. *The soft between-covariance matrix $S_B^{(q)}$ is defined conditionally to the posterior probabilities $t_{ik}^{(q)}$, obtained in the E-step, as follows:*

$$S_B^{(q)} = \frac{1}{n}\sum_{k=1}^{K} n_k^{(q)}(m_k^{(q)} - \bar{y})(m_k^{(q)} - \bar{y})^t,$$

where $n_k^{(q)} = \sum_{i=1}^{n} t_{ik}^{(q)}$, $m_k^{(q)} = \frac{1}{n}\sum_{i=1}^{n} t_{ik}^{(q)} y_i$ is the soft mean of the kth group at iteration q and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the empirical mean of the whole dataset.

Since the relation $S = S_W^{(q)} + S_B^{(q)}$ holds in this context as well, it is preferable from a computational point of view to use the covariance matrix $S = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^t$ of the whole dataset in the maximization problem, instead of $S_W^{(q)}$, as S remains fixed over the iterations. Moreover in Chapter 2, we have seen that the equivalence between the criteria $\text{trace}\left((U^t S_W U)^{-1} U^t S_B U\right)$ and $\text{trace}\left((U^t S U)^{-1} U^t S_B U\right)$ holds. Therefore, the F-step of the

Fisher-EM aims to solve, at iteration q, the following optimization problem:

$$\begin{aligned} \hat{U}^{(q)} &= \max_{U} \quad \text{trace}\left((U^t S U)^{-1} U^t S_B^{(q)} U\right), \\ &\quad \text{w.r.t.} \quad U^t U = \mathbf{I}_d. \end{aligned} \tag{4.1.2}$$

The following paragraphs propose three different procedures which keep the key idea of Fisher while providing orthonormal discriminative axes conditionally to the current soft partition of the data.

4.1.2.1 Gram-Schmidt orthonormalization

This first procedure follows the concept of the orthonormal discriminant vector (ODV) introduced by [55] in the supervised case and then extended by [75, 80, 114, 186]. The ODV procedure sequentially selects the most discriminative features by maximizing the Fisher criterion subject to the orthogonality of features. According to the optimization problem characterized in equation (4.1.2) and following the ODV procedure, the d discriminative axes are iteratively constructed by, first, computing an orthogonal complementary subspace to the current set of discriminative axes and, then, maximizing the Fisher criterion in this orthogonal subspace by solving the associated generalized eigenvalue problem.

To initialize this iterative procedure, the first vector of U is therefore the eigenvector associated with the largest eigenvalue of the matrix $S^{-1}S_B^{(q)}$. Then, assuming that the $r-1$ first orthonormal discriminative axes $\{u_1, \ldots, u_{r-1}\}$, which span the space $\mathcal{B}_{r-1}$, have been computed, the rth discriminative axis has to lie in the subspace $\mathcal{B}_{r-1}^{\perp}$ orthogonal to the space $\mathcal{B}_{r-1}$. The Gram-Schmidt orthonormalization procedure allows to find a basis $V^r = \{v_r, v_{r+1}, ..., v_d\}$ for the orthogonal subspace $\mathcal{B}_{r-1}^{\perp}$ such that:

$$v_l = \alpha_l (I_{\ell-1} - \sum_{j=1}^{\ell-1} v_j v_j^t)\psi_l, \qquad \ell = r, \ldots, p$$

where $v_j = u_j$ for $j = 1, ..., r-1$, α_ℓ is a normalization constant such that $||u_\ell|| = 1$ and ψ_ℓ is a vector linearly independent of u_j $\forall j \in \{1, \ldots, \ell-1\}$. Then, the rth discriminative axis is given by:

$$u_r = \frac{P_{r-1} u_r^{max}}{||u_r^{max}||},$$

where P_{r-1} is the projector on $\mathcal{B}_{r-1}$, u_r^{max} is the eigenvector associated with the largest eigenvalue of the matrix $S_r^{-1} S_{Br}^{(q)}$ with:

$$\begin{aligned} S_r &= {V^r}^t S V^r, \\ S_{Br}^{(q)} &= {V^r}^t S_B^{(q)} V^r, \end{aligned}$$

i.e. S_r and $S_{Br}^{(q)}$ are respectively the covariance and soft between-covariance matrices of the

data projected into the orthogonal subspace $\mathcal{B}_{r-1}^{\perp}$. This iterative procedure stops when the d orthonormal discriminative axes u_j are computed.

This procedure, based on the ODV procedure, builds a set of column vectors which are orthogonal but they are not guaranteed to be optimal. Moreover, an other limitation of such an approach is the well-known numerical instability of the Gram-Schmidt process.

4.1.2.2 Fisher's criterion as a regression criterion

The second procedure, computing the projection matrix of the F-step, reformulates the eigen-decomposition problem as a regression-type problem. This approach is based on the work of Qiao *et al.* [147] in the supervised context and has been presented in paragraph 2.2.2.4.

However, in the Qiao's work, the matrices H_W and H_B are computed according to the class membership. This is not possible in our case as we deal with an unsupervised context. In particular, in their approach, the matrix H_W which is based on the within covariance matrix needs to be centered from the class means and this can not be done in our case. Moreover, an additional problem occurs in our optimization problem since the DLM models assume that the discriminative latent subspace has an orthonormal basis and this constraint is not taken into account in the Qiao's work.

Consequently, we propose, in first, a reformulation of the matrices H_W and H_B such as they are computed at each iteration and conditionally to the E-step.

Definition 4.1.2. *The soft matrices $H_W^{(q)} \in \mathbb{R}^{p\times n}$ and $H_B^{(q)} \in \mathbb{R}^{p\times K}$ are defined conditionally to the posterior probabilities $t_{ik}^{(q)}$ computed in the E-step at iteration q:*

$$\begin{aligned} H_W^{(q)} &= \frac{1}{\sqrt{n}}\left[Y - \sum_{k=1}^{K} t_{1k}^{(q)} m_k^{(q)}, \ldots, Y - \sum_{k=1}^{K} t_{nk}^{(q)} m_k^{(q)}\right] \in \mathbb{R}^{p\times n} \\ H_B^{(q)} &= \frac{1}{\sqrt{n}}\left[\sqrt{n_1^{(q)}}(m_1^{(q)} - \bar{y}), \ldots, \sqrt{n_K^{(q)}}(m_K^{(q)} - \bar{y})\right] \in \mathbb{R}^{p\times K}, \end{aligned} \tag{4.1.3}$$

where $t_{ik}^{(q)}$, for $i = 1, \ldots, n$, stands for the posterior probability computed in the E-step, $n_k^{(q)} = \sum_{i=1}^{n} t_{ik}^{(q)}$ and $m_k^{(q)} = \frac{1}{n}\sum_{i=1}^{n} t_{ik}^{(q)} y_i$ is the soft mean vector of the cluster k.

According to these definitions, the matrices satisfy:

$$H_W^{(q)} H_W^{(q)t} = S_W^{(q)} \quad \text{and} \quad H_B^{(q)} H_B^{(q)t} = S_B^{(q)}, \tag{4.1.4}$$

where $S_W^{(q)}$ and $S_B^{(q)}$ stand for respectively, the soft within and between covariance matrices computed at iteration q. Then, the optimization problem presented in paragraph 2.2.2.4 can be reformulated, at iteration q, in terms of soft within and between covariance matrices defined conditionally to the E-step. The following optimization problem is solved in the F-step of the

Fisher-EM, at iteration q:

$$\begin{aligned}(\hat{A}^{(q)}, \hat{B}^{(q)}) \quad &= \quad \arg\min_{A,B} \sum_{k=1}^{K} \left\| R_W^{(q)-t} H_{B,k}^{(q)} - AB^t H_{B,k}^{(q)} \right\|_F^2 + \rho \sum_{j=1}^{d} \beta_j^t S_W^{(q)} \beta_j \\ &\text{w.r.t.} \quad A^t A = \mathbf{I}_d, \end{aligned} \tag{4.1.5}$$

where $S_W^{(q)} = R_W^{(q)t} R_W^{(q)}$ stands for the soft within covariance matrix and $R_W^{(q)} \in \mathbb{R}^{p \times p}$ is an upper triangular matrix. Besides, $H_{B,k}^{(q)}$ is the kth column of the matrix $H_B^{(q)}$ defined from the soft between covariance matrix $S_B^{(q)}$ in equation (4.1.4) and ρ is an hyper parameter to calibrate. Finally, $\|.\|_F$ denotes the Frobenius norm. By letting $\hat{B}^{(q)} = [\hat{\beta}_1^{(q)}, \ldots, \hat{\beta}_d^{(q)}]$ and according to the Qiao's results, the column vectors of the matrix $\hat{B} \in \mathbb{R}^{p \times d}$ span the same linear space as those of the projection matrix U.

However, the orthogonality constraint on the column vectors of the matrix U spanning the Fisher space is not guaranteed. To that end, we use a well-known result formulated in [66] which concerns the best approximation of a matrix by an orthogonal one. In particular, it is stated that: *Obtaining the best approximation of a matrix $X \in \mathbb{R}^{d \times p}$ by an orthonormal matrix with the same dimensionality is equivalent to an orthogonal Procrustes problem:*

$$\min \left\{ \|X - Q\|_F \, : \, Q^t Q = \mathbf{I}_p \right\},$$

then $Q = uv^t$ is the solution of such a problem where u and v are respectively the left and right singular vectors of the svd of X.
In our case, this result becomes:

Proposition 4.1.2. *By considering $\hat{A}^{(q)}$ and $\hat{B}^{(q)}$ solutions of the problem* (4.1.5)*, the best approximation of the projection matrix $U^{(q)}$ by an orthonormal one is solution of the following problem:*

$$\begin{aligned}\hat{U}^{(q)} \quad &= \quad \arg\min_{\mathcal{U}} \left\| \hat{B}^{(q)} - \mathcal{U} \right\|_F \\ &w.r.t. \quad \mathcal{U}^t \mathcal{U} = \mathbf{I_d}, \end{aligned}$$

where $\|.\|_F$ refers to the Frobenius norm. By considering the svd of $\hat{B}^{(q)} = u^{(q)} \Lambda^{(q)} v^{(q)t}$, then $\hat{U}^{(q)} = u^{(q)} v^{(q)t}$.

Proof. At iteration q, in the F-step and conditionally to the E-step, the following optimization problem is considered:

$$\begin{aligned}\left(\hat{A}^{(q)}, \hat{B}^{(q)}\right) \quad &= \quad \arg\min_{A,B} \sum_{k=1}^{K} \left\| \left(R_W^{(q)t}\right)^{-1} H_{B,k}^{(q)t} - AB^t H_{B,k}^{(q)t} \right\| + \rho \sum_{j=1}^{d} \beta_j^t S_W^{(q)} \beta_j \\ &\text{w.r.t.} \quad A^t A = \mathbf{I}_d \end{aligned}$$

and is solved from the Qiao's theorem developed in paragraph 2.2.2.4 of Chapter 2. Therefore, the column vectors of $\hat{B}^{(q)}$ span the same space as the solution of the eigendecomposition of $S_W^{(q)-1}S_B^{(q)}$ and the estimation of $\hat{A}^{(q)}$ is obtained by equation (2.2.31). Moreover, as we search the best approximation of the matrix $\hat{B}^{(q)}$ to an orthogonal matrix, then the optimization problem is equivalent to the following one:

$$\begin{aligned} \hat{U}^{(q)} &= \arg\min_{\mathcal{U}} \left\| \hat{B}^{(q)} - \mathcal{U} \right\|_F \\ &\text{w.r.t.} \quad \mathcal{U}^t\mathcal{U} = \mathbf{I_d}, \end{aligned}$$

where $\|.\|_F$ refers to the Frobenius norm. This problem is a nearest orthogonal Procrustes problem which can be solved by a singular value decomposition [66, 87]. The singular value decomposition of $\hat{B}^{(q)} = u^{(q)}\Lambda^{(q)}v^{(q)t}$ allows to write $\hat{U}^{(q)} = u^{(q)}v^{(q)t}$. According to Qiao's theorem, since $\hat{B}^{(q)}$ spans the same subspace as those obtained by the standard Fisher's criterion and according to the nearest Procrustes problem, $\hat{U}^{(q)}$ is an orthogonal matrix which best approximates the projection matrix U whose column vectors span the discriminative latent subspace. □

4.1.2.3 A modified Fisher criterion

The main purpose of this last approach is to ease the computation of the orthogonal projection matrix U, in the F-step. In particular, we propose a modified Fisher's criterion which aims to efficiently approximate the discriminative latent subspace. Instead of considering the optimization problem defined in (4.1.2), we look here for a $p \times d$ projection matrix U with orthogonal columns such as the associated latent subspace has a discrimination power as close as possible than the one of the whole observation space, *i.e.* such that the matrix $UU^tS^{-1}S_B^{(q)}$ best approximates the matrix $S^{-1}S_B^{(q)}$. Therefore, we can formulate this aim through the following minimization problem:

$$\begin{aligned} \hat{U}^{(q)} &= \arg\min_{U} \left\| S^{-1}S_B^{(q)} - UU^tS^{-1}S_B^{(q)} \right\|_F^2 \\ &\text{w.r.t} \quad U^tU = \mathbf{I_d}, \end{aligned} \tag{4.1.6}$$

where $\|.\|_F$ denotes the Frobenius norm. The solution of this new optimization problem is given by the following proposition:

Proposition 4.1.3. *At iteration q, the best approximation of the matrix $S^{-1}S_B^{(q)}$ onto an orthogonal subspace through a $p \times d$ projection matrix ($d < K-1$) is the solution of the following optimization problem:*

$$\begin{aligned} \hat{U}^{(q)} &= \arg\max_{U} \quad \text{trace}\left(U^t(S^{-1}S_B^{(q)})(S^{-1}S_B^{(q)})^tU\right), \\ &\quad w.r.t. \quad U^tU = \mathbf{I}_d. \end{aligned} \tag{4.1.7}$$

and the columns of $\hat{U}^{(q)}$ are the d first left eigenvectors of the singular value decomposition of $S^{-1}S_B^{(q)}$.

Proof. This proposition results directly from a Frobenius norm property which is demonstrated in details in the mathematical background material developed by Ripley in [149][1].

Firstly, we can notice that:

$$\begin{aligned}
\left\|S^{-1}S_B^{(q)} - UU^tS^{-1}S_B^{(q)}\right\|_F^2 &= \text{trace}((S^{-1}S_B^{(q)} - UU^tS^{-1}S_B^{(q)})^t(S^{-1}S_B^{(q)} - UU^tS^{-1}S_B^{(q)})),\\
&= -2\text{trace}((S^{-1}S_B^{(q)}UU^tS^{-1}S_B^{(q)}) + \text{trace}((S^{-1}S_B^{(q)})^tS^{-1}S_B^{(q)})\\
&\quad +\text{trace}((S^{-1}S_B^{(q)})^tUU^tUU^tS^{-1}S_B^{(q)}),\\
&= \left\|S^{-1}S_B^{(q)}\right\|_F^2 - \text{trace}((S^{-1}S_B^{(q)}UU^tUU^tS^{-1}S_B^{(q)}),\\
&= \left\|S^{-1}S_B^{(q)}\right\|_F^2 - \left\|UU^tS^{-1}S_B^{(q)}\right\|_F^2.
\end{aligned}$$

It implies that minimizing the quantity $\left\|S^{-1}S_B^{(q)} - UU^tS^{-1}S_B^{(q)}\right\|_F^2$ is equivalent to maximize $\left\|UU^tS^{-1}S_B^{(q)}\right\|_F^2$. Furthermore, since $U^tU = \mathbf{I}_d$, the following equalities hold:

$$\begin{aligned}
\left\|UU^tS^{-1}S_B\right\|_F^2 &= \text{trace}\left((UU^tS^{-1}S_B)(UU^tS^{-1}S_B)^t\right)\\
&= \text{trace}\left(U^t(S^{-1}S_B)(S^{-1}S_B)^tU(U^tU)\right)\\
&= \text{trace}\left(U^t(S^{-1}S_B)(S^{-1}S_B)^tU\right).
\end{aligned}$$

where $\|.\|_F$ denotes the Frobenius norm.

Let us also consider the svd of the $n \times p$ matrix $S^{-1}S_B^{(q)} = u\Lambda v^t$ where u and v stands for respectively the left and right singular vectors of $S^{-1}S_B^{(q)}$ and Λ is a diagonal matrix containing its associated singular values. As the matrix $S_B^{(q)}$ has a rank d at most equal to $K-1 < p$, with K the number of clusters, then the matrix $S^{-1}S_B^{(q)}$ is also of rank $d = \text{rank}(S^{-1}S_B^{(q)})$ at most equal to $K-1 < p$. Consequently, only the d singular values of the matrix $S^{-1}S_B^{(q)}$ are non zeros, which enables us to write $S^{-1}S_B^{(q)} = u\Lambda_d v^t$, where $\Lambda_d = \text{diag}(\lambda_1, \ldots, \lambda_d, 0, \ldots, 0)$. Moreover, by letting $U^{(q)} = u_d^{(q)}$ the d first left eigenvectors of $S^{-1}S_B^{(q)}$, then:

$$\begin{aligned}
\text{trace}\left(U^{(q)t}(S^{-1}S_B^{(q)})(S^{-1}S_B^{(q)})^tU^{(q)}\right) &= \text{trace}\left(U^{(q)t}(u\Lambda_d v^t)(u\Lambda_d v^t)^tU^{(q)}\right),\\
&= \text{trace}\left(U^{(q)t}u\Lambda_d\Lambda_d^tu^tU^{(q)}\right),\\
&= \sum_{j=1}^{d}\left(\lambda_j^{(q)}\right)^2.
\end{aligned}$$

Consequently, the $p \times d$ orthogonal matrix $U^{(q)}$ such that $\left\|S^{-1}S_B^{(q)} - U^{(q)}U^{(q)t}S^{-1}S_B^{(q)}\right\|_F^2$ is minimal is the matrix made of the d first left eigenvectors of $S^{-1}S_B^{(q)}$. □

[1] http://www.stats.ox.ac.uk/~ripley/MultAnal_HT2007

This approach allows to obtain a discriminative subspace such as the projected matrix $U^{(q)t}S^{-1}S_B$ in terms of Frobenius norms is maximized. Moreover, in a computational point of view, the estimation of the projection matrix $U^{(q)}$ is much easier than those obtained by the Gram-Schmidt or the regression procedures. Indeed, we only need to decompose by a singular value decomposition the matrix $S^{-1}S_B^{(q)}$ at iteration q. The projection matrix whose its columns span the discriminative latent subspace is fitted by the d first left singular vectors of $S^{-1}S_B^{(q)}$.

4.1.3 The M-step

This third step estimates the model parameters by maximizing the conditional expectation of the complete likelihood, conditionally to the projection matrix $U^{(q)}$ estimated in the previous step and noted $\hat{U}^{(q)}$. The following proposition provides the expression of the conditional expectation of the complete log-likelihood in the case of the $\text{DLM}_{[\Sigma_k\beta_k]}$ model. According to Proposition 3.1.6 proved in the previous chapter, the conditional expectation of the complete log-likelihood noted $Q(y_1,\ldots,y_n,\theta)$, in the case of the model $\text{DLM}_{[\Sigma_k\beta_k]}$ has the following expression:

$$\begin{aligned} Q(y_1,\ldots,y_n,\theta) = -\frac{1}{2}\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + \text{trace}(\Sigma_k^{-1}U^tC_kU) + \log(|\Sigma_k|) \\ + (p-d)\log(\beta_k) + \frac{1}{\beta_k}\left(\text{trace}(C_k) - \sum_{j=1}^{d} u_j^tC_ku_j\right) + \gamma\Big]. \end{aligned} \tag{4.1.8}$$

where C_k is the empirical covariance matrix of the kth group, u_j is the jth column vector of U, $n_k = \sum_{i=1}^{n} t_{ik}$ and $\gamma = p\log(2\pi)$ is a constant term. At iteration q, the maximization of Q conduces to an estimation of the mixture proportions π_k and the means μ_k for the K components by their empirical counterparts:

$$\begin{aligned} \hat{\pi}_k^{(q)} &= \frac{n_k}{n}, \\ \hat{\mu}_k^{(q)} &= \frac{1}{n_k}\sum_{i=1}^{n} t_{ik}^{(q)}\hat{U}^{(q)t}y_i, \end{aligned}$$

where $n_k = \sum_{i=1}^{n} t_{ik}^{(q)}$ and $\hat{U}^{(q)}$ contains the d discriminative axes $\hat{u}_j^{(q)}$, $j = 1,...,d$, in column vectors, fitted in the F-step, at iteration q. The following proposition provides estimates for the remaining parameters for the 12 DLM models which have to be updated at each iteration of the FEM procedure.

Proposition 4.1.4. *At iteration q, the estimates for variance parameters of the 12 DLM models are:*

- *Model* $\text{DLM}_{[\Sigma_k\beta_k]}$*:*

$$\hat{\Sigma}_k^{(q)} = \hat{U}^{(q)t} C_k^{(q)} \hat{U}^{(q)}, \quad \hat{\beta}_k^{(q)} = \frac{\text{trace}(C_k^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.9}$$

- *Model* $\text{DLM}_{[\Sigma_k\beta]}$*:*

$$\hat{\Sigma}_k^{(q)} = \hat{U}^{(q)t} C_k^{(q)} \hat{U}^{(q)}, \quad \hat{\beta}^{(q)} = \frac{\text{trace}(C^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.10}$$

- *Model* $\text{DLM}_{[\Sigma\beta_k]}$*:*

$$\hat{\Sigma}^{(q)} = \hat{U}^{(q)t} C^{(q)} \hat{U}^{(q)}, \quad \hat{\beta}_k^{(q)} = \frac{\text{trace}(C_k^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.11}$$

- *Model* $\text{DLM}_{[\Sigma\beta]}$*:*

$$\hat{\Sigma}^{(q)} = \hat{U}^{(q)t} C^{(q)} \hat{U}^{(q)}, \quad \hat{\beta}^{(q)} = \frac{\text{trace}(C^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.12}$$

- *Model* $\text{DLM}_{[\alpha_{kj}\beta_k]}$*:*

$$\hat{\alpha}_{kj}^{(q)} = \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}, \quad \hat{\beta}_k^{(q)} = \frac{\text{trace}(C_k^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.13}$$

- *Model* $\text{DLM}_{[\alpha_{kj}\beta]}$*:*

$$\hat{\alpha}_{kj}^{(q)} = \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}, \quad \hat{\beta}^{(q)} = \frac{\text{trace}(C^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.14}$$

- *Model* $\text{DLM}_{[\alpha_k\beta_k]}$*:*

$$\hat{\alpha}_k^{(q)} = \frac{1}{d} \sum_{j=1}^{d} \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}, \quad \hat{\beta}_k^{(q)} = \frac{\text{trace}(C_k^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.15}$$

- *Model* $\text{DLM}_{[\alpha_k\beta]}$*:*

$$\hat{\alpha}_k^{(q)} = \frac{1}{d} \sum_{j=1}^{d} \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}, \quad \hat{\beta}^{(q)} = \frac{\text{trace}(C^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.16}$$

- *Model* $\text{DLM}_{[\alpha_j\beta_k]}$*:*

$$\hat{\alpha}_j^{(q)} = \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}, \quad \hat{\beta}_k^{(q)} = \frac{\text{trace}(C_k^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.17}$$

- *Model* $\text{DLM}_{[\alpha_j\beta]}$*:*

$$\hat{\alpha}_j^{(q)} = \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}, \quad \hat{\beta}^{(q)} = \frac{\text{trace}(C^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.18}$$

- *Model* $\text{DLM}_{[\alpha\beta_k]}$*:*

$$\hat{\alpha}^{(q)} = \frac{1}{d} \sum_{j=1}^{d} \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}, \quad \hat{\beta}_k^{(q)} = \frac{\text{trace}(C_k^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C_k^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.19}$$

- *Model* $\text{DLM}_{[\alpha\beta]}$*:*

$$\hat{\alpha}^{(q)} = \frac{1}{d} \sum_{j=1}^{d} \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}, \quad \hat{\beta}^{(q)} = \frac{\text{trace}(C^{(q)}) - \sum_{j=1}^{d} \hat{u}_j^{(q)t} C^{(q)} \hat{u}_j^{(q)}}{p-d}, \tag{4.1.20}$$

where the vectors $\hat{u}_j^{(q)}$ *are the discriminative axes fitted by the F-step at iteration* q, $C_k^{(q)} = \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} t_{ik}^{(q)} (y_i - \hat{m}_k^{(q)})(y_i - \hat{m}_k^{(q)})^t$ *is the soft covariance matrix of the* k*th group,* $\hat{m}_k^{(q)} = \frac{1}{n} \sum_{i=1}^{n} t_{ik}^{(q)} y_i$ *and finally* $C = \frac{1}{n} \sum_{k=1}^{K} n_k C_k$ *is the soft within-covariance matrix of the* K *groups.*

In order not to surcharge the notations, the index q of the current iteration of the Fisher-EM algorithm is not indicated in the following proofs. We also define the matrices $\tilde{W}$ and $\bar{W}$ such that $W = \tilde{W} + \bar{W}$. The matrix $\tilde{W}$ is defined as a $p \times p$ matrix containing the d first vectors of W completed by zeros such as $\tilde{W} = [U, 0_{p-d}]$ and $\bar{W} = W - \tilde{W}$ is defined by $\bar{W} = [0_d, V]$.

Proof. In the case of the model $\text{DLM}_{[\Sigma_k\beta_k]}$, at iteration q, the conditional expectation of the complete log-likelihood $Q(y_1, \ldots, y_n, \theta | \theta^{(q-1)})$ of the observed data $\{y_1, \ldots, y_n\}$ has the following form:

$$\begin{aligned} Q(\theta) =& \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik} \log(\pi_k \phi(y_i, \theta_k)) \\ =& \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik} \Big[-\frac{1}{2} \log(|S_k|) - \frac{1}{2} (y_i - m_k)^t S_k^{-1} (y_i - m_k) + \log(\pi_k) - \frac{p}{2} \log(2\pi) \Big], \end{aligned} \tag{4.1.21}$$

The maximization of $Q(\theta)$ conduces for the DLM models to the following estimates.

Estimation of π_k The prior probability π_k of the group k can be estimated by maximizing $Q(\theta)$ with respect to the constraint $\sum_{k=1}^{K}\pi_k = 1$ which is equivalent to maximize the Lagrange function:

$$L = Q(\theta) + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right),$$

where λ is the Lagrange multiplier. Then, the partial derivative of L with respect to π_k is $\partial L/\partial\pi_k = n_k/\pi_k + \lambda$. Consequently:

$$\forall k = 1,\ldots,K, \qquad \frac{\partial L}{\partial \pi_k} = 0 \Longleftrightarrow \frac{n_k}{\pi_k} + \lambda = 0 \Longleftrightarrow n_k + \lambda\pi_k = 0,$$

and:

$$\sum_{k=1}^{K}(n_k + \lambda\pi_k) = n + \lambda = 0 \Longrightarrow \lambda = -n.$$

Replacing λ by its value in the partial derivative conduces to an estimation of π_k by:

$$\hat{\pi}_k = \frac{n_k}{n}.$$

Estimation of μ_k The mean μ_k of the kth group in the latent space can be also estimated by maximizing the expectation of the complete log-likelihood (equation 4.1.21), which can be written in the following way:

$$Q(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik}\Big[-\frac{1}{2}\log(|S_k|) - \frac{1}{2}(y_i - \hat{U}\mu_k)^t S_k^{-1}(y_i - \hat{U}\mu_k) + \log(\pi_k) - \frac{p}{2}\log(2\pi)\Big]. \quad (4.1.22)$$

Consequently, the partial derivative of Q with respect to μ_k is $\partial Q(\theta)/\partial\mu_k = -\frac{1}{2}\sum_{i=1}^{n} t_{ik}\hat{U}^t(y_i - \hat{U}\mu_k)$. Setting this quantity to 0 gives:

$$\frac{\partial Q(\theta)}{\partial \mu_k} = 0 \Longleftrightarrow \sum_{i=1}^{n} t_{ik}\hat{U}^t y_i = \sum_{i=1}^{n} t_{ik}\mu_k.$$

and conduces to:

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i=1}^{n} t_{ik}\hat{U}^t y_i.$$

Model $\mathrm{DLM}_{[\Sigma_k\beta_k]}$ From Equation (4.1.8), the partial derivative of $Q(\theta)$ with respect to Σ_k has the following form:

$$\frac{\partial Q(\theta)}{\partial \Sigma_k} = -\frac{n_k}{2}\frac{\partial}{\partial \Sigma_k}\left[\log(|\Sigma_k|) + \mathrm{trace}\left(\Sigma_k^{-1}\hat{U}^t C_k \hat{U}\right)\right].$$

Using the matrix derivative formula of the logarithm of a determinant, $\partial \log(|A|)/\partial A = \left(A^{-1}\right)^t$, and of the trace of a product, $\partial \text{trace}(A^{-1}B)/\partial A = -\left(A^{-1}BA^{-1}\right)^t$, the equality of $\partial Q(\theta)/\partial \Sigma_k$ to the $d \times d$ zero matrix yields to the relation:

$$\Sigma_k^{-1} = \Sigma_k^{-1}\hat{U}^t C_k \hat{U} \Sigma_k^{-1},$$

and, by multiplying on the left and on the right by Σ_k, we find out the estimate of Σ_k:

$$\hat{\Sigma}_k = \hat{U}^t C_k \hat{U}. \tag{4.1.23}$$

The estimation of β_k is also obtained by maximizing Q subject to β_k:

$$\frac{\partial Q(\theta)}{\beta_k} = 0 \Longleftrightarrow \frac{p-d}{\beta_k} - \frac{\text{trace}(C_k)}{\beta_k^2} + \frac{1}{\beta_k^2}\sum_{j=1}^{d} \hat{u}_j^t C_k \hat{u}_j = 0,$$

and it is possible to conclude:

$$\hat{\beta}_k = \frac{\text{trace}(C_k) - \sum_{j=1}^{d} \hat{u}_j^t C_k \hat{u}_j}{p-d}. \tag{4.1.24}$$

Model $\text{DLM}_{[\Sigma_k \beta]}$ In this case, Q has the following form:

$$\begin{aligned}
Q(\theta) =& -\frac{1}{2}\Big(\sum_{k=1}^{K} n_k \Big[-2\log(\pi_k) + \text{trace}(\Sigma_k^{-1}\hat{U}^t C_k \hat{U}) + \log(|\Sigma_k|)\Big] \\
&+ \sum_{k=1}^{K} n_k (p-d)\ \log(\beta) + \sum_{k=1}^{K} \frac{n_k}{\beta}\Big[\text{trace}(C_k) - \sum_{j=1}^{d} \hat{u}_j^t C_k \hat{u}_j\Big]\Big), \\
=& -\frac{1}{2}\Big(\sum_{k=1}^{K} n_k \Big[-2\log(\pi_k) + \text{trace}(\Sigma_k^{-1}\hat{U}^t C_k \hat{U}) + \log(|\Sigma_k|) + \gamma\Big] \\
&+ n(p-d)\ \log(\beta) + \frac{1}{\beta}\Big[n\ \text{trace}(C) - n\sum_{j=1}^{d} \hat{u}_j^t C \hat{u}_j\Big]\Big),
\end{aligned}$$

where C is the empirical within covariance matrix of the whole dataset. Setting to 0 the partial derivative of $Q(\theta)$ conditionally to β implies $(p-d)/\beta - 1/\beta^2 \text{trace}(C) + 1/\beta^2 \sum_{j=1}^{d} \hat{u}_j^t C \hat{u}_j = 0$ and this conduces to:

$$\hat{\beta} = \frac{1}{p-d}\left(\text{trace}(C) - \sum_{j=1}^{d} \hat{u}_j^t C \hat{u}_j\right), \tag{4.1.25}$$

and the estimation of Σ_k is given by Equation (4.1.23).

Model $\text{DLM}_{[\Sigma \beta_k]}$ The quantity Q can be rewritten in this manner:

$$Q(\theta) = -\frac{1}{2}\Big(\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k)\Big] + n\ \log(|\Sigma|) + n\ \text{trace}(\Sigma^{-1}\hat{U}^t C\hat{U})\Big] + \sum_{k=1}^{K} n_k\Big[(p-d)\ \log(\beta_k) + \frac{1}{\beta_k}\left(\text{trace}(C_k) - \sum_{j=1}^{d}\hat{u}_j^t C_k \hat{u}_j\right) + \gamma\Big]\Big),$$

then, the partial derivative of $Q(\theta)$ with respect to Σ is:

$$\frac{\partial Q(\theta)}{\partial \Sigma} = -\frac{n}{2}\frac{\partial}{\partial \Sigma}\left[\log(|\Sigma|) + \text{trace}\left(\Sigma^{-1}\hat{U}^t C\hat{U}\right)\right]$$

and setting to 0 provides the estimation of Σ:

$$\hat{\Sigma} = \hat{U}^t C\hat{U}. \tag{4.1.26}$$

Finally, the estimation of β_k is provided by Equation (4.1.24).

Model $\text{DLM}_{[\Sigma\beta]}$ The estimations of Σ and β have been already considered above and are given by Equations (4.1.26 and 4.1.25).

Model $\text{DLM}_{[\alpha_{kj}\beta_k]}$ In this case, Q has the following form:

$$Q(\theta) = -\frac{1}{2}\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + \sum_{j=1}^{d}\left(\log(\alpha_{kj}) + \frac{\hat{u}_j^t C_k \hat{u}_j}{\alpha_{jk}}\right) + (p-d)\ \log(\beta_k) + \frac{1}{\beta_k}\sum_{j=d+1}^{p}\hat{v}_j^t C_k \hat{v}_j + \gamma\Big].$$

The partial derivative of Q with respect to α_{kj} is $\partial Q(\theta)/\partial\alpha_{kj} = -1/(2n_k)\left(1/\alpha_{kj} - \hat{u}_j^t C_k \hat{u}_j/\alpha_{kj}^2\right)$ and setting to 0 provides the estimate of α_{kj}:

$$\hat{\alpha}_{kj} = \hat{u}_j^t C_k \hat{u}_j. \tag{4.1.27}$$

The estimation of β_k is provided by Equation (4.1.24).

Model $\text{DLM}_{[\alpha_{kj}\beta]}$ The estimations of α_{kj} and β have been already considered above and are given by Equations (4.1.27 and 4.1.25).

Model $\text{DLM}_{[\alpha_k\beta_k]}$ For this model, the expectation of the complete log-likelihood $Q(\theta)$ has the following form:

$$\begin{aligned}
Q(\theta) = &-\frac{1}{2}\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + \sum_{j=1}^{d}\left(\log(\alpha_k) + \frac{\hat{u}_j^t C_k \hat{u}_j}{\alpha_k}\right) \\
&+ (p-d)\ \log(\beta_k) + \frac{1}{\beta_k}\sum_{j=1}^{p-d}\hat{v}_j^t C_k \hat{v}_j + \gamma\Big], \\
Q(\theta) = &-\frac{1}{2}\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k) + d\ \log(\alpha_k) + \frac{1}{\alpha_k}\sum_{j=1}^{d}\hat{u}_j^t C_k \hat{u}_j \\
&+ (p-d)\ \log(\beta_k) + \frac{1}{\beta_k}\sum_{j=1}^{p-d}\hat{v}_j^t C_k \hat{v}_j + \gamma\Big].
\end{aligned}$$

The partial derivative of $Q(\theta)$ with respect to α_k is $\partial Q(\theta)/\partial\alpha_k = -1/(2n_k)\left(d/\alpha_k - 1/\alpha_k^2 \sum_{j=1}^{d}\hat{u}_j^t C_k \hat{u}_j\right)$, and setting this quantity to 0, provides:

$$\hat{\alpha}_k = \frac{1}{d}\sum_{j=1}^{d}\hat{u}_j^t C_k \hat{u}_j. \tag{4.1.28}$$

On the other hand, the estimation of β_k is the same as in Equation (4.1.24).

Model $\text{DLM}_{[\alpha_k\beta]}$ The estimations of α_k and β are respectively provided by Equations (4.1.28) and (4.1.25).

Model $\text{DLM}_{[\alpha_j\beta_k]}$ In this case, $Q(\theta)$ has the following form:

$$\begin{aligned}
Q(\theta) = &-\frac{1}{2}\sum_{k=1}^{K} n_k\Big(-2\log(\pi_k) + \sum_{j=1}^{d}\left(\log(\alpha_j) + \frac{\hat{u}_j^t C_k \hat{u}_j}{\alpha_j}\right) \\
&+ (p-d)\log(\beta_k) + \frac{1}{\beta_k}\sum_{j=1}^{p-d}\hat{v}_j^t C_k \hat{v}_j + \gamma\Big), \\
Q(\theta) = &-\frac{1}{2}\Big(\sum_{k=1}^{K} n_k\Big[-2\log(\pi_k)\Big] + n\sum_{j=1}^{d}\log(\alpha_j) + n\sum_{j=1}^{d}\frac{\hat{u}_j^t C \hat{u}_j}{\alpha_j} \\
&+ \sum_{k=1}^{K} n_k\Big[(p-d)\ \log(\beta_k) + \frac{1}{\beta_k}\sum_{j=1}^{p-d}\hat{v}_j^t C_k \hat{v}_j + \gamma\Big]\Big).
\end{aligned}$$

The partial derivative of $Q(\theta)$ with respect to α_j is $\partial Q(\theta)/\partial\alpha_j = -n/2\left(1/\alpha_j - 1/\alpha_j^2 \hat{u}_j^t C \hat{u}_j\right)$ and setting to 0 implies:

$$\hat{\alpha}_j = \hat{u}_j^t C \hat{u}_j, \tag{4.1.29}$$

and the estimation of β_k is the same as in Equation (4.1.24).

Model $\text{DLM}_{[\alpha_j\beta]}$ The estimations of α_j and β are respectively provided by Equations (4.1.29) and (4.1.25).

Model $\text{DLM}_{[\alpha\beta_k]}$ In this case, $Q(\theta)$ has the following form:

$$\begin{aligned}
Q(\theta) = &-\frac{1}{2}\sum_{k=1}^{K} n_k\Big(-2\log(\pi_k) + d\ \log(\alpha) + \frac{1}{\alpha}\sum_{j=1}^{d}\hat{u}_j^t C_k \hat{u}_j + \\
&+ (p-d)\log(\beta_k) + \frac{1}{\beta_k}\sum_{j=1}^{p-d}\hat{v}_j^t C_k \hat{v}_j + \gamma\Big), \\
Q(\theta) = &-\frac{1}{2}\Big(\sum_{k=1}^{K} n_k[-2\log(\pi_k)] + n\ d\ \log(\alpha) + \frac{n}{\alpha}\sum_{j=1}^{d}\hat{u}_j^t C \hat{u}_j \\
&+ \sum_{k=1}^{K} n_k\Big[(p-d)\log(\beta_k) + \frac{1}{\beta_k}\sum_{j=1}^{p-d}\hat{v}_j^t C_k \hat{v}_j + \gamma\Big]\Big),
\end{aligned}$$

The partial derivative of $Q(\theta)$ with respect to α is $\partial Q(\theta)/\partial\alpha = -n/2\left(d/\alpha - 1/\alpha^2\sum_{j=1}^{d}\hat{u}_j^t C \hat{u}_j\right)$ and setting this quantity to 0, we end up with:

$$\hat{\alpha} = \frac{1}{d}\sum_{j=1}^{d}\hat{u}_j^t C \hat{u}_j. \tag{4.1.30}$$

The estimation of β_k is the same as in Equation (4.1.24).

Model $\text{DLM}_{[\alpha\beta]}$ The estimations of α and β have been already computed and are provided by Equations (4.1.30) and (4.1.25). □

4.2 Convergence of the Fisher-EM algorithm

The Fisher-EM algorithm previously introduced is based on the EM algorithm. However, a F-step is added between the traditional E-step and M-step. In particular, the projection matrix U, which is updated in the F-step, is not obtained by maximization of the conditional expectation of the log-likelihood which implies that the convergence of the Fisher-EM algorithm is not directly guaranteed. Therefore, in this section we present a result on the convergence of the Fisher-EM algorithm, in the isotropic case.

We consider the $\text{DLM}_{[\alpha\beta]}$ model which supposes a common and spherical covariance matrix for each class in the latent subspace ($\forall k \in \{1,\ldots,K\}$, $\Sigma_k = \alpha\mathbf{I}_d$) and in the orthogonal complement of the latent subspace as well ($\forall k \in \{1,\ldots,K\}$, $\beta_k = \beta$). Then, in this case, the following proposition holds:

Proposition 4.2.1. *In the case of the $DLM_{[\alpha\beta]}$ model, optimizing the Fisher's criterion with respect to U is equivalent to maximizing the conditional expectation of the log-likelihood function with respect to U.*

Proof. On the one hand, without loss of generality, by assuming that the covariance matrix of the data is the identity matrix, the Fisher's criterion can be rewritten as:

$$J(U) = \text{trace}((U^tU)^{-1}(U^tS_WU)) = \text{trace}(U^tS_WU),$$

where S_W stands for the within covariance matrix.

On the other hand, in the M-step of the Fisher-EM algorithm, let us consider the quantity $-2Q(\theta)$ where $Q(\theta)$ stands for the conditional expectation of the complete log-likelihood:

$$\begin{aligned}
-2Q(\theta) &= -2\sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}\log\left(\pi_k\phi\left(y_i;\theta_k\right)\right) \\
&= \sum_{k=1}^{K}\left[\sum_{i=1}^{n} t_{ik}[-2\log(\pi_k) + p\log\left(2\pi\right) + \log|S_k| + (y_i-m_k)^tS_k^{-1}(y_i-m_k)]\right] \\
&= \sum_{k=1}^{K}\left[\sum_{i=1}^{n} t_{ik}[\log|S_k| + (y_i-m_k)^tS_k^{-1}(y_i-m_k)]\right] + \gamma_1,
\end{aligned}$$

where $\gamma_1 = \sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}[-2\log(\pi_k) + p\log\left(2\pi\right)]$ is a constant term with respect to U.

Let us consider the homoscedastic case which implies that $S_k = S = W\Delta W^t$, $\forall k \in \{1,\ldots,K\}$ and in addition, let us consider the $\text{DLM}_{[\alpha\beta]}$ model meaning that the matrix Δ has the following form:

$$\Delta = \begin{bmatrix} \alpha\mathbf{I}_d & \mathbf{0}_{p-d} \\ \mathbf{0}_d & \beta\mathbf{I}_{p-d} \end{bmatrix}. \tag{4.2.1}$$

Consequently, given these assumptions, the quantity $\sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}\log|S_k| = \sum_{k=1}^{K} n_k\log|S| = \gamma_2$, with $n_k = \sum_{i=1}^{n} t_{ik}$, is independent of U and then becomes a constant with respect to U.

Moreover, by denoting A the quantity $\sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}(y_i-m_k)^tS^{-1}(y_i-m_k)$, we can state that:

$$\begin{aligned}
A &= \sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}(y_i-m_k)^tS^{-1}(y_i-m_k) \\
&= \text{trace}\left(S^{-1}\sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}(y_i-m_k)(y_i-m_k)^t\right) \\
&= n\text{trace}\left(S^{-1}S_W\right)
\end{aligned}$$

where $S_W = \frac{1}{n}\sum_{k=1}^{K} n_kC_k$ stands for the soft within covariance matrix and $C_k = \frac{1}{n_k}\sum_{i=1}^{n} t_{ik}(y_i - m_k)(y_i-m_k)^t$ the empirical covariance of the cluster k, with $n_k = \sum_{i=1}^{n} t_{ik}$. Besides, since

$S^{-1} = W\Delta^{-1}W^t$ where W satisfying $WW^t = W^tW = \mathbf{I}_p$, the quantity A can be rewritten as:

$$\begin{aligned} A &= n\text{trace}\left(\left(W^t\Delta W\right)^{-1} S_W\right) \\ &= n\text{trace}\left(\Delta^{-1}W^tS_WW\right). \end{aligned}$$

Let us introduce the matrices $\tilde{W} = [U, 0_{p-d}]$ and $\bar{W} = [0_d, V]$ such as $W = \tilde{W} + \bar{W}$, where U is a $p \times d$ matrix with $d < p$ and stands for the projection matrix of the latent space and V, its orthogonal complement. In this case, the relation $W^tS_WW = \tilde{W}^tS_W\tilde{W}^t + \bar{W}^tS_W\bar{W}$ can be easily stated since $\tilde{W}^tS_W\bar{W}$ and $\bar{W}^tS_W\tilde{W}$ are null matrices. Therefore, according to the diagonal form of the matrix Δ (see equation (4.2.1)) then the quantity A becomes:

$$\begin{aligned} A &= n\text{trace}\left(\Delta^{-1}\left(\tilde{W}^tS_W\tilde{W}^t + \bar{W}^tS_W\bar{W}\right)\right) \\ &= n\left(\text{trace}\left(\frac{1}{\alpha}U^tS_WU\right) + \text{trace}\left(\frac{1}{\beta}V^tS_WV\right)\right) \\ &= \frac{n}{\alpha}\text{trace}\left(U^tS_WU\right) + \gamma_3, \end{aligned}$$

where $\gamma_3 = n\text{trace}\left(\frac{1}{\beta}V^tS_WV\right)$ is independent of U. Consequently, the conditional expectation of the complete log-likelihood $Q(\theta)$ can be rewritten as:

$$-2Q(\theta) = \frac{n}{\alpha}\text{trace}\left(U^tS_WU\right) + \gamma,$$

where $\gamma = \gamma_1 + \gamma_2 + \gamma_3$ is a term independent of U.

Consequently, maximizing $Q(\theta)$ with respect to U is equivalent to minimizing the quantity $\text{trace}(U^tS_WU)$ which is, up to a constant, the Fisher's criterion. This allows us to conclude. □

Therefore, the Fisher-EM algorithm, in the case of the $\text{DLM}_{[\alpha\beta]}$ model, is a traditional EM algorithm and its convergence is then guaranteed.

4.3 Computational and practical aspects

4.3.1 Computational aspects

4.3.1.1 Initialization

Although the EM algorithm is widely used, it is also well-known that the performance of the algorithm is linked to its initial conditions. Several strategies were proposed in the literature for initializing the EM algorithm. A popular practice [15] executes the EM algorithm several times, from a random initialization, and keeps only the set of parameters associated with the highest likelihood. The use of k-means or a random partition are also standard approaches for initializing the algorithm. McLachlan and Peel [127] also proposed an initialization through

the parameters by generating the mean and the covariance matrix of each mixture component from a multivariate normal distribution parametrized by the empirical mean and empirical covariance matrix of the data. In practice, this latter initialization procedure works well but, unfortunately, it cannot be applied directly to the Fisher-EM algorithm since model parameters live in a space different from the observation space. A simple way to adapt this strategy could be to first determine a latent space using PCA and then simulate mixture parameters in this initialization latent space.

4.3.1.2 Model selection

In model-based clustering, it is frequent to consider several models in order to find the most appropriate model for the considered data. Since a model is defined by its number of components K and its parametrization, model selection allows to both select a parametrization and a number of components. Several criteria for model selection were proposed in the literature and the famous ones are penalized likelihood criteria. Classical tools for model selection include the AIC [2], BIC [157] and ICL [14] criteria. The Bayesian Information Criterion (BIC) is certainly the most popular and consists in selecting the model which penalizes the likelihood by $\gamma(\mathcal{M})/2\log(n)$ where $\gamma(\mathcal{M})$ is the number of parameters in model $\mathcal{M}$ and n is the number of observations. On the other hand, the AIC criterion penalizes the log-likelihood by $\gamma(\mathcal{M})$ whereas the ICL criterion adds the penalty $\sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log(t_{ik})$ to the one of the BIC criterion in order to favor well separated models. The value of $\gamma(\mathcal{M})$ is of course specific to the model selected by the practitioner (*cf.* Table 3.1). In the experiments of the following sections, the BIC criterion is used because of its popularity but the ICL criterion should also be well adapted in our context.

4.3.1.3 Stopping criterion and convergence monitoring

To decide whether the algorithm has converged or not, we propose to use the Aitken's criterion [126]. This criterion estimates the asymptotic maximum of the log-likelihood in order to detect in advance the algorithm convergence. Since the convergence of the EM algorithm could be slow in practice due to its linear convergence rate, it is often not necessary to wait for the actual convergence for obtaining a good parameter estimate under standard conditions. At iteration q, the Aitken's criterion is defined by $A^{(q)} = \left(\ell^{(q+1)} - \ell^{(q)}\right) / \left(\ell^{(q)} - \ell^{(q-1)}\right)$ where $\ell^{(q)}$ is the log-likelihood value at iteration q. Then, asymptotic estimate of the log-likelihood maximum is given by:

$$\ell_{\infty}^{(q+1)} = \ell^{(q)} + \frac{1}{1 - A^{(q)}}\left(\ell^{(q+1)} - \ell^{(q)}\right),$$

and the algorithm can be considered to have converged if $\left|\ell_{\infty}^{(q+1)} - \ell_{\infty}^{(q)}\right|$ is smaller than a small positive number (provided by the user). In practice, if the criterion is not satisfied after a maximum number of iterations (provided by the user as well), the algorithm stops. Afterward, it is possible to check whether the provided estimate is a local maximum by computing the

Hessian matrix (using finite differentiation) which should be positive semi definite. In the experiments presented in the following section, the convergence of the Fisher-EM algorithm has been checked using such an approach.

4.3.1.4 Computational cost

Obviously, since the additional F-step is iterative, the computational complexity of the Fisher-EM procedure is somewhat bigger than the one of the ordinary EM algorithm. The F-step requires $d(d-2)/2$ iterations due to the Gram-Schmidt procedure used for the orthogonalization of U. However, since d is at most equal to $K-1$ and is supposed to be small compared to p, the complexity of the F-step is not a quadratic function of the data dimension which could be large. Furthermore, it is important to notice that the complexity of this step does not depend on the number of observations n. Although the proposed algorithm is more time consuming than the usual EM algorithm, it is altogether actually usable on recent PCs even for large scale problems. Indeed, we have observed on simulations that Fisher-EM appears to be 1.5 times slower on average than EM (with a diagonal model). As an example, 24 seconds are on average necessary for Fisher-EM to cluster a dataset of 1 000 observations in a 100-dimensional space whereas EM requires 16 seconds.

4.3.2 Practical aspects

The DLM models, for which we propose the Fisher-EM algorithm as an estimation procedure, presents several practical and numerical interests among which the ability to visualize the clustered data, to interpret the discriminative axes and to deal with the so-called $n < p$ problem.

4.3.2.1 Choice of d and visualization in the discriminative subspace

The proposed DLM models are parametrized by the intrinsic dimension d of the discriminative latent subspace. The choice of d is already fixed such as $d \leq K-1$. This result is provided by Fisher's theory. Indeed, the projection matrix U which spans the discriminative latent space is obtained through a criterion based on the quantity $S^{-1}S_B^{(q)}$. Then the intrinsic dimension of the discriminative latent space depends on the rank of $S^{-1}S_B^{(q)}$ which is governed by the rank of $S_B^{(q)}$. As the soft between covariance matrix $S_B^{(q)}$ is composed of the sum of K matrices based on the term $(m_k^{(q)} - \bar{y})$ and by noting that $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{k=1}^{K} n_k m_k^{(q)}$ with $m_k^{(q)} = \frac{1}{n_k}\sum_{i=1}^{n} t_{ik}^{(q)} y_i$, then only $K-1$ matrices are linearly independent. Consequently, $S_B^{(q)}$ has rank at most equal to $K-1$ and the dimension of the latent subspace is $d \leq K-1$. This remark is very interesting since in practice, it enables to propose very parsimonious models (see Table 3.1 in Chapter 3 which depicts the number of free parameters to estimate when $d = K-1$ for the DLM models compared to some classical models.) Even though the actual value of d is strictly smaller than $K-1$ for the dataset at hand, we recommend in practice to

set $d = K - 1$ when numerically possible in order to avoid stability problems with the Fisher-EM algorithm. Furthermore, it is always better to extract more discriminative axes than to miss relevant dimensions and $K - 1$ is often in practice a small value compared to p. Besides, once the Fisher-EM algorithm has converged, then the "real" intrinsic dimension of the latent space can be computed. Indeed, the rank of the projected matrix $(\hat{U}^t S\hat{U})^{-1}\hat{U}^t\hat{S}_B\hat{U}$ where $\hat{U}$ stands for the fitted projection matrix and $\hat{S}_B$ the between covariance matrix subject to the partition obtained after convergence of the algorithm can be computed. Besides, a natural use of the discriminative axes may certainly be the visualization of the clustered data. Indeed, it is nowadays clear that the visualization help human operators to understand the results of an analysis. With the Fisher-EM algorithm, it is easy to project and visualize the cluster data into the estimated discriminative latent subspace if $K \leq 4$. On the one hand, if the estimated value of d is at most equal to 3, the practitioner can therefore visualize his data by projecting them on the d first discriminative axes and no discriminative information loss is to be deplored in this case. On the other hand, if the estimated value of d is strictly larger than 3, the visualization becomes obviously more difficult but the practitioner may simply use the 3 first discriminative axes which are the most discriminative ones among the $K - 1$ provided axes. Let us finally notice that the visualization quality is of course related to the clustering quality. Indeed, the visualization provided by the Fisher-EM algorithm may be disappointing if the clustering results are poor, due to a bad initialization for instance. A good solution to avoid such a situation may be to initialize the Fisher-EM algorithm with the "mini-EM" strategy or with the results of a classical EM algorithm.

4.3.2.2 Dealing with the $n < p$ problem

Another important and frequent problem when clustering high-dimensional data is known as high dimension and low sample size (HDSS) problem or the $n < p$ problem (we refer to [85, Chap. 18] for an overview). The $n < p$ problem refers to situations where the number of features p is larger than the number of available observations n. This problem occurs frequently in modern scientific applications such as genomic or mass spectrometry. In such cases, the estimation of model parameters for generative clustering methods is either difficult or impossible. This task is indeed very difficult when $n < p$ since generative methods require, in particular, to invert covariance matrices which are ill-conditioned in the best case or singular in the worst one. In contrast with other generative methods, the Fisher-EM procedure can overcome the $n < p$ problem. Indeed, the E and M steps of Fisher-EM do not require the determination of the last $p - d$ columns of W (see equations (4.1.1) and (4.1.19)–(4.1.20)) and, consequently, it is possible to modify the F-step to deal with situations where $n < p$. To do so, let $\bar{Y}$ denote the centered data matrix and T denote the soft partition matrix. We define in addition the weighted soft partition matrix $\tilde{T}$ where the jth column $\tilde{T}_j$ of $\tilde{T}$ is the jth column T_j of T divided by $n_j = \sum_{i=1}^{n} t_{ij}$. With these notations, the between covariance matrix S_B can be written in its matrix form $S_B = \bar{Y}^t\tilde{T}^t\tilde{T}\bar{Y}$ and the F-step aims to maximize,

under orthogonality constraints, the function $f(U) = \text{trace}\left((U^t\bar{Y}^t\bar{Y}U)^{-1}U^t\bar{Y}^t\tilde{T}^t\tilde{T}\bar{Y}U\right)$. It follows from the classical result of kernel theory, the Representer theorem [107], that this maximization can be done in a different space and that U can be expressed as $U = \bar{Y}H$ where $H \in \mathbb{R}^{n\times p}$. Therefore, the F-step reduces to maximize, under orthogonality constraints, the following function:

$$f(H) = \text{trace}\left((H^tGGH)^{-1}H^tG\tilde{T}^t\tilde{T}GH\right), \tag{4.3.1}$$

where $G = \bar{Y}\bar{Y}^t$ is the $n \times n$ Gram matrix. The solution U^* of the original problem can be obtained afterward from the solution H^* of (4.3.1) by multiplying it by $\bar{Y}$. Thus, the F-step reduces to the eigendecomposition under orthogonality constraints of a $n \times n$ matrix instead of a $p \times p$ matrix. This procedure is useful for the Fisher-EM procedure only because it allows to determine $d \leq n$ axes which are enough for Fisher-EM but not for other generative methods which require the computation of the p axes.

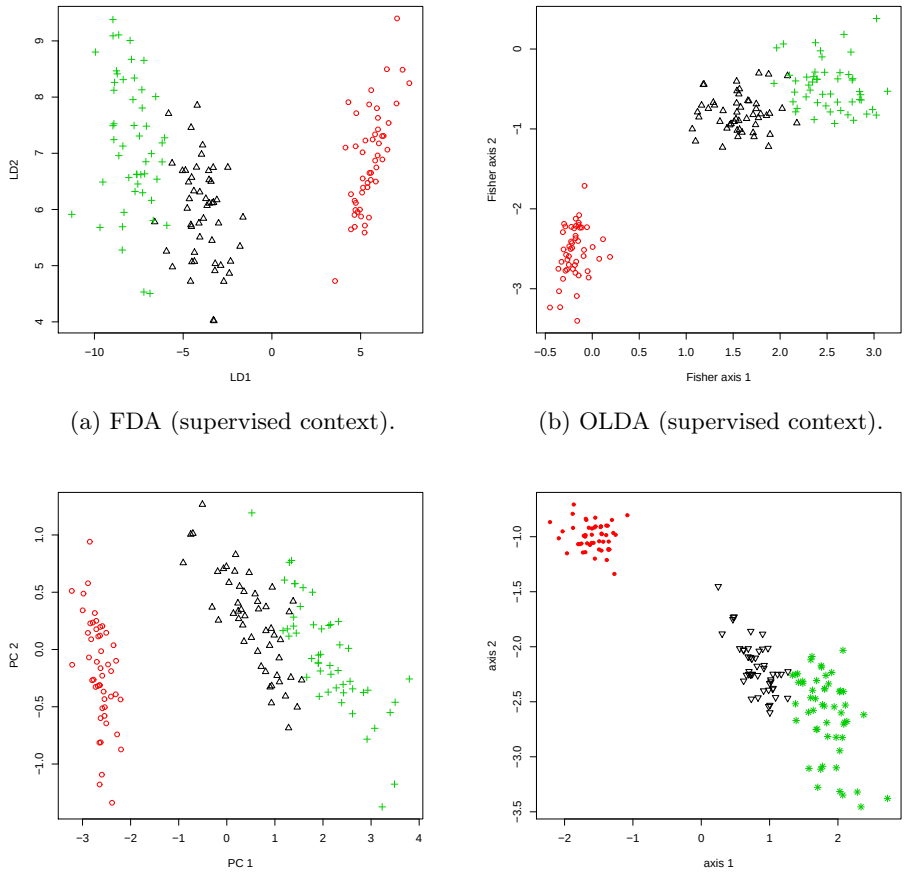
Chapter 5

Experimental results

This section presents experiments on simulated and real datasets, in order to highlight the main features of the clustering method introduced in the previous sections. In the first paragraph, the Fisher-EM algorithm is applied on the Fisher's irises as a glance to the Sir R. A. Fisher's work. The second paragraph aims to illustrate the convergence property of the Fisher-EM algorithm developed in Chapter 4, according to the evolution of Fisher's criterion. Moreover, this paragraph will compare the behaviors of the log-likelihood function, clustering accuracies and fitted error of the Fisher-EM with traditional algorithms (CEM and EM). The third paragraph aims to compare on simulations the differences between the three types of the F-step presented in Chapter 4. Then, a comparative study between subspace clustering approaches and the Fisher-EM algorithm will be presented on the well-known *Italian Wines* dataset. The robustness of our algorithm with high-dimensional data will be evaluated on simulations and the results will be compared to those obtained by traditional methods. The last experiment on simulations is developed in the 5th paragraph and aims to study the performance of BIC as a criterion for selecting both the DLM model and the number of components. Finally, the last paragraph will focus on comparing on benchmark datasets the efficiency of Fisher-EM with several linear and nonlinear existing methods, including the most recent ones.

5.1 An introductory example: the Fisher irises

As we chose to name the clustering algorithm proposed in this work after Sir R. A. Fisher, the least we can do is to first apply the Fisher-EM algorithm to the iris dataset that Fisher used in [54] as an illustration for his discriminant analysis. This dataset, in fact collected by E. Anderson [3] in the Gaspé peninsula (Canada), is made of 3 groups corresponding to different species of iris (*setosa*, *versicolor* and *virginica*) among which the groups *versicolor* and *virginica* are difficult to discriminate: they are at least not linearly separable. The dataset consists of 50 samples from each of 3 species and four features were measured from each sample. The four measurements are the length and the width of the sepal and the petal. This dataset is used here as an introductory example because of the link with Fisher's work but



(a) FDA (supervised context).

(b) OLDA (supervised context).

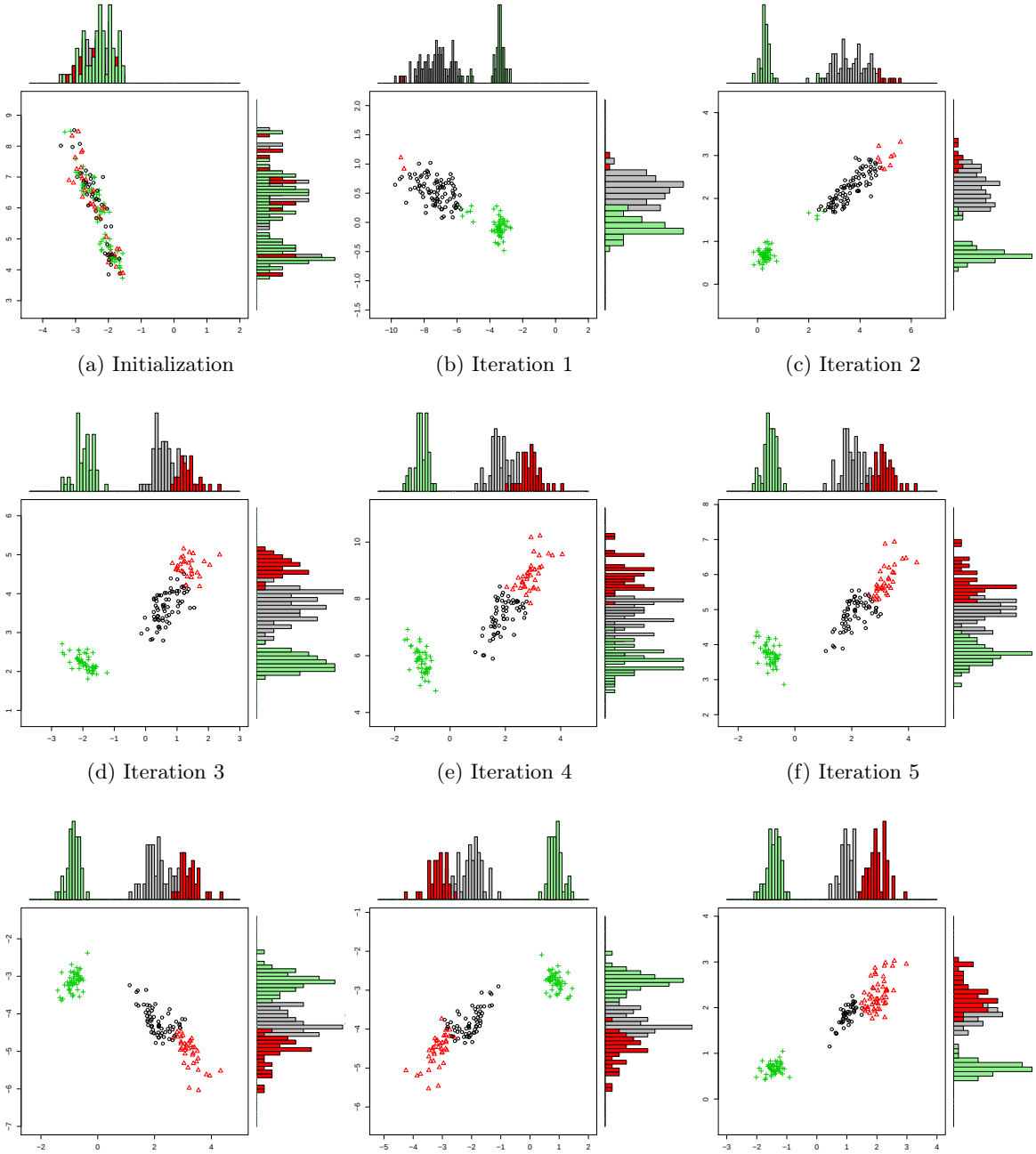
(c) PCA (unsupervised context).

(d) Fisher-EM (unsupervised context).

Figure 5.1: Projection of the irises on the traditional Fisher axes in a supervised case (a), on the orthogonal Fisher axes in a supervised case (b), on the 2 first principal components of PCA (c) and into the latent discriminative subspace estimated by Fisher-EM (d).

also of its popularity in the clustering community. Before applying the Fisher-EM algorithm, the data have been projected in the traditional Fisher subspace and in the orthogonal Fisher subspace in the supervised context as it is illustrated by Figure 5.1a and Figure 5.1b. This last representation has been obtained through an orthogonal linear discriminant analysis (OLDA) method developed by Ye *et al.* [186]. Moreover, Figure 5.1c stands for the projected data on the 2 first principal components of PCA. In this first experiment, Fisher-EM has been applied to the iris data and logically, the labels have been used only for performance evaluation and not for building discriminative axes. The Fisher-EM results have been compared to the ones obtained in the supervised case with the OLDA method [186]. The results have been obtained with a random initialization on the $\text{DLM}_{[\alpha_k\beta]}$ model where the number of classes has been fixed to 3. Figure 5.1d stands for the projection of the irises in the estimated discriminative space with Fisher-EM and Figures 5.3a and 5.3b show respectively the evolution of the log-likelihood and of the Fisher criterion on 25 iterations until convergence.

First of all, it can be observed in Figure 5.1d, that the estimated latent space discriminates



(a) Initialization (b) Iteration 1 (c) Iteration 2

(d) Iteration 3 (e) Iteration 4 (f) Iteration 5

(g) Iteration 10 (h) Iteration 20 (i) Iteration 25

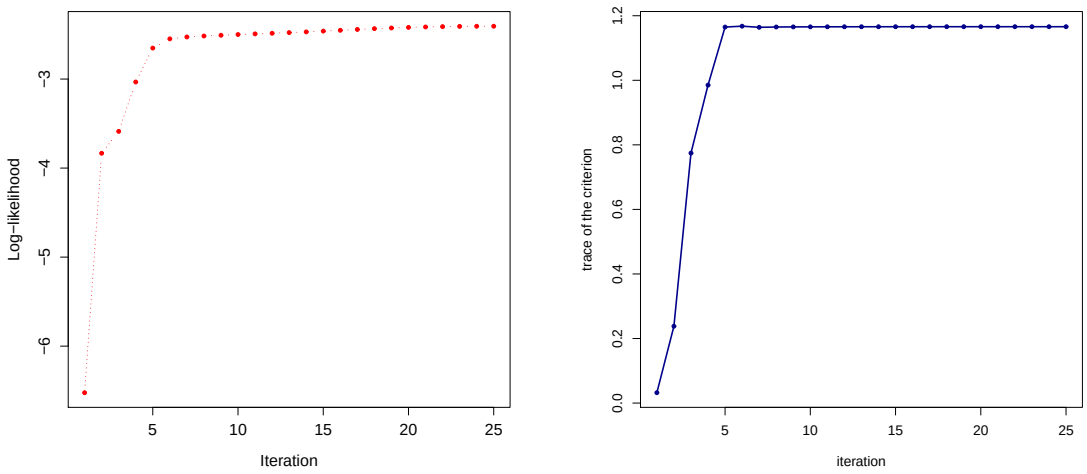
Figure 5.2: Steps of the Fisher-EM algorithm in the iris datasets.

almost perfectly the 3 different groups, compared to the representation of the data in the principal components of PCA in Figure 5.1c. Moreover, this discriminative latent space, built without knowing the class membership, is as informative in terms of structure as the one obtained in a supervised orthogonal context (see Figure 5.1b). Fisher-EM appears to be a powerful algorithm to find the intrinsic structure of the data in an unsupervised context, whatever the initialization is.

Figures 5.2a-i. illustrate at each iteration the data projected in the fitted latent discriminative subspace and on each axis, the empirical density of fitted clusters has been drawn. It can be observed in Figure 5.2a that the data are clustered randomly in the discriminative subspace, as the initialization is random. From the second iteration until the third one, the discrimination between 2 groups begins. From the third iteration (see Figure 5.2d), a structure of 3 different classes appears and we can see that the histograms, obtained on the axis plot, separate distinctly the densities. In particular, the second Fisher's axis well-discriminates the 3 clusters. Finally, the last iterations enable to refine the estimations of the means and the covariance matrices of 3 clusters until convergence.

The improvement of the partition in the latent space can also be evaluated by the evolution of the discriminative criterion based on the trace of $S^{-1}S_B^{(q)}$ where S and $S_B^{(q)}$ stands for respectively the total and the soft between covariance matrices computed at iteration (q). In fact, in Figure 5.3b, it can be observed that the Fisher's criterion is very small at the beginning, as the initialization of the partition is random. Then, this criterion increases drastically until the 5th iteration. After the 5th iteration, the criterion increases but very slowly until the maximum of the trace is reached. The final partition illustrated in Figure 5.2i. presents clusters which are well-separated and compact. Moreover, by considering the histograms of the data on the top and on the right of the plot, it can also be observed that the first axis estimated by Fisher-EM is very discriminative for the 3 classes.

For this experiment, the clustering accuracy has reached 98% with the $\text{DLM}_{[\alpha_k\beta]}$ model of Fisher-EM. Secondly, Figure 5.3a shows the monotonicity of the evolution of the log-likelihood and the convergence of the algorithm to a stationary state and Figure 5.3b shows the increase in the criterion which reaches a maximum. It can be observed that the log-likelihood and the Fisher's criterion have the same behavior: they both go up sharply until the 5th iteration and then the raising becomes very slow until a maximum state is reached. Table 5.1 presents the confusion matrices for the partitions obtained with supervised (OLDA) and unsupervised (Fisher-EM) methods from the MAP classification rule. OLDA has been used for the supervised case (reclassification of the learning data) whereas Fisher-EM has provided the clustering results. One can observe that the obtained partitions induced by both methods is almost the same. This confirms that Fisher-EM has correctly modeled both the discriminative subspace and the groups within the subspace. It is also interesting to look at the loadings provided by both methods. Table 5.2 stands for the linear coefficients of the discriminative axes estimated, on the one hand, in the supervised case (OLDA) and, on the other hand, in the unsupervised



(a) Log-likelihood function.

(b) Fisher's criterion (trace($s^{-1}s_B$))

Figure 5.3: Evolution of the associated log-likelihood (a) and of the Fisher criterion (b) in function of the iterations of Fisher-EM algorithm.

OLDA				Fisher-EM			
	cluster				*cluster*		
class	1	2	3	*class*	1	2	3
Setosa	50	0	0	Setosa	50	0	0
Versicolor	0	48	2	Versicolor	0	47	3
Virginica	0	1	49	Virginica	0	0	50
Misclassification rate = 0.02				*Misclassification rate = 0.02*			

Table 5.1: Confusion tables for the iris data with OLDA method (supervised) and Fisher-EM (unsupervised).

	OLDA		Fisher-EM	
	axis		*axis*	
variable	1	2	1	2
sepal length	0.209	0.044	-0.203	-0.108
sepal width	0.386	0.665	-0.422	0.088
petal length	-0.554	-0.356	0.602	0.736
petal width	-0.707	0.655	0.646	-0.662

Table 5.2: Fisher axes estimated by OLDA (supervised method) and by Fisher-EM (unsupervised method).

case (Fisher-EM). The first axes of each approach appear to be very similar and the scalar product of these axes is -0.996. This highlights the performance of the Fisher-EM algorithm in estimating the discriminative subspace of the data without knowing the class membership. Furthermore, according to these results, the 3 groups of irises can be mainly discriminated by the petal size, meaning that only one axis would be sufficient to discriminate the 3 iris species. Besides, this interpretation turns out to be in accordance with the recent work of Trendafilov and Joliffe [166] on variable selection in discriminant analysis *via* the LASSO.

5.2 Convergence properties of the Fisher-EM algorithm

This paragraph presents two experiments: the first experiment aims to implement the convergence property of the Fisher-EM algorithm defined in Chapter 4 based on the increase of the Fisher's criterion and the second one aims to compare the Fisher-EM algorithm in terms of log-likelihood values, clustering accuracies and estimation errors with the traditional EM and CEM algorithms.

5.2.1 Fisher-EM loglikelihood versus Fisher's criterion

This first experiment aims to validate the convergence property of the Fisher-EM algorithm developed in Chapter 4. A dataset consisting of 300 observations has been simulated according to the $\text{DLM}_{[\Sigma_k\beta]}$ model. We have simulated in the latent space of dimension 2 a Gaussian mixture model of 3 components with vector means:

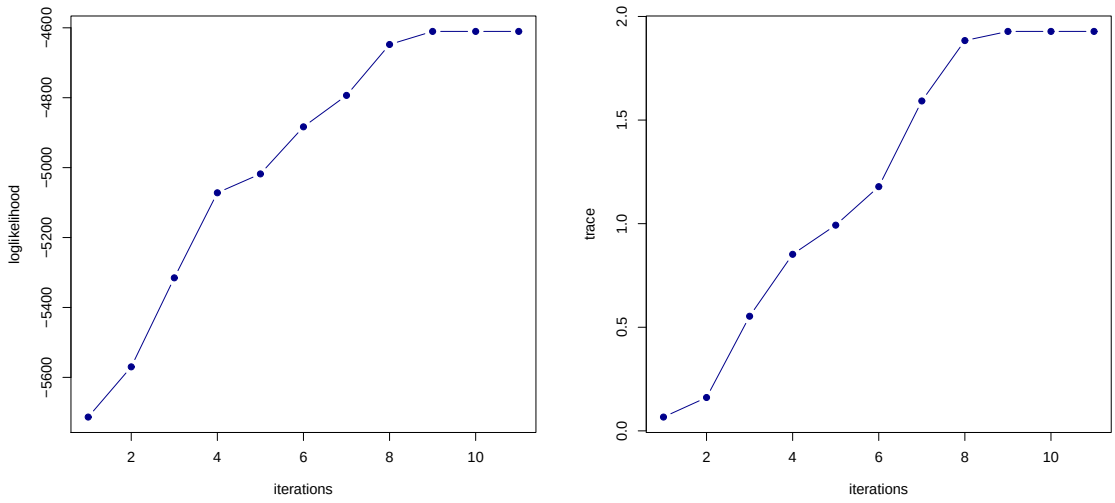
$$\mu_1 = (10, 0),\ \mu_2 = (-10, 0) \text{ and } \mu_3 = (0, 10)$$

respectively for the cluster 1,2 and 3, and their respective covariance matrices:

$$\Sigma_1 = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 2 \end{bmatrix},\ \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \text{ and } \Sigma_3 = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}.$$

Moreover, 8 orthogonal dimensions of Gaussian noise with variance $\beta = 10$ have been added. The transformation matrix W has been randomly simulated such that $W^tW = WW^t = I_p$ and, for this experiment, the dimension of the observed space is fixed to 10.

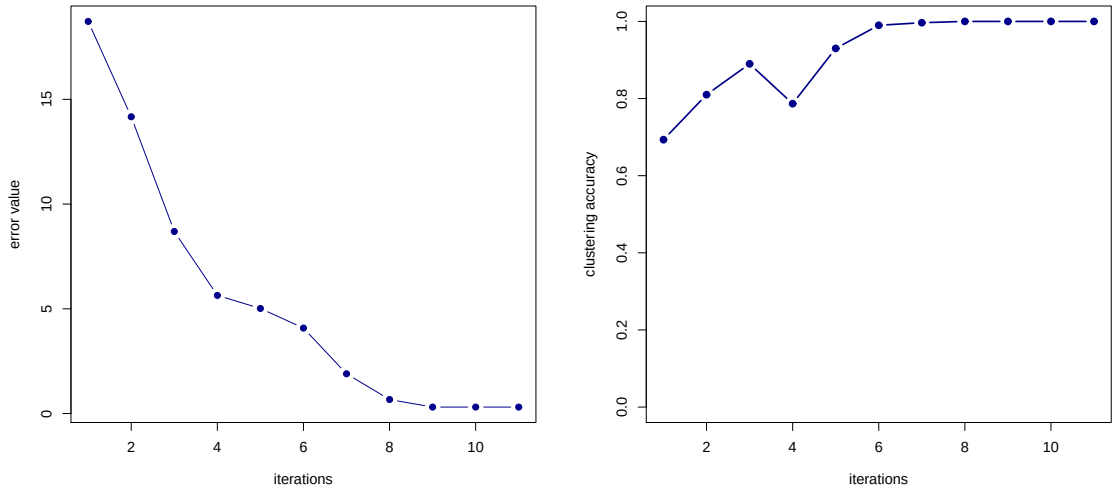
Figure 5.4a. stands for the evolution of the log-likelihood according to the iterations of Fisher-EM algorithm. Figure 5.4b. presents the evolution of the quantity $\text{trace}(s^{-1}s_B^{(q)})$ where s and $s_B^{(q)}$ are respectively the total and the soft between covariance matrices computed in the latent space at the iteration (q). First of all, it can be observed that the Fisher-EM algorithm has converged as the 11th iteration *i.e.* when the quantity $\left|\ell_\infty^{(q+1)} - \ell_\infty^{(q)}\right|$ based on the asymptotic estimation of the log-likelihood defined in Section 4.3.1.3 is become inferior to 10^{-6}. We can also see that both the log-likelihood and the trace criterion increase. At each iteration, the quantity $\text{trace}(s^{-1}s_B^{(q)})$ is maximized, which implies that the trace of $s^{-1}s_W^{(q)}$,



(a) Log-likelihood function.

(b) Fisher's criterion (trace($s^{-1}s_B$)).

Figure 5.4: Evolution of the loglikelihood function (a) versus Fisher's criterion (trace($s^{-1}s_B$)) evaluated in the latent space (b) according to the iterations of the Fisher-EM algorithm.



(a) Estimation errors.

(b) Clustering accuracy.

Figure 5.5: Evolution of the estimation errors evaluated in the latent space (a) and the clustering accuracy (b) according to the iterations of the Fisher-EM algorithm.

	cluster 1	cluster 2	cluster 3
True parameters	$\mu_1 = (\,10\,,\,0\,)$	$\mu_2 = (-10\,,\,0\,)$	$\mu_3 = (\,0\,,\,10\,)$
Fitted parameters	$\hat{\mu}_1 = (9.93, -0.28)$	$\hat{\mu}_2 = (-9.99, 0.29)$	$\hat{\mu}_3 = (-0.26, 9.97)$

Table 5.3: Fitted parameters of the means vectors of 3 components in the latent space obtained by Fisher-EM at convergence.

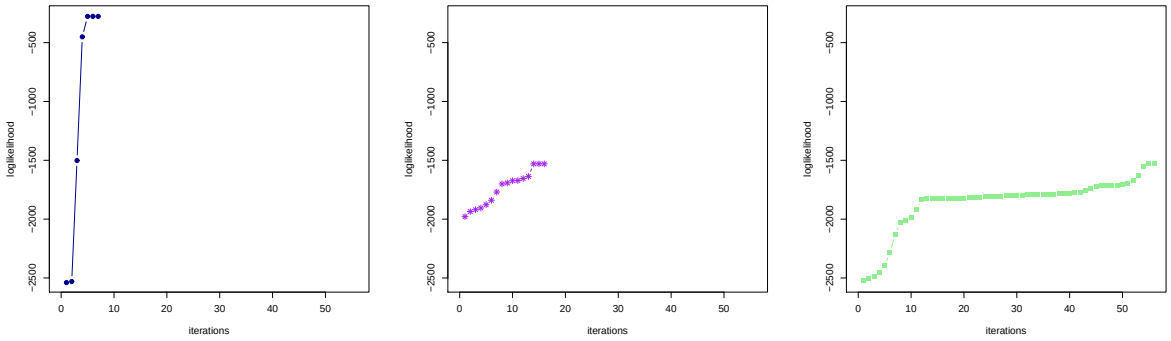
where $s_W^{(q)}$ stands for the soft within covariance matrix computed in the latent space, is minimized. This diminution has a positive effect in the log-likelihood function which consequently raises. Moreover, we can observe that the curves of Fisher's criterion and of the log-likelihood function have closed shapes and the maximum value of Fisher's criterion is reached as the 9th iteration and remains stable until convergence. In a certain way, this criterion can be viewed as an indicator of the convergence of the Fisher-EM algorithm. Finally, the evolution of the estimation error and the clustering accuracy are illustrated respectively in Figures 5.5a. and 5.5b. Indeed, since the true parameters of the mixture model are known, it is possible to evaluate the estimation error which has been computed by a Euclidean distance between means parameters fitted in the latent space and the true parameters. We can observe that the fitted error globally decreases according to the iterations of the Fisher-EM algorithm: the fitted error tends to zero when the algorithm has converged which suggests that the fitted parameters are closed to the true parameters of the mixture model. This result is detailed in Table 5.3 which stands for the fitted parameters obtained when the Fisher-EM algorithm has converged.

Finally the clustering efficiency evaluated with the true labels is illustrated in Figure 5.5b. and we can observe that it keeps on growing throughout the iterations of the Fisher-EM algorithm and tends to 1 at convergence.

5.2.2 Fisher-EM algorithm versus EM and CEM algorithms

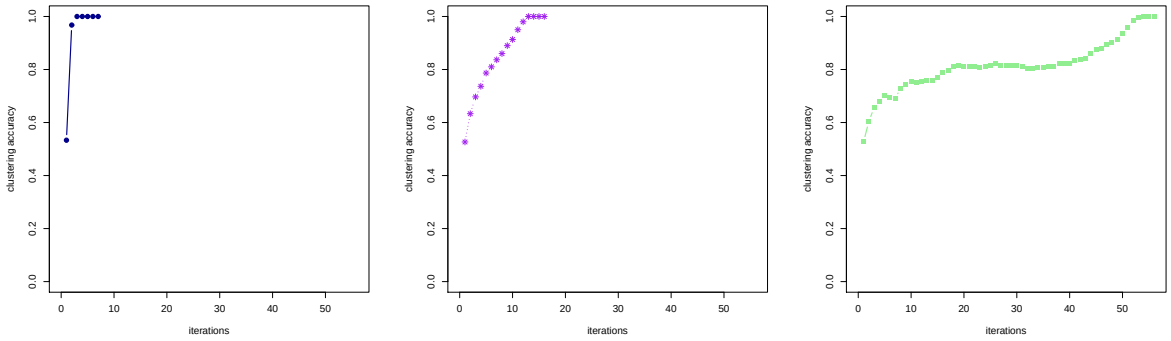
This second experiment aims to compare the behavior of the Fisher-EM algorithm with the standard CEM and EM algorithms in terms of evolution of log-likelihood function, fitted errors and clustering accuracies. To that end, we have considered the same simulation of the mixture model in the latent space as the previous experiment: we have generated 300 random vectors from each of $K = 3$ different two-dimensional multivariate normal distributions. The means parameters μ_1, μ_2, μ_3 and the covariance parameters $\Sigma_1, \Sigma_2, \Sigma_3$ of the 3 components in the latent space are the same as those defined previously. The only differences remain in the variance noise term which has been voluntary decreased ($\beta = 0.1$) and only 2 orthogonal dimensions of Gaussian noise have been added, in order to ease the clustering task and to obtain comparable clustering results between the Fisher-EM, EM and CEM algorithms.

For this experiment, the dimension of the observed space is fixed to 5 and the Fisher-EM, CEM and EM algorithms have been trained from a same random initialization. Figures 5.6 stand for the evolution of the log-likelihood according to the iterations of the 3 algorithms until convergence, Figures 5.7, the clustering accuracies and Figures 5.8, the fitted errors. First of all, it can be observed that Fisher-EM converges quicker than the CEM and EM algorithms. Indeed, as the 7th iteration Fisher-EM has converged conversely to the CEM and EM algorithms which converge respectively at the end of 16 and 56 iterations respectively. Besides, in this experiment, one re-find the advantage developed in Chapter 2 of the CEM algorithm based on its quickness to converge compared to the traditional EM algorithm. In



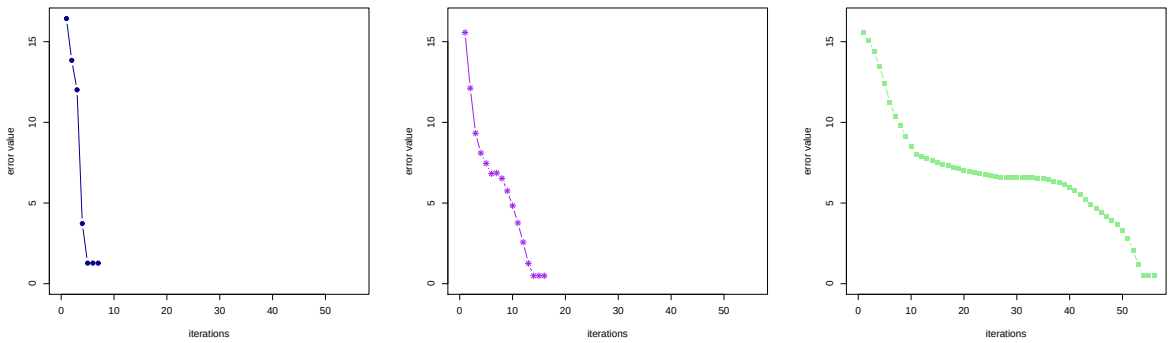
(a) Fisher-EM. (b) CEM. (c) EM.

Figure 5.6: Evolution of the log-likelihood according to the number of iterations for the Fisher-EM (a), CEM (b) and EM (c) algorithms.



(a) Fisher-EM. (b) CEM. (c) EM.

Figure 5.7: Evolution of the clustering accuracy (CA) according to the number of iterations for the Fisher-EM (a), CEM (b) and EM (c) algorithms.



(a) Fisher-EM. (b) CEM. (c) EM.

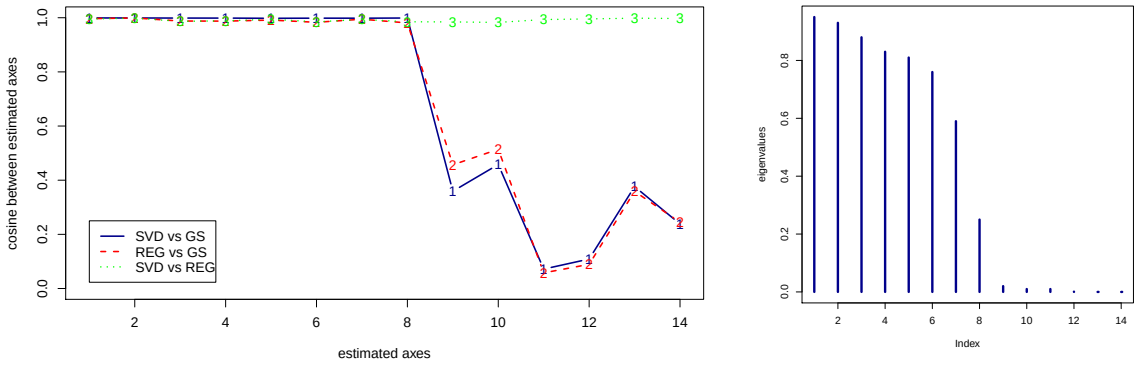
Figure 5.8: Evolution of the estimation error on the means according to the number of iterations for the Fisher-EM (a), CEM (b) and EM (c) algorithms.

Figures 5.7, we can observe that the quality of the 3 obtained partitions are very similar. Indeed, at convergence, the 3 algorithms reach almost 100% of clustering accuracy. This observation is confirmed by the evolution of the fitted error with the true parameters. As it can be observed in Figure 5.8a., the estimation error tends to zero for the 3 algorithms. However, in the Fisher-EM algorithm, we can observe that the error is a little bit higher than those obtained by CEM and EM. It can be explain by the easy clustering case of this experiment and the speed of the convergence of the Fisher-EM algorithm to a stable and very discriminative state.

5.3 Comparison of the 3 different optimizations for the F-step

We have introduced in Section 4.1.2 3 different ways to estimate the discriminative latent space in the F-step. This experiment aims to compare the 3 different approaches based either on the Gram-Schmidt (GS) orthogonalization procedure, on a singular value decomposition (SVD) or through a regression procedure (REG). For this simulation, 750 observations have been simulated following the $\text{DLM}_{[\Sigma\beta]}$ model with the parameters $\Sigma = 2\mathbf{I}_d$, $d = 8$ and $\beta = 15$. The difference between clusters happens to be entirely on the means vectors. The simulated dataset is made of 15 groups of 50 observations and each group is modeled by a Gaussian density in a 8-dimensional space completed by 7 orthogonal dimensions of Gaussian noise. The transformation matrix W has been randomly simulated such as $W^tW = WW^t = I_p$ and, for this experience, the dimension of the observed space is fixed to 30. In the aim to compare the efficiency to estimate the 3 different procedures of the F-step, we consider in this experiment the supervised context. Consequently, the true labels are used to initialize the Fisher-EM algorithm which is iterated once meaning that only one F-step and M-step are considered before re-classifying the data with an E-step.

Figure 5.9a stands for cosine value between the 14 axes estimated by the 3 procedures: the blue line is the cosine between the axes estimated by SVD and Gram-Schmidt procedures, the red line stands for the cosine computed between axes estimated by regression and Gram-Schmidt procedures and the green line stands for those obtained between the procedures SVD and regression. Since the intrinsic dimension of the latent subspace is theoretically at most equal to $d = K - 1$, then the cosines of 14 potential discriminative axes have been computed. Firstly, it can be observed in Figure 5.9a that the first estimated axes are exactly the same, whatever the procedure is. Moreover, the differences between the cosines are not significantly far from the first axis until the 8th axis. A little difference appears from the estimation of the second axis between the regression procedure and the 2 others but the cosines remain very close to 1 in general. From the 8th axis, we can observe a gap between the axes estimated by the Gram-Schmidt procedure and those estimated by SVD or by a linear regression. Therefore, an increasing gap in terms of classification accuracy is expected between these 3 procedures. However, such a difference does not exist as we can observe in Figure 5.10 which stands for



(a) Cosine between fitted axes.

(b) Scree plot

Figure 5.9: Evolution of the cosine between estimated axes according to the methodology used in the F-step : SVD, Gram-Schmidt or regression procedures (a) and scree plot of the eigenvalues of the matrix $s^{-1}s_B$ in the latent space (b).

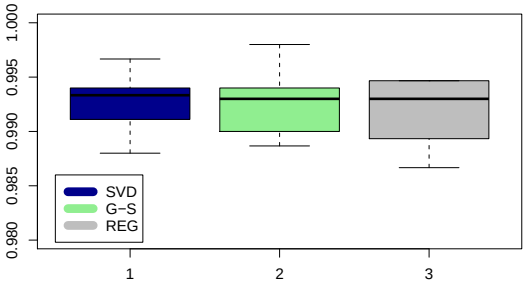


Figure 5.10: Boxplots of correct classification rates obtained on 25 replications for the 3 different procedures for the F-step (SVD, Gram-Schmidt and Regression).

Approach:	Elapsed real time	CPU time
SVD	4.2702 ± 0.0561	0.1164 ± 0.0460
G-S	4.4318 ± 0.0639	0.1556 ± 0.0654
REG	4.3884 ± 0.0683	0.1319 ± 0.0599

Table 5.4: Elapsed real time and CPU time computed for the 3 procedures of the F-step in the Fisher-EM algorithm.

the boxplots of correct classification rate obtained on 25 trials for each method. This can be explained by the fact that the last axes have a weak discriminative power. Besides, Figure 5.9b illustrates the scree plot of the eigenvalues associated to the eigen decomposition of the matrix $s^{-1}s_B$, where s and s_B stands for respectively the empirical covariance and the between covariance matrices in the fitted latent subspace. As we can observe, the discriminative power of axes decreases very quickly towards 0 from the 1st to the 8th axis and from the dimension 9, the eigenvalues are almost equal to 0. This is explained as that the most discriminative information is concentrated in the first axes which span the discriminative latent space. The main difference between the 3 procedures remains in the axes which have no discriminative power. Indeed, after convergence of the Fisher-EM algorithm, these last axes are removed since the intrinsic dimension is $d = \text{rank}(s_B) = 8$. Consequently, the 3 procedures used in the F-step are equivalent to estimate the latent subspace. However, in order to choose a method between the 3 approaches, the elapsed real time and the central processing unit (CPU) have been computed for each F-step procedure. The elapsed real time is the time taken from the start of the function until the end as measured by an ordinary clock and the CPU time is the amount of time charged for execution by the system on behalf of the calling process. Table 5.4 stands for these computation times according to the 3 procedures (SVD, G-S, REG). We can observe that the F-step computed by singular value decomposition is smaller than those obtained by the two other procedures. The performances between these 3 procedures are comparable in terms of clustering accuracy but the SVD procedure remains the quickest one. The SVD procedure has been preferred for the experiments on real datasets of high dimension.

5.4 Comparison with subspace clustering methods

This experiment aims to compare subspace clustering approaches with the family of DLM models on the Italian wines dataset. Originally, the *Wines* dataset has been introduced by Forina *et al.* [56] in 1986 and consisted in 28 chemical and physical properties on 3 different types of Italian wines. However, in this experiment, we consider a subset of this dataset traditionally used in clustering and available in the UCI Machine Learning data repository, which consists of 13 variables of the original dataset. The Italian wines dataset consists of 178 observations divided in 3 types: the *Barolo* (59), the *Grignolino* (71) and the *Barbera* (48).

For this experiment, the dataset has been scaled and 3 different subspace clustering approaches are compared: we have considered the family of 12 DLM models introduced in Chapter 3, the Hd-GMM family developed by Bouveyron *et al.* [22] and the mixture of common factor analyzer (MCFA) introduced by Baek *et al.* [8]. To that end, the HDclassif and mcfa softwares of R (R development Core Team 2004) have been used to fit respectively both Hd-GMM and MCFA models. For each approach, the algorithms have been executed on scaled data with the same random initializations and this experiment has been repeated 25 times. Nevertheless, the number of components has been fixed to 3, after checking that most

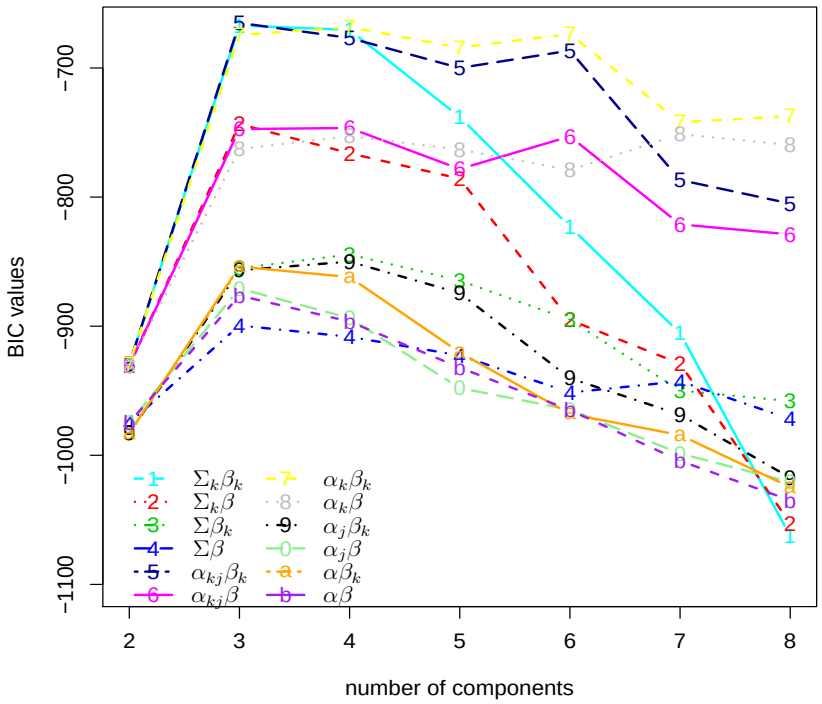
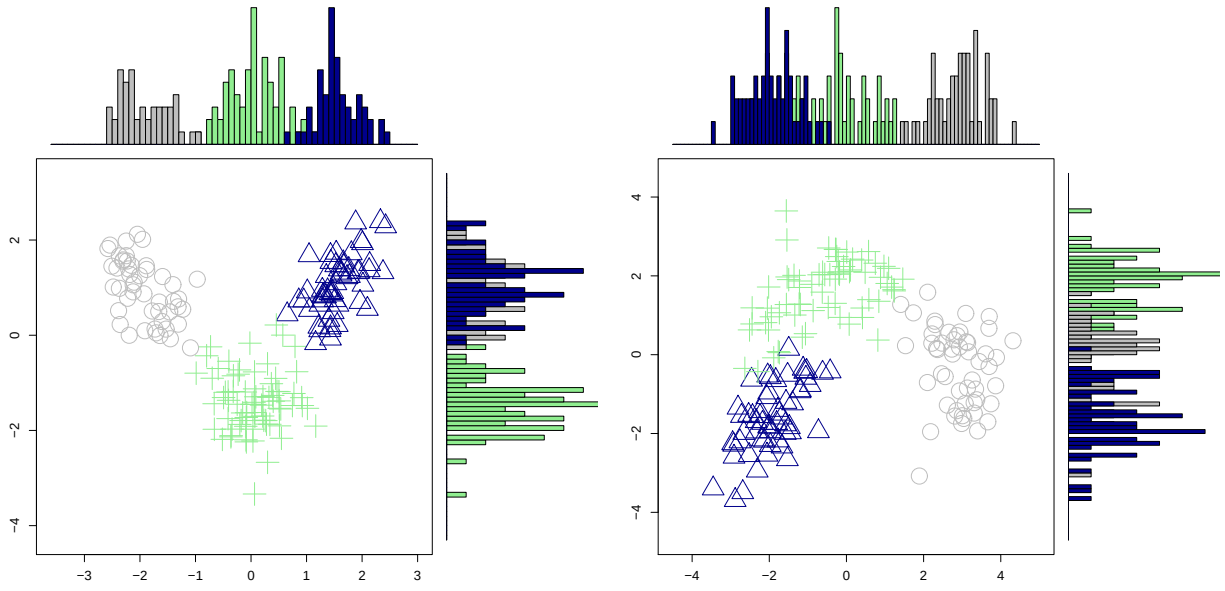


Figure 5.11: BIC values computed for the 12 DLM models and fitted for 2 until 8 components in the wines dataset.



(a) Fisher-EM. (b) MCFA.

Figure 5.12: Projection of the wines dataset fitted with the DLM$_{[\alpha_{kj}\beta_k]}$ model in the estimated latent discriminative subspace (a) and in the subspace estimated by the MCFA approach (b).

of the time, the BIC criterion selected a model with 3 components for each approach. Besides, Figure 5.11 stands for the BIC values computed amongst the 12 DLM models which have been fitted for a number of components varying from 2 to 8. It can be observed that the highest BIC is obtained for the $\text{DLM}_{[\alpha_{kj}\beta_k]}$ model with 3 components and this model will be used for the rest of the experiment. In the same manner, for the HDDC approach, the model selected by BIC is the model $[a_k b_k Q_k d]$ with intrinsic dimensions $d_k = 12$ for the 3 components. For MCFA, the model with a number of factor loadings $q = 3$ has been selected also by the BIC criterion. By considering these models, Table 5.5 stands for the average and the standard deviation of clustering accuracies and the maximum adjusted rand index obtained by the 3 approaches on the 25 iterations. It can be observed that the Fisher-EM algorithm executed with the $\text{DLM}_{[\alpha_{kj}\beta_k]}$ model is very stable and efficient on this dataset, compared to both HDDC and MCFA approaches. Its performances are equivalent in terms of clustering accuracy with the Hd-GMM model. However the DLM model provides more parsimonious and interpretable results since the intrinsic dimension of clusters are common and equal to 2 whereas in the Hd-GMM model, the subspaces are different for each cluster and their intrinsic dimension is equal to 12.

Moreover, cross tabulations obtained from the MAP rule are presented in Table 5.6 according to the 3 proposed approaches: these tables correspond to the best clustering accuracy reached for each approach amongst the 25 repetitions. Finally, contrary to HDDC, Fisher-EM and MCFA provide a visualization of the projected data in the fitted latent subspace since the subspace is common for the 3 groups. Besides, Figures 5.12a and 5.12b illustrate respectively the projection of the clustered data in the latent discriminative subspace fitted by the $\text{DLM}_{[\alpha_{kj}\beta_k]}$ and according to the MCFA approach. The histograms of the 3 clusters are also drawn on each estimated axis. These figures are linked to the best clustering accuracy reached respectively by the $\text{DLM}_{[\alpha_{kj}\beta_k]}$ (97.2%) and the MCFA model (96.6%) amongst the 25 repetitions. It can be observed that the representation of the clustered data in the subspace fitted by Fisher-EM is much more discriminative than the one obtained in the subspace estimated by the MCFA approach. Indeed, the clusters are compacter in the discriminative subspace fitted by the Fisher-EM algorithm whereas they overlap in the subspace fitted by the MCFA procedure.

Therefore, in this experiment, Fisher-EM which aims to estimate a discriminative subspace to cluster the data outperforms in average the traditional subspace clustering approaches (HDDC, MCFA) based on the maximization of the covariance of each cluster and provide a visualization which stresses the intrinsic structure of the dataset.

5.5 Simulation study: influence of the dimension

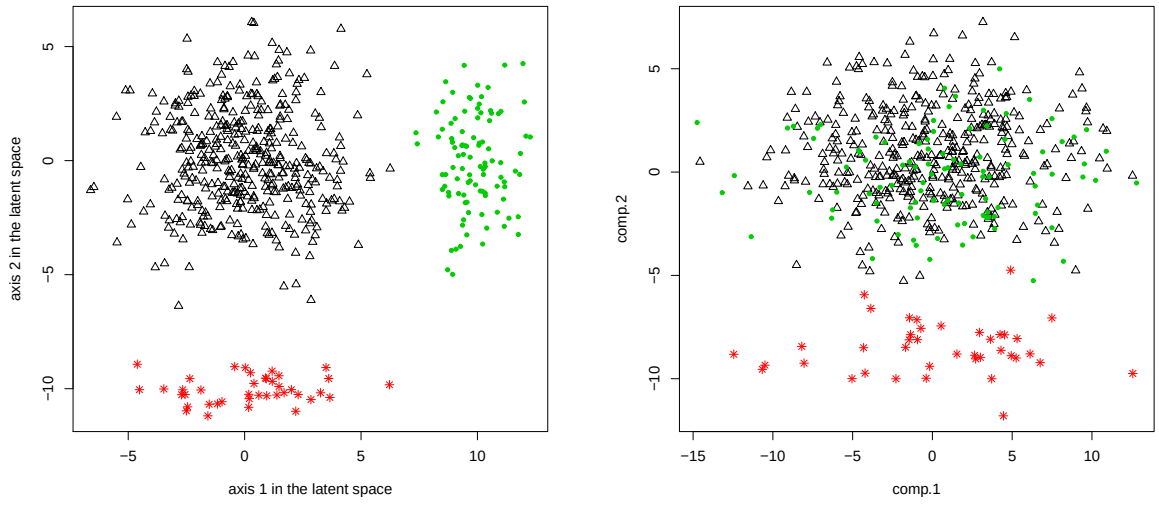
This fifth experiment aims to compare with traditional methods the stability and the efficiency of the Fisher-EM algorithm in partitioning high-dimensional data. Fisher-EM is compared here

approach:	model:	Clustering accuracy	Averaging Adjusted Rand Index
Fisher-EM	$\text{DLM}_{[\alpha_{kj}\beta_k]}$	97.19 ± 0.00	0.9129 ± 0.00
HDDC	Hd-GMM$[a_{kj}b_kQ_kd]$	96.90 ± 1.24	0.9062 ± 0.04
MCFA	-	91.62 ± 8.12	0.7075 ± 0.02

Table 5.5: Clustering accuracies and its corresponding standard deviation obtained from the MAP rule and averaged on 25 trials and the maximum adjusted rand index.

Fisher-EM according to $\text{DLM}_{[\alpha_{kj}\beta]}$				HDDC according to $[a_kb_kQ_kd]$				MCFA according to $q = 2$			
	cluster				*cluster*				*cluster*		
class	1	2	3	*class*	1	2	3	*class*	1	2	3
Barbera	47	1	0	Barbera	47	1	0	Barbera	48	0	0
Grignolino	1	70	0	Grignolino	0	71	0	Grignolino	4	67	0
Barolo	0	3	56	Barolo	0	2	57	Barolo	0	2	57
Misclassification rate = 2.81%				*Misclassification rate = 1.68%*				*Misclassification rate = 3.37%*			

Table 5.6: Cross tabulations of the MAP classifications obtained from the $\text{DLM}_{[\alpha_{kj}\beta]}$, Hd-$\text{GMM}_{[a_kb_kQ_kd_k]}$ and MCFA ($q = 2$) models with $K = 3$ in the case of their maximum clustering accuracy reached amongst the 25 repetitions.



(a) Discriminative latent subspace

(b) Projection in the 2 first components of PCA.

Figure 5.13: Visualization of the simulated data: data in their latent space (left) and data projected on the 2 first principal components (right).

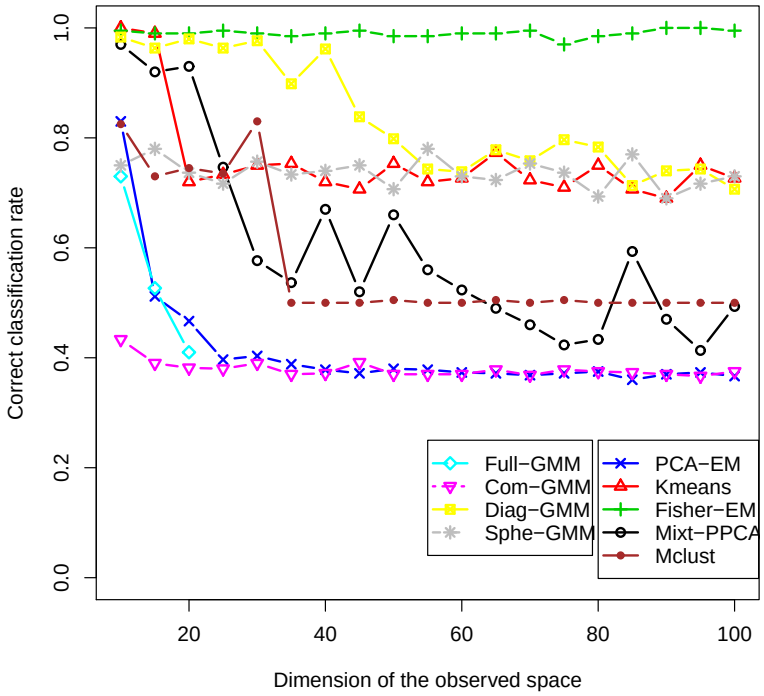
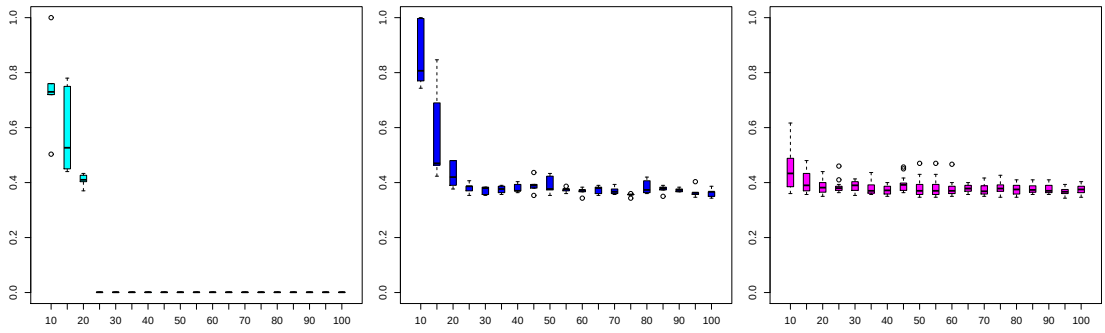


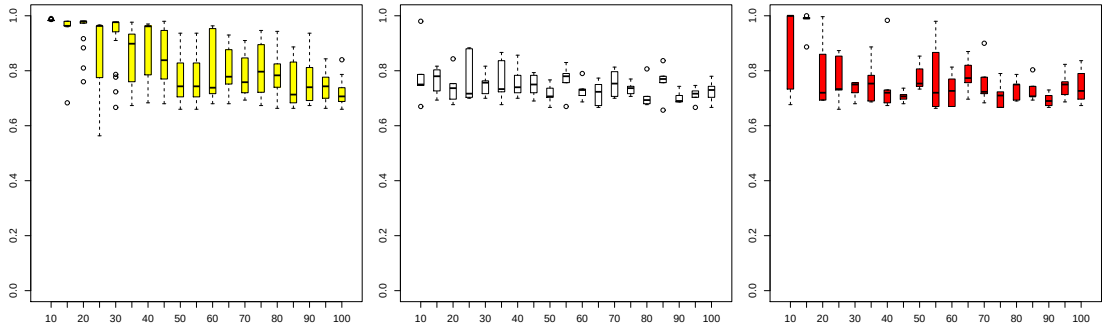
Figure 5.14: Influence of the dimension of the observed space on the correct classification rate for Full-GMM, PCA-EM, Com-GMM, Mixt-PPCA, k-means, Diag-GMM, Sphe-GMM and Fisher-EM algorithms.

with the standard EM algorithm (Full-GMM) and its parsimonious models (Diag-GMM, Sphe-GMM and Com-GMM), the EM algorithm applied in the first components of PCA explaining 90% of the total variance (PCA-EM), the k-means algorithm and the mixture of probabilistic principal component analyzers (Mixt-PPCA).

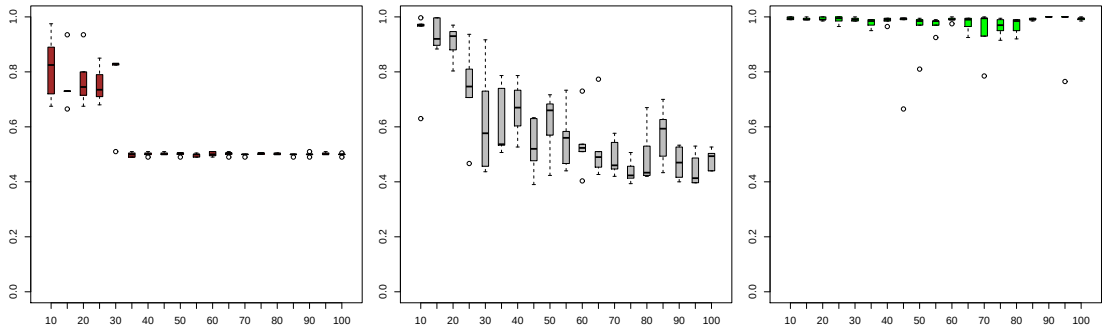
For this simulation, 600 observations have been simulated following the $\text{DLM}_{[\alpha_{kj}\beta_k]}$ model proposed in Chapter 3. The simulated dataset is made of 3 unbalanced groups and each group is modeled by a Gaussian density in a 2-dimensional space completed by orthogonal dimensions of Gaussian noise. The transformation matrix W has been randomly simulated such as $W^tW = WW^t = I_p$ and, for this experience, the dimension of the observed space varies from 5 to 100. The left panel of Figure 5.13 shows the simulated data in their 2-dimensional latent space whereas the right panel presents the projection of 50-dimensional observed data on the two first axes of PCA in the observed space. As one can observe, the representation of the data on the two first principal components is actually not well suited for clustering these data while it exists a representation which discriminates perfectly the 3 groups. Moreover, to make the results of each method comparable, the same randomized initialization has been used for the 8 algorithms. The experimental process has been repeated 20 times for each dimension of the observed space in order to see both the average performances and their variances.



(a) Full-GMM (b) PCA-Full-GMM (c) Com-GMM

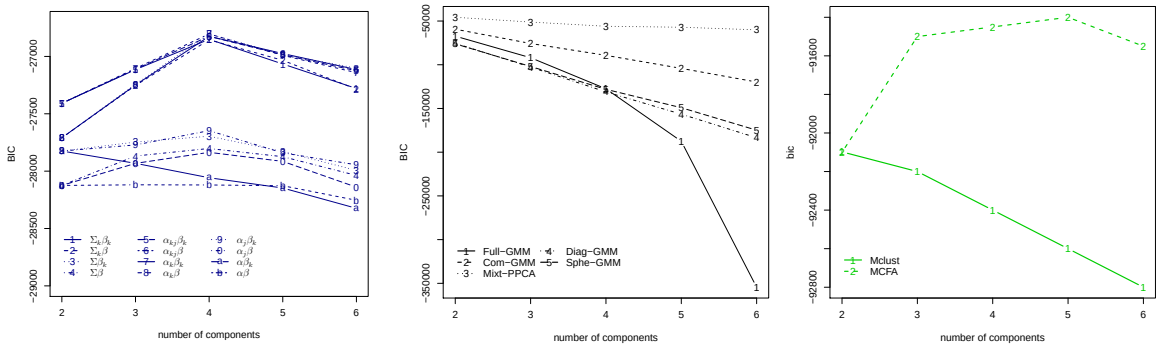


(d) Diag-GMM (e) Sphe-GMM (f) k-means



(g) Mclust (h) Mixt-PPCA (i) Fisher-EM

Figure 5.15: Boxplots of 9 clustering methods: standard deviation of the clustering accuracy in function of the evolution of the dimension of the observed space.

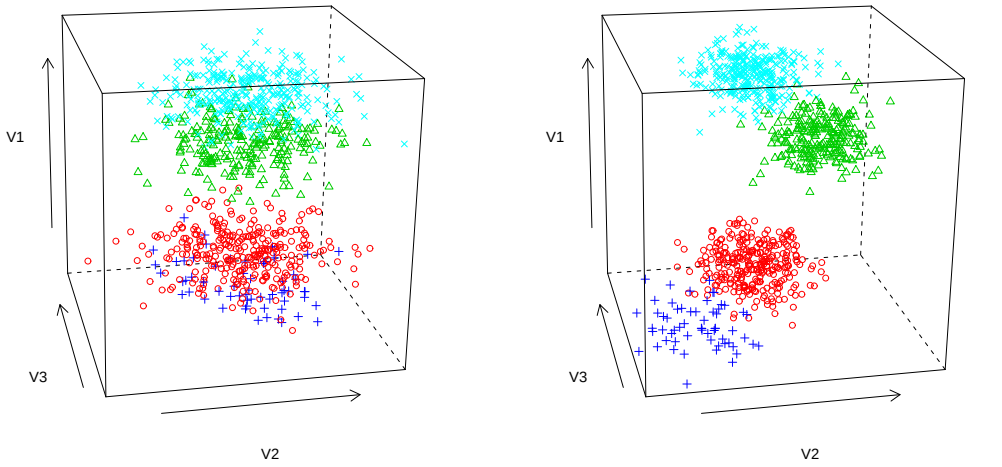


(a) 12 DLM models. (b) Traditional GMM. (c) MCFA and Mclust.

Figure 5.16: BIC values subject to the number of components varying from 2 until 6 and to different approaches: (a) the family of 12 DLM models, (b) Traditional GMM models and (c) Mclust and MCFA.

Figure 5.14 presents the evolution of the clustering accuracy of each method (EM, PCA-EM, k-means, Mixt-PPCA, Fisher-EM, Diag-GMM, Sphe-GMM and Com-GMM) according to the data dimensionality and Figure 5.15 presents their respective boxplots.

First of all, it can be observed that the Full-GMM, PCA-EM and Com-GMM have decreasing performances when the dimension increases. In fact, the Full-GMM model does not work upon the 15th dimension and still remains unstable in a low dimensional space as well as the Com-GMM model. Similarly, the performances of PCA-EM fall down as the 10th dimension. This can be explained as the latent subspace provided by PCA does not allow to well discriminate the groups, as already suggested by Figure 5.13. However, the PCA-EM approach can be used whatever the dimension is whereas Full-GMM cannot be used as the 20th dimension because of numerical problems linked to singularity of the covariance matrices. Moreover, their boxplots show a large variation on the clustering accuracy. Secondly, Sphe-GMM, Diag-GMM and k-means present the same trend with high performances in low-dimensional spaces which decrease until they reach a clustering accuracy of 0.75. However, Diag-GMM seems to resist a little bit more than k-means to the dimension increasing. Mixt-PPCA and Mclust both follow the same tendency as the previous methods but from the 30th dimension their performances fall down until the clustering accuracy reaches 0.5. The poor performances of Mixt-PPCA can be explained as Mixt-PPCA models each group in a different subspace whereas the model used for simulating the observations assumes a common discriminative subspace. Finally, Fisher-EM appears to be more effective than the other methods and, more importantly, it remains very stable while the data dimensionality increases. Furthermore, the boxplot associated with the Fisher-EM results suggests that it is a steady algorithm which succeeds in finding out the discriminative latent subspace of the data even with random initializations.



(a) Projection in the PCA subspace.

(b) Projection in the fitted latent subspace.

Figure 5.17: Projection of the data in the 2 first principal components of PCA (a) and in the discriminative latent subspace estimated by the Fisher-EM algorithm (b).

5.6 Simulation study: model selection

This last experiment on simulations aims to study the performance of BIC for both model and component number selection. For this experiment, 4 Gaussian components, of 75 observations each, have been simulated according to the $\text{DLM}_{[\alpha_k\beta]}$ model in a 3-dimensional space completed by 47 orthogonal dimensions of Gaussian noise (the dimension of the observation space is therefore $p = 50$). The transformation matrix W has been again randomly simulated such as $W^tW = WW^t = I_p$. Table 5.7 presents the BIC values and the adjusted rand index for the family of DLM models and, in a comparative purpose, the BIC values for 7 other methods already used in the last experiments: EM with the Full-GMM, Diag-GMM, Sphe-GMM and Com-GMM models, Mixt-PPCA, Mclust [58] (with model [EEE] which is the most appropriate model for these data), PCA-EM and MCFA. Moreover, BIC is computed for different partition numbers varying between 2 and 6 clusters.

First of all, one can observe that the BIC values, linked to the models which are different from the DLM model, are very low compared to the DLM models. This suggests that the models which best fit the data are the DLM models. Secondly, 10 of the 12 DLM models select the right number of components ($K = 4$). Moreover, Figure 5.16a stands for the BIC values obtained for the 12 DLM models with respect to the number of components. It can be observed 2 groups between the DLM models: in the first hand, the models which supposes variable covariance matrices in the latent space ($\text{DLM}_{[\Sigma_k\beta_k]}$, $\text{DLM}_{[\Sigma\beta_k]}$, $\text{DLM}_{[\Sigma_k\beta]}$,$\text{DLM}_{[\alpha_{kj}\beta_k]}$, $\text{DLM}_{[\alpha_{kj}\beta]}$, $\text{DLM}_{[\alpha_k\beta_k]}$ and $\text{DLM}_{[\alpha_k\beta]}$) and in the second hand, the models assuming common covariance matrices for each group ($\text{DLM}_{[\Sigma\beta]}$, $\text{DLM}_{[\alpha_j\beta_k]}$, $\text{DLM}_{[\alpha_j\beta]}$, $\text{DLM}_{[\alpha\beta_k]}$ and $\text{DLM}_{[\alpha\beta]}$). It appears that the models with variable covariance matrices represent the best models in

	number of components				
models	2	3	4	5	6
$DLM_{[\Sigma_k\beta_k]}$	-27408	-27114	-26850	-27066	-27279
	(0.498)	*(0.707)*	*(0.993)*	*(0.965)*	*(0.863)*
$DLM_{[\Sigma_k\beta]}$	-27709	-27252	-26855	-27034	-27285
	(0.498)	*(0.707)*	*(0.993)*	*(0.960)*	*(0.866)*
$DLM_{[\Sigma\beta_k]}$	-27824	-27746	-27695	-27828	-27993
	(0.498)	*(0.672)*	*(0.891)*	*(0.841)*	*(0.818)*
$DLM_{[\Sigma\beta]}$	-28124	-27867	-27802	-27873	-28037
	(0.498)	*(0.669)*	*(0.909)*	*(0.802)*	*(0.808)*
$DLM_{[\alpha_{kj}\beta_k]}$	-27408	-27111	-26825	-26986	-27125
	(0.498)	*(0.707)*	*(0.987)*	*(0.977)*	*(0.873)*
$DLM_{[\alpha_{kj}\beta]}$	-27709	-27249	-26825	-26975	-27114
	(0.498)	*(0.707)*	*(0.993)*	*(0.941)*	*(0.863)*
$DLM_{[\alpha_k\beta_k]}$	-27408	-27105	-26804	-26991	-27142
	(0.498)	*(0.707)*	*(0.993)*	*(0.971)*	*(0.874)*
$DLM_{[\alpha_k\beta]}$	-27709	-27243	**-26801**	-26984	-27102
	(0.498)	*(0.707)*	*(0.993)*	*(0.936)*	*(0.886)*
$DLM_{[\alpha_j\beta_k]}$	-27824	-27771	-27647	-27840	-27944
	(0.498)	*(0.665)*	*(0.861)*	*(0.849)*	*(0.832)*
$DLM_{[\alpha_j\beta]}$	-28124	-27932	-27836	-27914	-28139
	(0.498)	*(0.672)*	*(0.890)*	*(0.802)*	*(0.806)*
$DLM_{[\alpha\beta_k]}$	-27824	-27931	-28057	-28147	-28324
	(0.498)	*(0.655)*	*(0.498)*	*(0.803)*	*(0.791)*
$DLM_{[\alpha\beta]}$	-28124	-28119	-28120	-28127	-28254
	(0.498)	*(0.669)*	*(0.498)*	*(0.791)*	*(0.784)*
Full-GMM	-67268	-91956	-127490	-187520	-354461
	(0.001)	*(0.002)*	*(0.004)*	*(0.004)*	*(0.007)*
Com-GMM	-59343	-75680	-89253	-104157	-119745
	(0.498)	*(0.497)*	*(0.498)*	*(0.498)*	*(0.498)*
Mixt-PPCA	-45706	-51178	-56000	-57091	-60051
	(0.498)	*(0.713)*	*(0.890)*	*(0.905)*	*(0.919)*
Diag-GMM	-75530	-102638	-131043	-156537	-183476
	(0.498)	*(0.713)*	*(0.785)*	*(0.611)*	*(0.832)*
Sphe-GMM	-76064	-101992	-128091	-149127	-174868
	(0.498)	*(0.707)*	*(0.886)*	*(0.624)*	*(0.876)*
PCA-EM	-50017	-67729	-89604	-117674	-150783
	(0.498)	*(0.701)*	*(0.707)*	*(0.494)*	*(0.492)*
$Mclust_{[EII]}$	-91920	-92160	-92405	-92650	-92887
	(0.498)	*(0.497)*	*(0.496)*	*(0.494)*	*(0.492)*
MCFA	-91895	-91457	-91399	-91370	-91515
	(0.332)	*(0.701)*	*(0.707)*	*(0.986)*	*(0.712)*

Table 5.7: BIC values for model selection and adjusted rand index in brackets.

terms of BIC values and they correspond also to the best rand adjusted indexes (given in brackets in Table 5.7) compared to the other group. Concerning the GMM models constrained or not (Full-GMM, Com-GMM, Diag-GMM, Sphe-GMM, Mclust) and Mixt-PPCA, they all under-estimate the number of clusters and select only 2 components whereas MCFA tends to over-estimate it as it can be observed in Figures 5.16b and 5.16c. BIC has the highest value for the $\text{DLM}_{[\alpha_k\beta]}$ model with 4 components which is actually the model used for simulating the data. Finally, the right-hand side of Figure 5.17 presents the projection of the data on the discriminative subspace of 3 dimensions estimated by Fisher-EM with the $\text{DLM}_{[\alpha_k\beta]}$ model whereas the left-hand side figure represents the projection of the data on the 3 first principal components of PCA. As one can observe, in the PCA case, the axes separate only 2 groups, which is in accordance with the model selection pointed out by BIC for this method. Conversely, in the Fisher-EM case, the 3 discriminative axes separate well the 4 groups and such a representation could clearly help the practitioner in understanding the clustering results.

5.7 Real data set benchmark

This last experimental paragraph will focus on comparing, on real-world datasets, the efficiency of Fisher-EM with several linear and nonlinear existing methods, including the most recent ones. In one hand, Fisher-EM will be compared to the 9 already used clustering methods: EM with the Full-GMM, Diag-GMM, Sphe-GMM and Com-GMM models, Mixt-PPCA, Mclust (with its most adapted model for these data), PCA-EM, k-means and MCFA with a number of unobserved factors fixed to 3. On the other hand, the new Fisher-EM challengers will be k-means computed on the two first components of PCA (PCA–k-means), an heteroscedastic factor mixture analyzer (HMFA) method [136] and 3 discriminative versions of k-means: LDA–k-means [47], Dis–k-means and DisCluster (see [187] for more details). The comparison has been made on 7 different benchmark datasets coming mostly from the UCI machine learning repository:

- The **iris** dataset which is made of 3 different groups and described by 4 variables. This dataset has been described in detail in Section 5.1.
- The **wine** dataset is composed by 178 observations which are split up into 3 classes and characterized by 13 variables. This dataset has also been detailed in Section 5.4.
- The **chironomus** data contain 148 larvae which are split up into 3 species and described by 17 morphometric attributes. This dataset is described in detailed in [136].
- The **zoo** dataset includes 7 families of 101 animals characterized by 16 Boolean-valued attributes which stands for the color, if the animal is a predator or not, if it has a backbone, etc.
- The **glass** data are composed by 214 observations belonging to 6 different groups and described by 7 variables based on chemical properties. This dataset has been created to

Method	iris	wine	chiro	zoo	glass	satimage	usps358
$DLM_{[\Sigma_k\beta_k]}$	86.8±7.3†	97.8±0.0*	91.2±6.1	80.1±5.7	48.5±2.6	69.6±0.0*	81.1±5.4*†
$DLM_{[\Sigma_k\beta]}$	92.6±11	89.3±0.0	**98.2**±3.4	-	47.9±2.7	64.5±0.0	77.4±9.1
$DLM_{[\Sigma\beta_k]}$	80.5±3.4	93.8±1.1	94.7±4.2	72.6±5.3	49.4±2.9	65.7±1.3	73.7±7.4
$DLM_{[\Sigma\beta]}$	79.1±2.9	89.8±0.8	85.2±3.2	79.6±5.6	48.6±3.6	65.5±1.6	76.4±9.9
$DLM_{[\alpha_{kj}\beta_k]}$	87.8±0.5*	97.2±0.0†	85.0±1.4	71.8±6.6†	49.6±2.6†	**70.1**±0.0	**82.3**±4.7
$DLM_{[\alpha_{kj}\beta]}$	**97.8**±0.1	95.2±1.6	98.1±5.2	71.4±8.0	**51.1**±2.1*	61.7±0.2	73.2±9.5
$DLM_{[\alpha_k\beta_k]}$	92.8±2.1	**98.9**±0.0	85.5±14*†	71.8±6.9*	48.5±2.2	68.8±0.0	70.9±13.6
$DLM_{[\alpha_k\beta]}$	95.8±7.3	97.1±0.9	97.8±5.0	71.0±6.4	49.5±2.4	68.8±0.0	68.3±11.2
$DLM_{[\alpha_j\beta_k]}$	81.6±4.5	91.6±0.5	93.8±4.1	68.5±6.7	49.3±1.8	62.9±0.0†	76.1±11.0
$DLM_{[\alpha_j\beta]}$	73.6±6.7	89.8±0.9	89.7±4.1	79.1±4.9	47.4±1.2	67.6±2.8	77.4±10.7
$DLM_{[\alpha\beta_k]}$	80.1±6.9	91.4±3.2	89.3±1.9	70.1±6.5	48.9±1.3	68.7±1.9	80.5±6.0
$DLM_{[\alpha\beta]}$	66.8±0.0	89.5±1.0	89.2±5.7	**80.2**±5.3	47.0±1.7	62.1±0.0	69.9±14.2
Full-GMM	79.0±5.7	60.9±7.7	44.8±4.1	-	38.3±2.1	35.9±3.1	-
Com-GMM	57.6±18.3	61.0±14.9	51.9±10.9	59.9±10.3	38.3±3.1	26.1±1.5	38.2±1.1
Mixt-PPCA	89.1±4.2	63.1±7.9	56.3±4.5	50.9±6.5	37.0±2.3	40.6±4.7	53.1±9.6
Diag-GMM	93.5±1.3	94.6±2.8	92.1±4.2	70.9±12.3	39.1±2.4	60.8±5.2	45.9±9.1
Sphe-GMM	89.4±0.4	96.6±0.0	85.9±9.9	69.4±5.4	37.0±2.1	60.2±7.5	78.7±11.2
PCA-EM	66.9±9.9	64.4±5.7	66.1±4.0	61.9±6.2	39.0±1.7	56.2±4.2	67.6±11.2
k-means	88.7±4.0	95.9±4.0	92.9±6.0	68.0±7.4	41.3±2.8	66.6±4.1	74.9±13.9
MCFA	80.6±12.6	92.9±8.2	75.4±7.8	-	47.7±6.9	67.9±8.8	54.2±8.7
Mclust	96.7	97.1	97.9	65.3	41.6	58.7	55.5
Model name	*(VEV)*	*(VVI)*	*(EEE)*	*(EII)*	*(VEV)*	*(VVV)*	*(EEE)*

Table 5.8: Clustering accuracies and their standard deviations (in percentage) on 3 UCI datasets (iris, wine and chironomus) averaged on 25 trials. No standard deviation is reported for Mclust since its initialization procedure is deterministic and always provides the same initial partition. The signs † and ∗ indicates the model selection obtained by BIC and ICL respectively amongst the 12 DLM models.

Method	iris	wine	chironomus	zoo	glass	satimage	usps358
PCA–k-means [47]	88.7	70.2	-	79.2	47.2	-	-
LDA–k-means [47]	98.0	82.6	-	84.2	51.0	-	-
Dis–k-means [187]	-	-	-	-	-	65.1	-
DisCluster [187]	-	-	-	-	-	64.2	-
HMFA [136]	-	-	98.7	-	-	-	-

Table 5.9: Clustering accuracies (in percentage) on UCI datasets found in the literature (these results have been obtained with slightly different experimental setups).

criminological investigations since at the scene of the crime, the glass left can be used as evidence...

- The 4435 **satellite images** are split up into 6 classes and are described by 36 variables. The database consists of the multi-spectral values of pixels in 3×3 neighborhoods in a satellite image generating by NASA and proposed by the Australian Center for Remote Sensing.

- Finally, the last dataset is the **USPS data** which stands for 7291 handwritten digits from 0 to 9 scanned and stretched in a rectangular box 16×16 in a gray scale of 256 values from around 80 persons. Each pixel of each image was scaled into a Boolean (1/0) value using a fixed threshold. In this experiment, only the classes which are difficult to discriminate are considered. Consequently, this dataset consists of 1756 records (rows) and 256 attributes divided in 3 classes (numbers 3, 5 and 8).

Table 5.8 presents the average clustering accuracies and the associated standard deviations obtained for the 12 DLM models and for the methods already used in the previous experiments. The results for the 19 first methods of the table have been obtained by averaging 20 trials with random initializations except for Mclust which has its own deterministic initialization and this explains the lack of standard deviation for Mclust. Moreover, for each dataset, the DLM model corresponding to the BIC average has been marked by the sign † and by $*$ for the maximum ICL criterion. Besides, Table 5.9 provides the clustering accuracies found in the literature for the recent methods on the same datasets. It is important to notice that the results of Table 5.9 have been obtained in slightly different benchmarking situations. Moreover, missing values in Table 5.9 are due to non-convergence of the algorithms whereas missing values in Table 5.9 are due to the unavailability of the information for the concerned method. First of all, one can remark that Fisher-EM outperforms the other methods for most of the UCI datasets such as wine, iris, zoo, glass, satimage and usps358 datasets. However, the selection of the model type by BIC or ICL do not match all the time with the model having the best clustering accuracies. In particular, for the wine dataset, BIC and ICL select the $\text{DLM}_{[\Sigma_k \beta_k]}$ having a clustering accuracy reaching 97.8% whereas the best model in terms of clustering accuracy reaches 98.9% and is the $\text{DLM}_{[\alpha_k \beta_k]}$ model. Such a remark has been already observed by Fraley and Raftery in [59] and the model with the highest BIC do not guarantee to have the best clustering performance. Finally, it is interesting from a practical point of view to notice that some DLM models work well in most situations. In particular, the $\text{DLM}_{[.\beta]}$ models, in which the variance outside the discriminant subspace is common to all groups, provide very satisfying results for all the datasets considered here.

Chapter 6

Sparsity and discriminative variable selection

To deal with high-dimensional data, we proposed, in the previous chapters, a discriminative subspace clustering approach which both clusters and finds a low-dimensional discriminative subspace chosen such as it best discriminates the groups. This strategy is based on two key-assumptions: firstly, the latent space is linked to the observation space by a linear relationship, and secondly, the subspace orientation mainly defined by the between covariance matrix of the projected data is maximum. Moreover, conversely to most of subspace clustering approaches not assuming a common-subspace for the clusters, the visualization of the clustered data, in the case of Fisher-EM algorithm, is eased as it is always possible to project the data in the latent space. However, even though Fisher-EM presents good performances to discriminate and model clusters in high-dimensional spaces, a limitation still remains. Indeed, the latent space is defined by "latent variables" which are a linear combination of original variables.

The first consequence of this situation is the difficult interpretation of resulting clusters according to original variables. An intuitive way to avoid such a limitation is to keep only large loadings variables by thresholding loading absolute values beyond which they are constrained to be equal to 0. Even though this approach is very common, it has been particularly criticized by Cadima [29] since it induces some misleading information: a basic threshold can identify variables which are not the real important ones. This problem occurs in linear dimension reduction and particularly in the PCA context since principal components are also a linear combination of original variables. To that end, different approaches were proposed. In particular, some authors suggested to introduce sparsity in the loadings of principal components to ease the interpretation. Vines [172] for example, proposed an algorithm to produce simple approximate principal components directly from a variance–covariance matrix and loadings of original variables are restricted to values $-1, 1$ or 0. Other authors used the fact that PCA can be explained as a regression-type optimization problem. Indeed, the context of multiple regression allows to have accurate and sparse models by using an ℓ_1 penalty on standard least

square regression as proposed by Tibshirani [163] or on ridge regression as proposed by Zou and Hastie [194]. In particular, Zou *et al.* [193] have extended this approach by considering the PCA problem as a ridge regression and proposed a two-step algorithm to deal with such a task. More recently, Jenatton *et al.* [98] proposed a structured sparse approach for PCA.

In the literature, similar approaches treat about sparsity in the supervised context. For example, Trendafilov and Jolliffe [166], Clemmensen *et al.* [41], Qiao *et al.* [147] proposed various sparse approaches of FDA. More recently, Murphy *et al.* [137] on one side, and Maugis *et al.* [119], on the other side, developed a general framework for variable selection in model-based discriminant analysis and Witten and Tibshirani [179] proposed a penalized version of Fisher's linear discriminant analysis.

In addition to this interpretation problem, there still remains a limitation due to noise or non-informative variables. Of course, in the clustering context, it happens frequently that a large number of noise variables are in the set of original variables. However, since the latent variables are defined by a linear combination of the originals' ones, it implies that the noisy variables remain in the loadings of the projection matrix. Because of these noisy variables, the underlying structure of clusters can be masked and this may produce a deterioration of clustering results. To do so and since the 2000's, many authors were interested in introducing sparsity in the clustering task. In the literature, there are two main approaches which both select discriminative variables and cluster observations. In particular, there are researches of Law *et al.* [111], Raftery and Dean [148] and also an extended research by Maugis *et al.* [121] which recast the variables selection problem as a model selection one in an unsupervised context. A second approach consists in introducing directly sparsity in a penalized clustering criterion as did Pan and Shen [142], Wang and Zhou [173] for example, in the finite mixture model context, and also Witten and Tibshirani [178].

In first section of Chapter 6, we will treat the existing approaches dealing simultaneously with sparsity and clustering. Then, a second section will focus on the introduction of sparsity into the loadings of the projection matrix fitted in the Fisher-EM algorithm. To that end, 3 different approaches, based on ℓ_1 penalty terms, will be presented: the first one is based on a two-step approach, the second, on a regression criterion and finally, the third one uses a penalized svd criterion. Finally, the last section will focus on numerical experiments on simulated and real datasets which will provide interesting results in both regarding the simplicity of axis interpretation and the clustering accuracy.

6.1 State-of-the-art in variable selection for clustering

Since the clustering can provide a poor partition when original variables are noisy or irrelevant in regard to the clustering task, different approaches were proposed in the literature to deal with such a problem. All these approaches are based on the same assumption which supposes that the true underlying clusters differ only with respect to some of the original features.

Therefore, the goal of sparse clustering is to group the data on a subset of features. The main advantages of such a procedure remain in improving clustering results by removing noisy features, in facilitating both the identification of the clusters and their interpretation.

In the literature, this task which both selects variables and clusters the data, is apprehended differently according to the authors and can be divided into two different approaches: on the one hand, some authors tackle the problem of variable selection for model-based clustering within a Bayesian framework. We can cite in particular, the works of Liu *et al.* [113], Law *et al.* [111], Raftery and Dean [148] or Maugis *et al.* [120]. On the other hand, some authors favor an approach through a penalized clustering criterion. In this case, they introduce a penalty term in a clustering criterion either in the log-likelihood function as Galimberti *et al.* [63] or Xie and Pan [142] for example, or in a penalized criterion in a more general clustering framework, as Witten and Tibshirani [178].

This section details both approaches, particularly through the works of Raftery and Dean [148] or Maugis *et al.* [120] in which they recast the variable selection as a model selection problem for the model-based clustering context and those of Witten and Tibshirani [178] for the penalized clustering criterion.

6.1.1 Variable selection recasted as a model selection problem

The underlying idea of the works of Law *et al.* [111], Raftery and Dean [148] and Maugis *et al.* [120], is to find the variables which are relevant for the clustering task. The determination of the role of each variable is apprehended by the authors ([148, 120]) as a model selection problem. Their approach is developed in the GMM context. In particular, Raftery and Dean and Maugis *et al.* consider a collection of parsimonious and interpretable models, developed by Banfield and Raftery [9] and Celeux and Govaert [36], based on a specific decomposition of the mixture component variance matrix (see Section 2.1.3 for more details).

In the Raftery and Dean's approach, the authors define two different sets of variables: $\mathcal{S}$ which denotes the set of relevant variables and $\mathcal{S}_c$ which is the set containing the irrelevant variables. An interesting aspect of their approach is that they do not assume that the irrelevant variables are independent of the clustering variables conversely to Law *et al.* [111]. In particular, they define the irrelevant variables as those which are independent of the clustering but which remain all dependent of the set of relevant variables according to a linear relationship. The models in competition are compared with the integrated log-likelihood *via* a BIC approximation. Thus, the selected model maximizes the following quantity:

$$\left(\hat{K}, \hat{m}, \hat{\mathcal{S}}\right) = \arg \max_{(K,m,r,\ell,V)} \left\{ \mathrm{BIC}_{\mathrm{clust}}(\mathbf{y}^{\mathcal{S}}|\mathrm{K}, \mathrm{m}) + \mathrm{BIC}_{\mathrm{reg}}(\mathbf{y}^{\mathcal{S}_c}|\mathbf{y}^{\mathcal{S}}) \right\}, \tag{6.1.1}$$

where K is the number of clusters and $m \in \mathcal{M}$ is a model which belongs to the family of parsimonious models available in the Mclust [58] software. Note that the quantity to be maximized, in expression (6.1.2), can be decomposed into two different parts: the first term

corresponds to the Gaussian mixture model of K components on the subset of relevant variables $\mathcal{S}$, whereas the second one is relative to the regression of irrelevant variables $\mathcal{S}_c$ on the set of all clustering variables $\mathcal{S}$. However, the dependence assumption which defines the irrelevant set of variables according to all the relevant ones remains debatable. Indeed, on the one hand, by considering only the case where the irrelevant variables are independent on both the clustering and the relevant partition as it was considered in the works of Law *et al.* [111] seems to be unrealistic. On the other hand, considering that all the irrelevant variables depends on the relevant variables by a linear equation, seems to be also a very strong hypothesis which is not valid in certain practical cases. An other limitation of the Raftery and Dean's procedure is linked to their variable selection algorithm. Indeed, they propose a forward stepwise algorithm which considers only few variables at the beginning and it prevents from taking into account the block interactions between variables.

To overcome these limitations, Maugis *et al.* [120, 121] relaxe such restrictions and propose a more general variable role modeling. Indeed, in their approach, they define two subsets of variables: on the one hand, the relevant ones, which are grouped in $\mathcal{S}$ and on the other hand, its complementary $\mathcal{S}_c$, which is formed by the irrelevant variables. Maugis *et al.* consider, then, two types of behaviors among these irrelevant variables: these which can be explained by a linear regression from a subset $\mathcal{R}$ of the clustering variables and grouped in $\mathcal{U}$ and the ones which are totally independent of all the relevant variables ($\mathcal{W}$). Such a variable partition allows them to both consider the context developed by Law *et al.* [111] and also those defined by Raftery and Dean [148] and is referred to by the model collection SRUW. From this characterization, the authors also recast the variable selection problem into a model selection problem through an approximation of the integrated log-likelihood functions. Then the selected model satisfies:

$$\left(\hat{K}, \hat{m}, \hat{r}, \hat{\ell}, V\right) = \arg \max_{(K,m,r,\ell,V)} \left\{ \mathrm{BIC}_{\mathrm{clust}}(\mathbf{y}^{\mathcal{S}}|\mathrm{K}, \mathrm{m}) + \mathrm{BIC}_{\mathrm{reg}}(\mathbf{y}^{\mathcal{U}}|\mathrm{r}, \mathbf{y}^{\mathcal{R}}) + \mathrm{BIC}_{\mathrm{ind}}(\mathbf{y}^{\mathcal{W}}|\ell) \right\}, \tag{6.1.2}$$

where $V = (\mathcal{S}, \mathcal{R}, \mathcal{U}, \mathcal{W})$ stands for the variable partition. The first term of this expression, called $\mathrm{BIC}_{\mathrm{clust}}$, corresponds to the BIC criterion for a Gaussian mixture of K components on the relevant subset of variables $\mathcal{S}$. The model m belongs here to a collection of 28 parsimonious models which are available in the mixmod software [16] and include the GMM family introduced by Banfield and Raftery [9] and Celeux and Govaert [36]. The second term denoted by $\mathrm{BIC}_{\mathrm{reg}}$, is linked to the BIC criterion for a linear regression of the irrelevant variables $\mathcal{U}$ on a subset of clustering variables $\mathcal{R}$. Note that the index r stand for the structure of the covariance matrix which can be assumed to be spherical, diagonal or non-constraint. Finally, the last term depicts the BIC criterion for a Gaussian density on the variable subset $\mathcal{W}$ independent of the clustering variables. This Gaussian marginal distribution is characterized by a variance matrix σ which is constrained to be either diagonal or spherical and is specified by the index

ℓ in the expression (6.1.2).

The identifiability of the SRUW model is proved and also the consistency of their variable selection problem. Finally, they propose an algorithm based on a backward stepwise selection: it implies that all the variables are considered at the beginning of the procedure and only a block of variables is either included or excluded of the clustering relevant set of features. Such approach enables them to take into account variable block interactions if they exist. Then a second algorithm is executed to select both the model and the number of components of the mixture model.

6.1.2 Penalized log-likelihood

An other way which combines variable selection and clustering is the introduction of penalties in clustering criteria in order to yield sparsity in the features. This technique is used, in particular, in the GMM context. The main idea is to penalize the log-likelihood by introducing a penalty term in the finite mixture model. By assuming standardized n observations with mean 0 and variance 1 which are n realizations of a random vector $Y \in \mathbb{R}^p$ and by assuming a mixture of Gaussians, the penalized log-likelihood function is:

$$\mathcal{L}_{\mathrm{p}}(\theta) = \ell(\theta) - \mathrm{p}_\lambda(\theta) \tag{6.1.3}$$

where $\ell(\theta) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \phi(y_i; \theta_k)$ stands for the log-likelihood function, $\phi(.)$ is a Gaussian density function with parameters $\theta_k = \{m_k, S_k\}$ and $\{\pi_1, \ldots, \pi_K\}$ are the mixture proportions. The last term $\mathrm{p}_\lambda(\theta)$ is the penalty function.

In this context, Pan and Shen [142] propose a penalized log-likelihood criterion by assuming a Gaussian mixture model with common diagonal covariance matrices meaning that $\forall k \in \{1, \ldots, K\}$, $S_k = S = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_j^2, \ldots, \sigma_p^2)$ where $\sigma_j^2 \in \mathbb{R}$. The penalty function is focused on the means of K clusters ($m_{1k}, \ldots, m_{pk}$, $\forall k \in \{1, \ldots, K\}$) and has the following form:

$$\mathrm{p}_\lambda(\theta) = \lambda_1 \sum_{k=1}^{K} \sum_{j=1}^{p} |m_{kj}|\,, \tag{6.1.4}$$

where m_{kj} denotes the mean of the jth variable in the component k and λ_1 an hyperparameter which stands for the desired level of sparsity. Thus, since the observations are standardized, if the means of a variable j on each component are equal *i.e.* $m_{1j} = \cdots = m_{Kj} = 0$, then this variable is irrelevant and can be removed from the clustering variables. Therefore, a variable selection is realized when some m_{kj}'s can be shrunken toward 0. This situation occurs for an ℓ_1 penalty term large enough. In the same spirit, Wang and Zhou [173] propose two other penalty terms. The first one is based on ℓ_∞-norm:

$$\mathrm{p}_\lambda(\theta) = \lambda_\infty \sum_{j=1}^{p} \max_{k \in \{1,\ldots,K\}} \left(|m_{k1}|\right), \tag{6.1.5}$$

which has the advantage to incorporate group information. Thus, this penalty tends to shrink all the m_{kj}'s toward 0 as soon as the jth variable is non informative. However, such a penalty tends to shrink the m_{kj}'s in the same magnitude and thus does no take into account the situation where a variable is different from 0 on only one component. To that end, Wang and Zhou propose a second penalty function based on hierarchical penalties. These three penalized log-likelihood functions are developed with the restriction of the diagonal common covariance matrix of each cluster in the mixture model. Xie *et al.* [184] extend the model of Pan and Shen [142] by releaving the constraint on equal variance. Indeed, they propose an approach dealing with the case of cluster specific diagonal covariance matrices ($\forall k \in \{1, \ldots, K\}$, $S_k = \text{diag}(\sigma_{k1}^2, \ldots, \sigma_{kp}^2)$) which implies an additional term in the penalty function compared to equation 6.1.4:

$$\text{p}_\lambda(\theta) = \lambda_1 \sum_{k=1}^{K} \sum_{j=1}^{p} |m_{kj}| + \lambda_2 \sum_{k=1}^{K} \sum_{j=1}^{p} \left|\sigma_{kj}^2 - 1\right|. \tag{6.1.6}$$

In this case, a second regularized term is added and holds on the variance of the variable j of the kth component σ_{kj}^2 which can be shrunk towards 0. As previously, the hyperparameters λ_1 and λ_2 are selected through a modified BIC criterion, which takes into account the level of sparsity in the model complexity term in the BIC formula. Finally, Zhan *et al.* [192], more recently, propose a penalization in the case of Com-GMM model ($S_k = S$, $\forall k \in \{1, \ldots K\}$) in which they add the constraint $\lambda_2 \sum_{\ell=1}^{p} \sum_{j=1}^{p} |C_{j\ell}|$ in the penalty function, defined by Pan and Shen, in equation (6.1.4). The elements $\{C_{j\ell}\}_{j,\ell=1}^{p}$ belongs to the matrix $C = S^{-1}$ which defines the inverse covariance matrix. In the same work, Zhan *et al.* also propose an estimation procedure to deal with the $n < p$ case.

The introduction of a penalty term in the log-likelihood function is also used in the subspace clustering approaches. In particular, in the case of MFA models, Galimberti *et al.* [63] introduce an ℓ_1-penalty on the factor loadings in the log-likelihood function such as:

$$\text{p}_\lambda(\theta) = \lambda_2 \sum_{\ell=1}^{d} \sum_{j=1}^{p} |b_{\ell j}| \tag{6.1.7}$$

where $b_{\ell j}$ stands for the factor loadings. In a very recent work, Xie *et al.* [185] propose a penalized MFA approach from the model introduced by Gharamani and Hinton (see Chapter 2) where the covariance matrix of the noise term is diagonal and common to all factors. The penalty function has the following form:

$$\text{p}_\lambda(\theta) = \lambda_1 \sum_{k=1}^{K} \sum_{j=1}^{p} |m_{kj}| + \lambda_2 \sum_{k=1}^{K} \sum_{j=1}^{p} \|b_{kj}\|_2, \tag{6.1.8}$$

where b_{kj} stands for the factor loading of the kth factor. As in the previous approaches, the

first term based on the ℓ_1-norm is used to shrink the means m_{kj} to be exactly equal to 0 while the second term serves as a grouped variable penalty. Indeed, this last penalty aims to shrink the estimates of factor loadings b_{kj} which are close to 0 to be exactly equal to 0. Consequently, if a variable has a common mean equal to 0 and a common variance on each factor across the clusters and is independent with all other cluster such as $b_{kj} = 0\ \forall k$, then this variable is irrelevant and do not contribute in the clustering task.

6.1.3 Penalized clustering criterion

A general non-probabilistic framework for variable selection problem is recently proposed by Witten and Tibshirani [178]. They develop a framework for sparse clustering based on a general penalized criterion, which governs both variable selection and clustering.

In particular, they first propose a general clustering criterion based on a function f which has to be maximized:

$$\max_{\Theta} \sum_{j=1}^{p} f_j(Y_j, \Theta) \tag{6.1.9}$$

where $Y_j \in \mathbb{R}^n$ stands for the jth feature of the data amongst p variables and Θ a set of parameters. Many clustering methods can be reformulated from such a criterion. In particular, by considering f_j as the between cluster sum of squares for feature j, the well-known k-means method is obtained. The sparsity in the clustering task is introduced in the optimization problem defined in equation (6.1.9) as follows:

$$\max_{w,\Theta} \sum_{j=1}^{p} w_j f_j(x_j, \Theta) \quad \text{s.t.} \quad \|w\|^2 \leq 1,\ \|w\|_1 \leq \lambda \,\text{and}\, w_j \geq 0,\ \forall j, \tag{6.1.10}$$

where the threshold λ stands for the desired level of sparsity: a small value of this tuning parameter implies a high level of sparsity. Compared to the previous optimization criterion, Witten and Tibshirani consider functions f_j weighted by w_j on which ℓ_1 and ℓ_2 penalties are added. To optimize (6.1.10), the authors propose an iterative algorithm which first optimizes w holding Θ fixed and then, optimizes Θ given w. Even though this algorithm is not guaranteed to reach a global optimum, it is however guaranteed to increase the objective function at each iteration. Besides, as we can observe, the level of sparsity depends on the tuning parameter λ. To that end, Witten and Tibshirani propose a procedure based on the "gap statistic", proposed earlier by Tibshirani *et al.* [164], for estimating the number of clusters in a set of data. The technique uses the output of any clustering algorithm comparing the change in within-cluster dispersion with the one expected under an appropriate reference null distribution. Witten and Tibshirani apply their general criterion on two specific algorithms: the sparse k-means clustering and the sparse hierarchical clustering method.

6.2 Sparsity in the Fisher-EM algorithm

Since the DLM model is defined in a Gaussian mixture model context, it would be a natural way to introduce penalized terms in the log-likelihood function. For example, a penalized term based on the loadings of the projection matrix can be added in the log-likelihood function in order to introduce sparseness, as it has been already proposed by [63, 185] for the mixture of factor analyzers context. However, in the Fisher-EM algorithm, the projection matrix is not obtained by maximization of the log-likelihood function but by maximizing the Fisher's criterion in order to estimate a discriminative latent subspace. In the same way, the variable selection recasted as a model selection problem in regards to the works of [111, 148, 120, 121] has many advantages but do not seem appropriate to deal with our model. In particular, one of the advantage of the DLM model is to fit a discriminative latent subspace which could be very useful to extract the discriminative variables. However, by using the Bayesian framework, such information would be useless.

In this paragraph, we propose different approaches to introduce sparsity in the loadings. These three approaches are not based on the penalization of the log-likelihood function, but rather on a ℓ_1 penalty term added to the discriminative criterion. Consequently, we are going to focus on penalized Fisher's criteria whose the estimated procedures operate in the F-step of the Fisher-EM algorithm. To do that, we propose three different procedures. The first one presents a two-step approach by introducing sparsity once the projection matrix be fitted. The second procedure that we suggest, recasts the optimization problem of Fisher's criterion as a lasso regression-type problem in an unsupervised context. This method is based on the works of Qiao [147] who proposes sparse regularized FDA method for constructing sparse discrimination vectors in the supervised context. Finally, the last approach that we propose in this paragraph is based on the work of Witten and Tibshirani [180] on penalized matrix decomposition. After, a brief summary of their approach, we will expose how to use such a penalization in the modified Fisher's criterion developed in Chapter 4. This paragraph will end with a discussion about the choice of the tuning parameter for the sparse Fisher-EM and also about the different manners to implement the sparsity in the proposed algorithm.

6.2.1 Three sparse procedures

6.2.1.1 A two-step approach

This first approach divides the estimation of a sparse discriminative subspace into two steps: firstly, at iteration q, the traditional F-step of the Fisher-EM algorithm estimates the orientation matrix $U^{(q)}$ of the discriminative latent space conditionally to the posterior probabilities. Secondly, given this matrix $\hat{U}^{(q)}$, we look for introducing sparsity in its loadings according to an ℓ_1-penalty term.

Let us assume that the projection $\hat{U}^{(q)}$ has been estimated at iteration q, according to the traditional F-step. Then the following relation linking the latent and the observation spaces

is stated:

$$X^{(q)} = \hat{U}^{(q)t} Y.$$

This can also be reformulated from the row-coordinates of the matrix $X^{(q)} = \begin{bmatrix} x_1^{(q)} \\ \vdots \\ x_d^{(q)} \end{bmatrix} \in \mathbb{R}^{d \times n}$, denoted by $x_j^{(q)} \in \mathbb{R}^{1 \times n}$ with $j \in \{1, \ldots, d\}$, such that:

$$x_j^{(q)t} = Y^t \hat{u}_j^{(q)},$$

where $\hat{u}_j^{(q)}$ stands for the jth column vector of the projection matrix $\hat{U}^{(q)}$, estimated at iteration q, and $Y \in \mathbb{R}^{p \times n}$ denotes the original data matrix. As $\hat{U}^{(q)}$ is given, it is then possible to generate $x_j^{(q)}$ and to consider a regression of the row vectors $x_j^{(q)}$ on Y. In this case, $\hat{u}_j^{(q)}$ is solution of the following regression problem:

$$\hat{u}_j^{(q)} = \arg\min_{\beta_j} \left\| x_j^{(q)t} - Y^t \beta_j \right\|^2. \tag{6.2.1}$$

As $\hat{U}^{(q)}$ is solution of the least square regression of $X^{(q)}$ on Y, now, the main idea is to find a sparse solution through a least square regression. A common way, is to consider a penalized regression problem by introducing an ℓ_1-penalty term in the regression problem (6.2.1). Naturally, we look for:

$$\hat{\beta}_j^{(q)} = \arg\min_{\beta_j} \left\| x_j^{(q)t} - Y^t \beta_j \right\|^2 + \lambda_j \left| \beta_j \right|_1 \tag{6.2.2}$$

where $\hat{\beta}_j^{(q)}$ is a sparse approximation of the jth discriminative axis of the projection matrix $U^{(q)}$ and λ_j a constant term which denotes the level of sparseness. More generally, this problem can be reformulated as a multivariate penalized regression problem:

$$\hat{\beta}^{(q)} = \arg\min_{\beta} \left\| X^{(q)t} - Y^t \beta \right\|^2 + \sum_{j=1}^{d} \lambda_j \left| \beta_j \right|_1, \tag{6.2.3}$$

where $\hat{\beta}^{(q)} = [\hat{\beta}_1^{(q)}, \ldots, \hat{\beta}_d^{(q)}]$ denotes the sparse approximation of the projection matrix $\hat{U}^{(q)}$ estimated at iteration q. Note that different values of λ_j can be applied to penalize the loadings of different discriminative axes of U. However, in practice, we are going to consider a same level of sparseness ($\forall j \in \{1, \ldots, d\}$, $\lambda_j = \lambda$) for all discriminative axes.

There remains however an issue as the DLM model assumes that the projection matrix is constrained to be orthonormal *i.e.* $U^t U = \mathbf{I_d}$. This constraint is not taken into account in the penalized regression problem (6.2.3). We therefore propose to approximate the sparse matrix by considering a nearest orthogonal Procrustes problem [87], which can be formulated in our context, in the following manner:

Proposition 6.2.1. *Let $\hat{\beta}^{(q)}$ denotes the sparse approximation of the projection matrix $\hat{U}^{(q)}$, at iteration q, solution of the penalized least square regression defined in (6.2.3). Then, the orthogonal and sparse matrix $\tilde{U}^{(q)}$ which best-approximates $\hat{U}^{(q)}$ in terms of Frobenius norm and conditionally to the E-step, is solution of:*

$$\begin{aligned} \tilde{U}^{(q)} \quad &= \quad \arg\min_{\mathcal{U}} \left\| \hat{\beta}^{(q)} - \mathcal{U} \right\|_F \\ & \quad w.r.t. \quad \mathcal{U}^t\mathcal{U} = \mathbf{I_d}. \end{aligned}$$

In this case, by considering the singular value decomposition of $\hat{\beta}^{(q)} = u^{(q)}\Lambda^{(q)}v^{(q)t}$, then $\tilde{U}^{(q)} = u^{(q)}v^{(q)t}$ is sparse and orthonormal, and stands for the best approximation of $\hat{\beta}^{(q)}$.

Proof. At iteration q, in the F-step and conditionally to the E-step, the projection matrix $\hat{U}^{(q)}$ is estimated according to the maximization of the Fisher's criterion (see Chapter 4) which leads to project the observations in the discriminative latent space spanned by the column vectors of $\hat{U}^{(q)}$. We can therefore consider a penalized multivariate regression as in equation (6.2.3). The column vectors of $\hat{\beta}^{(q)}$ produce sparse discriminant vectors and approximate those of the projection matrix $\hat{U}^{(q)}$. However, we search the best approximation of the matrix $\hat{\beta}^{(q)}$ to an orthogonal matrix. Then, we consider the following problem:

$$\begin{aligned} \tilde{U}^{(q)} \quad &= \quad \arg\min_{\mathcal{U}} \left\| \hat{B}^{(q)} - \mathcal{U} \right\|_F \\ & \quad \text{w.r.t.} \quad \mathcal{U}^t\mathcal{U} = \mathbf{I_d}, \end{aligned}$$

where $\|.\|_F$ refers to the Frobenius norm. This problem is a nearest orthogonal Procrustes problem which can be solved by a singular value decomposition [66, 87]. The singular value decomposition of $\hat{\beta}^{(q)} = u^{(q)}\Lambda^{(q)}v^{(q)t}$ allows to write $\tilde{U}^{(q)} = u^{(q)}v^{(q)t}$, which allows us to conclude. □

The d first sparse and orthogonal discriminative axes can be obtained within the Fisher-EM algorithm through the following modified F-step: firstly, the projection matrix $\hat{U}^{(q)}$ is estimated according to the original procedure used in the F-step at iteration q. Then, the data are projected in this fitted latent subspace such as $X^{(q)} = Y\hat{U}^{(q)t}$. In order to obtain d sparse factors of $\hat{U}^{(q)}$, the adaptative lasso problem described below is solved iteratively, for $j \in \{1, \ldots, d\}$:

$$\hat{\beta}_j^{(q)} = \arg\min_{\beta_j} \left\| x_j^{(q)t} - Y^t\beta_j \right\|^2 + \lambda \left|\beta_j\right|_1 .$$

By denoting $\hat{\beta}^{(q)} = [\hat{\beta}_1^{(q)}, \ldots, \hat{\beta}_d^{(q)}]$, this matrix is decomposed through a singular value decomposition such as $\hat{\beta}^{(q)} = u^{(q)}\Lambda^{(q)}v^{(q)t}$ and finally the sparse and orthogonal projection matrix $\tilde{U}^{(q)} = u^{(q)}v^{(q)t}$ is obtained. This problem can be extended to a more general penalized regression by adding a ridge penalty term in order to handle the $n < p$ case and to provide a unique solution.

A limitation of such a procedure may be the deconnection between the estimation of the discriminative subspace and the introduction of the sparseness in the loadings of the projection matrix. To that end, two following approaches aim to propose penalized Fisher's criteria which fit directly sparse and discriminative latent subspaces.

6.2.1.2 Sparse Fisher criterion as a penalized regression-type problem

In the supervised context, Qiao *et al.* [147] propose a regularized linear discriminant analysis method in order to build sparse discriminant vectors. In particular, they firstly recast the optimization problem of Fisher's criterion as a regression-type problem (see Chapter 2 for more details) before adding a ℓ_1-penalty term to the objective function defined in equation (4.1.5). In this case, the penalized regression problem becomes then:

$$(\hat{A}, \hat{B}) = \arg\min_{A,B} \sum_{k=1}^{K} \left\| \left(R_W^t\right)^{-1} H_{B,k} - AB^t H_{B,k} \right\|_F^2 + \rho \sum_{j=1}^{d} \beta_j^t S_W \beta_j + \sum_{j=1}^{d} \lambda_{1,j} \left\|\beta_j\right\|_1 \quad (6.2.4)$$

$$\text{w.r.t. } A^t A = \mathbf{I_d},$$

where $B = [\beta_1, \ldots, \beta_d]$, $\hat{B} \in \mathbb{R}^{p \times d}$ is a sparse matrix which spans a linear subspace approximating the discriminant subspace such as:

$$\forall j \in \{1, \ldots, d\},\ \hat{\beta}_j = \arg\min_{\beta_j} \left(\left\| H_B^t S_W \alpha_j - H_B^t \beta_j \right\|_F^2 + \rho \beta_j^t S_W \beta_j + \lambda_1 \left\|\beta_j\right\|_1 \right), \quad (6.2.5)$$

with a fixed $A = [\alpha_1, \ldots, \alpha_d]$ and:

$$\hat{A} = EP, \quad (6.2.6)$$

with E the d first eigenvectors of $R_W^{-t} S_B R_W^{-1}$ and P, an arbitrary $d \times d$ orthogonal matrix. Besides, we recall that $\|.\|_F^2$ stands for the square Frobenius norm, $R_W \in \mathbb{R}^{p \times p}$ is an upper triangular matrix provided by the Cholesky decomposition of $H_W H_W^t = R_W^t R_W$ and the matrices H_B and H_W are defined as in equations (2.2.23) and (2.2.24).

However, two limits occur in the Qiao's work to be able to use such an approach in our context: on the one hand, the regression problem is mainly defined from the matrices H_W and H_B which are computed according to the class membership. Thus, their method can not be directly applied in our case, as the labels are unknown. In particular, in their approach, the matrix H_W needs to be centered from the class means and this can not be done in our unsupervised context. On the other hand, an additional problem occurs, as the DLM models assume that the column vectors of the projection matrix spanning the discriminative latent space are orthogonal. This constraint is not taken into account in the Qiao's work since the provided discriminative axes are sparse but non-orthogonal.

Consequently, to deal with the first problem, we propose to define the soft matrices $H_W^{(q)}$ and $H_B^{(q)}$ which are computed at each iteration and conditionally to the E-step, as following:

Definition 6.2.1. *The soft matrices $H_W^{(q)}$ and $H_B^{(q)}$ associated to the soft partition computed at iteration q in the E-step are:*

$$H_W^{(q)} = \frac{1}{\sqrt{n}}\left[Y - \sum_{k=1}^{K} t_{1k}^{(q)} m_k^{(q)}, \ldots, Y - \sum_{k=1}^{K} t_{nk}^{(q)} m_k^{(q)}\right] \in \mathbb{R}^{p\times n} \quad (6.2.7)$$

$$H_B^{(q)} = \frac{1}{\sqrt{n}}\left[\sqrt{n_1^{(q)}}(m_1^{(q)} - \bar{y}), \ldots, \sqrt{n_K^{(q)}}(m_K^{(q)} - \bar{y})\right] \in \mathbb{R}^{p\times K}, \quad (6.2.8)$$

where $t_{ik}^{(q)}$, for $i = 1, \ldots, n$ stand for the posterior probabilities computed in the E-step, $n_k^{(q)} = \sum_{i=1}^{n} t_{ik}^{(q)}$ and $m_k^{(q)} = \frac{1}{n}\sum_{i=1}^{n} t_{ik}^{(q)} y_i$ is the soft mean vector of the cluster k.

According to these definitions, both conditions $H_W^{(q)} H_W^{t(q)} = S_W^{(q)}$ and $H_B^{(q)} H_B^{t(q)} = S_B^{(q)}$ are still satisfied. In this case, the optimization problem defined conditionally to the E-step, at iteration q, becomes:

$$(\hat{A}^{(q)}, \hat{B}^{(q)}) = \arg\min_{A,B} \sum_{k=1}^{K} \left\| \left(R_W^{(q)t}\right)^{-1} H_{B,k}^{(q)} - AB^t H_{B,k}^{(q)} \right\|_F^2 + \rho \sum_{j=1}^{d} \beta_j^t S_W^{(q)} \beta_j + \sum_{j=1}^{d} \lambda_{1,j} \left\|\beta_j\right\|_1$$

$$\text{w.r.t.} \quad A^t A = \mathbf{I_d}, \quad (6.2.9)$$

and $\hat{B}^{(q)} = [\hat{\beta}_1^{(q)}, \ldots, \hat{\beta}_d^{(q)}]$ stands for the sparse projection matrix. However, such a problem does not take into account the orthogonality constraint of the projection matrix defined in the DLM model. This issue is tackled by approximating the sparse matrix $\hat{B}^{(q)}$ fitted by the optimization problem (6.2.9) with an orthogonal one according to a nearest Procrustes problem [66, 87]. In this case, the nearest Procrustes problem can be formulated as:

Proposition 6.2.2. *By considering $\hat{A}^{(q)}$ and $\hat{B}^{(q)}$ the solutions of the optimization problem (6.2.9), an orthogonal and sparse matrix $\hat{U}^{(q)}$ which best-approximates $U^{(q)}$ in terms of Frobenius norm, is solution of:*

$$\begin{aligned} \hat{U}^{(q)} &= \arg\min_{\mathcal{U}} \left\| \hat{B}^{(q)} - \mathcal{U} \right\|_F \\ &\textit{w.r.t.} \quad \mathcal{U}^t\mathcal{U} = \mathbf{I_d}, \end{aligned}$$

where $\|.\|_F$ refers to the Frobenius norm. By considering the svd of $\hat{B}^{(q)} = u^{(q)}\Lambda^{(q)}v^{(q)t}$, $\hat{U}^{(q)} = u^{(q)}v^{(q)t}$ is a sparse and orthonormal matrix and stands for the best approximation of $\hat{\beta}^{(q)}$.

Proof. At iteration q, in the F-step and conditionally to the E-step, the following optimization problem is considered:

$$(\hat{A}^{(q)}, \hat{B}^{(q)}) = \arg\min_{A,B} \sum_{k=1}^{K} \left\| \left(R_W^{(q)t}\right)^{-1} H_{B,k}^{(q)} - AB^t H_{B,k}^{(q)} \right\|_F^2 + \rho \sum_{j=1}^{d} \beta_j^t S_W^{(q)} \beta_j + \sum_{j=1}^{d} \lambda_1 \left\|\beta_j\right\|_1$$

$$\text{w.r.t.} \quad A^t A = \mathbf{I_d},$$

where $\hat{A}^{(q)}$ and $\hat{B}^{(q)}$ are defined in equations (6.2.6) and (6.2.5) respectively, and the column vectors of $\hat{B}^{(q)}$ produce sparse discriminant vectors as showed in Qiao *et al.* [147]. Moreover, as we search the best approximation of the matrix $\hat{B}^{(q)}$ to an orthogonal matrix, then, we reformulate the optimization problem as:

$$\begin{aligned} \hat{U}^{(q)} \quad &= \quad \arg\min_{\mathcal{U}} \left\| \hat{B}^{(q)} - \mathcal{U} \right\|_F \\ &\text{w.r.t.} \quad \mathcal{U}^t \mathcal{U} = \mathbf{I_d}, \end{aligned}$$

where $\|.\|_F$ refers to the Frobenius norm. This problem is a nearest orthogonal Procrustes problem which can be solved by a singular value decomposition [66, 87]. The singular value decomposition of $\hat{B}^{(q)} = u^{(q)} \Lambda^{(q)} v^{(q)t}$ allows to write $\hat{U}^{(q)} = u^{(q)} v^{(q)t}$, which allows us to conclude. □

With an algorithmic point of view, the optimization problem can be solved by alternatively optimizing over $A^{(q)}$ and then over $B^{(q)}$ as in the work of Qiao *et al.* [147]. Then, with fixed $A^{(q)} = [\alpha_1^{(q)}, \ldots, \alpha_d^{(q)}]$, the independent lasso problem is solved iteratively for $j \in \{1, \ldots, d\}$:

$$\hat{\beta}_j^{(q)} = \arg\min_{\beta_j} \left(\beta_j^t W^{(q)t} W^{(q)} \beta_j^t - 2 Y_j^{(q)} W^{(q)} \beta_j + \lambda_1 \left\|\beta_j\right\|_1 \right),$$

where $W^{(q)t} W^{(q)} = S_B^{(q)} + \rho S_W^{(q)}$ and $Y_j^{(q)} W^{(q)} = \alpha_j^{(q)t} \left(R_W^{(q)}\right)^{-1} S_B^{(q)t}$.

For a fixed $B^{(q)}$ *i.e.* $\hat{B}^{(q)} = (\hat{\beta}_1, \ldots, \hat{\beta}_d)$, we only need to compute the singular value decomposition of the quantity $R_W^{(q)-1}(H_B^{(q)} H_B^{(q)t}) \hat{B}^{(q)} = u^{(q)} \Lambda^{(q)} v^{(q)t}$ and the update of $\hat{A}^{(q)} = u^{(q)} v^{(q)t}$ follows. Both steps need to be computed several times until convergence.

Finally, in order to guarantee orthogonal column vectors of the fitted sparse discriminative matrix $\hat{B}^{(q)}$, we consider the svd of the matrix $\hat{B}^{(q)} = u'^{(q)} \Lambda'^{(q)} v'^{(q)t}$ and, to conclude, the best approximation of $\hat{B}^{(q)}$ is $\hat{U}^{(q)} = u'^{(q)} v'^{(q)t}$.

6.2.1.3 Sparse Fisher criterion with a PMD criterion

In Chapter 4, we proposed a modified Fisher's criterion based on the quantity $J(U) = \text{trace}\left(U^t (S^{-1} S_B^{(q)}) (S^{-1} S_B^{(q)})^t U\right)$. As a result, we showed that the projection matrix U which maximizes such a quantity, with respect to the orthogonality of its column vectors, consists of the right singular vectors of the svd of $S^{-1} S_B^{(q)}$. We would like then, to introduce sparsity in the loadings of the column vectors of U. Since we need to decompose by svd the matrix

$S^{-1}S_B^{(q)}$, then, we can use a result obtained by Witten and Tibshirani [180] who propose a penalized singular value decomposition, in order to provide interpretable factors.

From their work, we can formulate our goal in terms of penalized optimization problem. In the case of a rank 1 approximation of the matrix $S^{-1}S_B^{(q)}$ *i.e.* $d = 1$, the optimization problem to consider is:

$$\hat{u}_1^{(q)} = \arg\max_{u_1} u_1^t \left(S^{-1}S_B\right) v_1 \text{ s.t. } \|u_1\|_2^2 \leq 1, \|v_1\|_2^2 \leq 1, \sum_{j=1}^{p} |u_{1j}| \leq \gamma_1. \tag{6.2.10}$$

where $\hat{u}_1^{(q)}$ is the sparse approximation of the first column vector of the right singular vector $u^{(q)}$ of $S^{-1}S_B^{(q)}$ and $\gamma_1 > 0$ denotes the level of sparsity. Note that, in our case and conversely to Witten and Tibshirani proposing a more general case, no penalty term is associated to v_1 since we only need to introduce sparsity into the loadings of U. In order to obtain a sparse approximation of the $p \times d$ matrix U containing the right singular vectors ($d \geq 1$) of $S^{-1}S_B^{(q)}$, the optimization problem (6.2.10) is executed repeatedly. As $d > 1$, instead of using the $S^{-1}S_B^{(q)}$ matrix, we use the residuals obtained by subtracting from the $S^{-1}S_B^{(q)}$ the previous factors found. This leads to the following algorithm:

1. Let $M^1 = S^{-1}S_B^{(q)}$ and $d = \text{rank}\left(S^{-1}S_B\right)$.
2. For $j \in \{1, \ldots, d\}$:
   (a) Resolve $\hat{u}_j^{(q)} = \arg\max_{u_j} u_j^t M_j v_j$ w.r.t. $\|u_j\|_2^2 \leq 1$, $\|v_j\|_2^2 \leq 1$, $\sum_{\ell=1}^{p} \left|u_{j\ell}^{(q)}\right| \leq \gamma_1$.
   (b) Update $M^{j+1} = M^j - \lambda_j u_j^{(q)} v_j^t$.
3. $\hat{U}^{(q)} = [\hat{u}_1^{(q)}, \ldots, \hat{u}_d^{(q)}]$.

Certainly, the sparsity is taken into account in the loadings of the resulted matrix $\hat{U}^{(q)}$, nevertheless, the orthogonality constraint on its column vectors is not considered. Once again, this issue is tackled by approximating the sparse matrix $\hat{U}^{(q)}$ fitted by, repeatedly, the optimization problem (6.2.10) with an orthogonal one according to a nearest Procrustes problem [66, 87]. This can be formulated as:

Proposition 6.2.3. *By considering $\hat{U}^{(q)} = [\hat{u}_1^{(q)}, \ldots, \hat{u}_d^{(q)}]$ where $\hat{u}_j^{(q)}$ for $j \in \{1, \ldots, d\}$, $d = \text{rank}(S^{-1}S_B^{(q)})$, is solution of the optimization problem* (6.2.10) *repeated d times, then an orthogonal and sparse matrix $\tilde{U}^{(q)}$ which best approximates the projection matrix $\hat{U}^{(q)}$ in terms of Frobenius norm, is solution of:*

$$\begin{aligned} \tilde{U}^{(q)} \quad &= \quad \arg\min_{\mathcal{U}} \left\| \hat{U}^{(q)} - \mathcal{U} \right\|_F \\ &\textit{w.r.t.} \quad \mathcal{U}^t\mathcal{U} = \mathbf{I_d}, \end{aligned}$$

where $\|.\|_F$ *refers to the Frobenius norm. By considering the svd of* $\hat{U}^{(q)} = \tilde{u}^{(q)}\tilde{\Lambda}^{(q)}\tilde{v}^{(q)t}$, $\tilde{U}^{(q)} = \tilde{u}^{(q)}\tilde{v}^{(q)t}$ *is a sparse and orthonormal matrix and stands for the best approximation of* $\hat{U}^{(q)}$.

Proof. The proof is obvious by considering the definition of the nearest Procrustes problem. □

6.2.2 Practical aspects

We propose to introduce sparseness in the Fisher-EM algorithm because this presents several practical aspects among which the ability to interpret the discriminative axes. However, two different questions occur: the first one is linked to the choice of the hyperparameter which determines the level of sparsity and the second one corresponds to the implementation strategy of the sparsity in the Fisher-EM algorithm. Both aspects are discussed in the following paragraphs.

6.2.2.1 Choice of the tuning parameter

The choice of the threshold λ is an important problem since the number of zeros on the d discriminative axes depends on the degree of sparsity. Admittedly the sparsity on the axes improves the interpretation of the clustered data, but regarding the discriminative power of the axes, it has to remain a reasonable value.

In the work of Zou *et al.* [193] based on sparse principal component analysis, the choice of the hyperparameter depends on the explanation of the variance approximated by the sparse principal components which has to be equivalent as the traditional case. In the sparse k-means method, developed by Witten and Tibshirani [178], the choice of the tuning parameter is based on a permutation method closely related to the gap statistic previously proposed by Tibshirani *et al.* [164] for estimating the number of components in standard k-means. In particular, the algorithm which selects the tuning parameter is based on independent permutations of the dataset within each feature. The main idea is to compute the difference between an objective function performed by sparse k-means on the standard datasets with a fixed parameter and the same objective function computed on the permuted datasets. Note that the objective function is defined as the difference between the dissimilarities between each pair of observations and the average dissimilarities within clusters. The value of the hyperparameter corresponding to the largest gap is selected.

In the GMM context, the BIC criterion is often used to choose the number of components of the mixture or the appropriate model. It seems then possible to use such a criterion to also determine an appropriate value for λ. However, it is not an easy task since the effect of the parameter λ is not clearly defined in the model complexity, until the first conjectures of Efron *et al.* [50] in the Lasso regression. In this particular case, the first results, obtained by Zou *et al.* [195], show that the number of non-zero coefficients is an unbiased estimate for the degrees of freedom and is asymptotically consistent. More recently, this result is extended to the $n < p$

case by Kachour *et al.* [102]. According to the result obtained by Zou *et al.*, several authors working on ℓ_1-penalized log-likelihood function in the GMM context ([142, 184, 63, 185]) adapt this result, to compute a modified BIC criterion. In particular, Pan and Shen [142] propose to compute the model complexity of the BIC in regards to the non-zero values. In their case, the BIC criterion takes the following form:

$$PenBIC = -2\log\left(f\left(y|m, K, \hat{\theta}\right)\right) + \gamma_e \log(n) \tag{6.2.11}$$

where f stands for the observed likelihood of the data y given the model m and $\gamma_e = (K-1) + p + (Kp - d_e)$ denotes the effective number of parameters to estimate in the considered model. The quantity $(K-1)$ stands for the number of mixing proportions to estimate and p denotes the number of diagonal terms in the covariance matrix since they assume a diagonal and common covariance matrix between the K clusters. Finally, the last terms $(Kp - d_e)$ stand for the number of non-zero means in the model with d_e the number of mean components equal to 0.

In our model, the sparse constraint is applied on the loadings of the projection matrix which implies that the effective number of parameters to estimate in the $\text{DLM}_{[\Sigma_k\beta_k]}$ model is:

$$\gamma_e = (K-1) + Kd + (d[p-(d+1)/2] - \mathbf{d_e}) + Kd(d+1)/2 + K$$

where the quantity $d[p-(d+1)/2] - \mathbf{d_e}$ stands for the number of non-zero loadings in the projection matrix. In the same manner, this effective number of parameters to estimate can be declined for the 11 other submodels of the DLM family.

6.2.2.2 Implementation of the sparsity in the Fisher-EM algorithm

The proposed algorithmic strategy for introducing sparsity into the F-step can be incorporated in the Fisher-EM algorithm through two different ways.

Sparse Fisher-EM the usual F-step of the Fisher-EM algorithm can be replaced by the different sparse F-steps developed previously. The resulting algorithm, called hereafter sparse Fisher-EM, sparsifies at each iteration the projection matrix U before estimating the model parameters. As the first iteration, the projection matrix is sparse. This can lead to some drawbacks since an early introduction in the Fisher-EM algorithm of the ℓ_1 penalty could too much penalize the loadings of the projection matrix. In particular, the Fisher-EM algorithm is based on the EM one meaning that the final solution remains dependent of the starting point. Moreover, the introduction of sparsness adds constraints on the estimation of the latent subspace and this increases the dependency of the Fisher-EM solution to the initial conditions. However, it occurs, in practice that the solutions provided by the sparse Fisher-EM algorithms remain stable.

Regarding the computational complexity of the proposed algorithm, the sparse Fisher-

EM algorithm is quite efficient as its computational complexity does not depend on n. The estimation procedure of sparse loadings is iterative and the projection matrix is a $p \times d$ matrix which implies that the higher is the dimension of the observation space, the higher is the number of parameters to estimate. However, its computational complexity remains much smaller comparing to the approaches proposed by Raftery and Dean [148] or Maugis *et al.* [121].

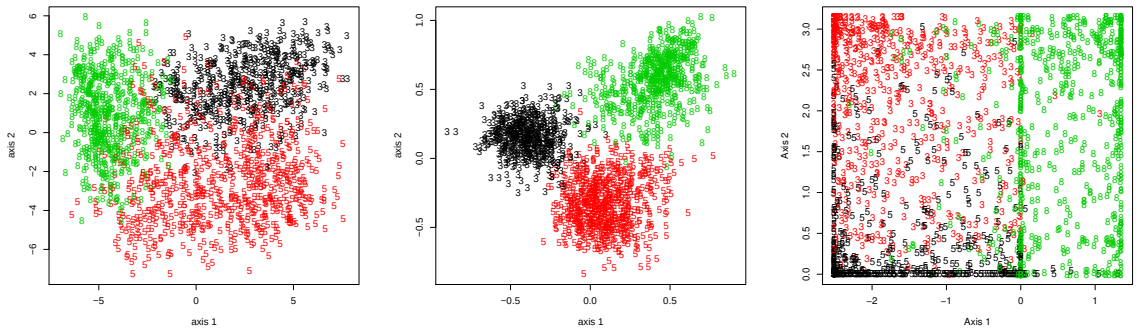
Traditional Fisher-EM with a sparse-step It is also possible to first run the traditional Fisher-EM algorithm and once it has converged, to sparsify the estimated loadings using the modified F-step. The main asset of the first approach is to keep the clustering performance of the Fisher-EM algorithm but to introduce sparsity into the loadings of the fitted projection matrix at the end. However, this approach presents some limitations when a considered dataset is described by many noisy variables. Indeed, since the latent variables fitted by the Fisher-EM algorithm are defined by a linear combination of the original variables, this implies that these noisy variables remain present in the loadings of the projection matrix. Therefore, the underlying structure of clusters can be partially masked because of this noise and it can imply a deterioration of clustering results. Thus, it is possible that running only one step of sparsity is not enough to remove all the noise variables in the loadings of the projection matrix. In practice, we suggest therefore to run the traditional Fisher-EM algorithm until convergence and then to iterate a dozen of iterations of the sparse Fisher-EM algorithm. This guarantees the efficiency of the standard Fisher-EM algorithm but also provides a set of discriminative variables.

6.3 Experiments and results

This paragraph presents experiments on simulated and real datasets in order to highlight the main features of the sparse procedures in the Fisher-EM algorithm. The first experiment aims to evaluate on the USPS datasets the differences which occur between the Fisher-EM algorithm and a sparse Fisher-EM algorithm. As we have introduced three procedures to deal with the sparsity in Section 6.2.1, the second experiment aims to compare them on the benchmark dataset already used in the previous chapter. The 3th experiment, on simulations, will focus to compare the performance of the sparse Fisher-EM algorithms and several existing approaches based on a Bayesian framework or on penalized clustering criteria. Finally, the last paragraph will focus on comparing, on real-world datasets, the efficiency of the sparse Fisher-EM algorithms with existing variable selection procedures.

6.3.1 Influence of the lasso penalty in the Fisher-EM algorithm

This first experiment focuses on both the visualization and interpretation sides of high-dimensional data analysis. In particular, it aims to illustrate the impact of sparsity in the discriminative variable selection and on the clustering accuracy. The sparse version of the



(a) PCA (b) Fisher-EM (c) Sparse Fisher-EM

Figure 6.1: Visualization of the USPS358 dataset with PCA (a), Fisher-EM (b) and sparse Fisher-EM (c). The class labels are only used for visualization purpose and have not been used by the studied algorithms when building the data projection.

algorithm enables in fact to select among the original variables the most discriminative ones and therefore considerably eases the interpretation of the data.

To illustrate these specific features, the Fisher-EM algorithm has been first applied and compared to PCA on the well-known USPS dataset. The dataset[1] is made of $7,291$ images divided in 10 classes corresponding to the digits from 0 to 9. Each digit is a 16×16 gray level image represented as a 256-dimensional vector. We first extracted a subset of the data ($n = 1,756$) corresponding to the digits 3, 5 and 8 which are usually difficult to discriminate. This smaller dataset is hereafter called USPS358. We applied the Fisher-EM algorithm on the USPS358 dataset, two different implementation of the intuitive approach (Fisher-EM + 1 sparsity step and sparse Fisher-EM) and PCA as a reference method.

Figure 6.1 first presents the obtained visualizations of the USPS358 data respectively with PCA and Fisher-EM. We recall that all the studied methods do not used the class labels when building the data projection and the true labels are used only for visualization purpose. It first appears that the projection provided by Fisher-EM is more informative than the one of PCA since the three digit groups are clearly separated. The introduction of sparsity in the loadings of Fisher-EM tends to push the groups in the corners of the window due to the use of the ℓ_1 penalty. Moreover, for this experiment, the level of sparsity has been fixed to 0.1 which implies that the loadings of the projection matrix have been constrained to be very sparse. The most useful effect of the ℓ_1 penalty is the selection of the discriminative original variables.

Figure 6.2 shows both the estimated means of the groups and the loadings obtained with Fisher-EM whereas Figures 6.3 and 6.4 stand for the loadings obtained by its sparse versions. Each estimated axis has been rebuilt as a 16×16 gray level image in order to visualize the discriminative pixels. In particular, a black pixel means that the absolute value of a loading

[1]This dataset can be found on the site of the university of Aachen: *http://www-i6.informatik.rwth-aachen.de/ keysers/usps.html.*

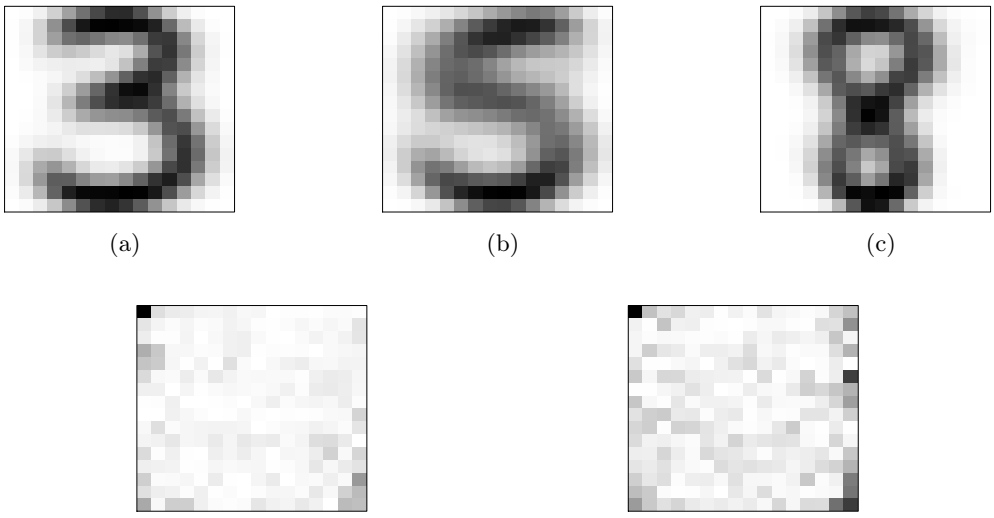
is maximum and equal to 1. Conversely, a white pixel represents a loading equals to 0. As we can see, the sparse versions of Fisher-EM succeed in selecting only a few original variables as discriminative. In particular, if one observes the loadings of the two sparse versions of Fisher-EM, it is possible to see that the darker pixels allow to discriminate the three digit groups. For instance, the darker pixels of the first loading allow to discriminate the digits 8 from the digits 3 and 5. Similarly, the dark pixel of the second loading clearly discriminate the digit 5 from the digits 3 and 8.

Finally, note that sparse Fisher-EM remains very efficient compared to the standard Fisher-EM or to the Fisher-EM algorithm + 1 sparsity step. In particular, in this experiment the sparse Fisher-EM algorithm has provided a clustering accuracy of 84.5% whereas the standard Fisher-EM algorithm and the Fisher-EM algorithm + 1 sparsity step have respectively reached 84.9% and 84.6% of correct classification rate. Consequently, despite the high level of sparsity, sparse Fisher-EM remains very competitive in terms of clustering accuracy compared to the standard Fisher-EM while facilitating the interpretation of the discriminative axes and provides in addition a selection of discriminative variables among the original variables.

6.3.2 Comparison between the 3 penalized procedures in the Fisher-EM algorithm

This experiment aims to compare the 3 sparse procedures in the F-step of the Fisher-EM algorithm developed in the last paragraph on the USPS358. From 25 random initializations, the 3 algorithms based on a two-step approach (sparseFEM-int), on a lasso regression approach (sparseFEM-reg) and on a penalized Fisher's criterion (sparseFEM-pen) have been run. For this experiment, the level of sparsity has been fixed to $\lambda = 0.1$. The mean of clustering accuracy computed from the true labels and on 25 replications are reported in Table 6.1 for the sparseFEM-int, sparseFEM-reg and sparseFEM-pen procedures. In addition, we can find in the same table the number (in average) of discriminative variables retained by each algorithm which corresponds to the number of non-zero variables. Moreover, the elapsed real time and the central processing unit (CPU) have been computed for each sparse F-step procedure (sparseFEM-int, sparseFEM-reg, sparseFEM-pen) and are also reported in Table 6.1.

First of all, we observe that, surprisingly, the two-step approach of sparse-FEM presents the best performances in terms of both clustering accuracy and level sparsity. Indeed, in average the sparseFEM-int procedure reaches a correct clustering rate of 82.69% whereas the sparseFEM-reg and sparseFEM-pen procedures reach respectively 81.42% and 80.62%. In addition, with a fixed sparse parameter fixed to $\lambda = 0.1$, only 6 variables are sufficient to discriminate the numbers $3, 5$ and 8 in the case of the sparseFEM-int algorithm whereas this number is much more important for sparseFEM-reg and sparseFEM-pen which need respectively 10 or 16 variables in average. The performances of these algorithms are really linked to the discriminative variable selection process. Indeed, let us consider Figures 6.5 and 6.6 which stand for respectively the loadings of the first and respectively the second



(a) (b) (c)

(d) Fisher-EM (axis 1). (e) Fisher-EM (axis 2).

Figure 6.2: Group means (top) and heat map of loadings (bottom) obtained with the Fisher-EM algorithm (84.9% of clustering accuracy): a black pixel means an absolute value of loading equal to 1 and a white pixel supposes an absolute value equal to 0.

(a) SparseFEM+1 sparsity step (axis 1)

(b) sparseFEM+1 sparsity step (axis 2)

Figure 6.3: Loadings of the projection matrix obtained with the Fisher-EM algorithm + 1 sparsity step (83.3% of clustering accuracy)

(a) SparseFEM-int (axis 1)

(b) SparseFEM-int (axis 2)

Figure 6.4: Loadings of the projection matrix obtained with the Sparse Fisher-EM algorithm (83.1% of clustering accuracy)

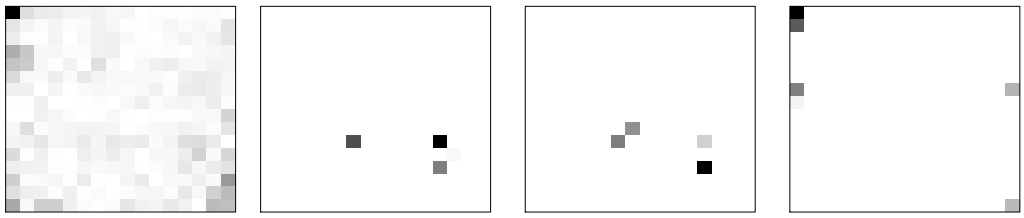
Approaches:	Clustering accuracy	Non-zero variables	Elapsed real time	CPU time
Fisher-EM	82.3 ± 4.7	256	158.7 ± 3.4	1.83 ± 0.35
sparseFEM-int	82.7 ± 6.8	5.6 ± 1.0	967.8 ± 1.1	3.15 ± 0.03
sparseFEM-reg	81.4 ± 6.8	16.0 ± 0.0	325.3 ± 1.1	2.36 ± 0.03
sparseFEM-pen	80.6 ± 8.1	10.1 ± 4.6	58.3 ± 2.6	1.52 ± 0.05

Table 6.1: Means of Clustering accuracies (in percentage) and their corresponding standard deviations computed for the 3 sparse procedures (intuitive (sparseFEM-int), regression (sparseFEM-reg) and penalized criterion (sparseFEM-pen)) on 25 replications. The elapsed real time and CPU time and their standard deviations have been also reported.

discriminative axis estimated by the traditional Fisher-EM (*i.e.* with no sparsity) and its sparse versions. These images correspond to a same random initialization for which the maximum clustering accuracy has been reached among the 25 random trials. In particular, the standard Fisher-EM algorithm has reached 84.9% of correct classification rates, the sparseFEM-int approach (84.5%), the sparseFEM-reg (84.4%) and the sparseFEM-pen (83.7%).

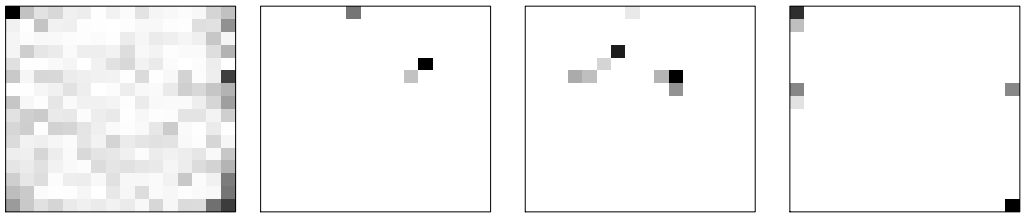
First of all, we can notice that Figures 6.5b, 6.5c and Figures 6.6b, 6.6c have in common subsets of discriminative loadings. Indeed, in the first axis (in Figures 6.5b and 6.5c) the three darker pixels which stand for the non-zero loadings estimated by sparseFEM-int are found also in the first axis estimated by sparseFEM-reg. Similarly, on the second axis, the two darker pixels in Figure 6.6b are also the discriminative ones in the case of sparseFEM-reg procedure in Figure 6.5c. The main difference between these both approaches is based on the number of discriminative loadings in each axis. Indeed, the intuitive approach turns out to be much sparser than the penalized regression approach and this can explain in a certain way the best performances in regard to the clustering efficiency of the intuitive procedure. Concerning the procedure based on the penalized Fisher's criterion (Figures 6.5d and 6.6d), its behavior is really different from the intuitive and the penalized regression procedures. Indeed, the darker pixels in both axes are the darker pixels obtained from the standard Fisher-EM algorithm in Figures 6.5a and 6.6a. In particular, the penalization introduced directly in the svd decomposition of the Fisher's criterion behaves as a thresholding on the loadings of the projection matrix. Such a behavior can present some drawbacks as it has been criticized in particular by Cadima [29]: a simple threshold can mis-identify the real important variables. This remark can thus explain the relative poor performances of the sparseFEM-pen procedure compared to sparseFEM-int and sparseFEM-reg and represents maybe the main disadvantage of this method.

However, even though the sparseFEM-pen procedure seems to be the least relevant in terms of interpretation, its clustering performances remain good. Moreover, this procedure appears to be much faster than the 2 others as it is illustrated in Table 6.1. Indeed, in the case of high-dimensional data ($n = 1,756$ and $p = 256$), sparseFEM-pen presents a low computational times conversely to the sparseFEM-int approach which needs twice more times



(a) Fisher-EM. (b) SparseFEM-int. (c) sparseFEM-reg. (d) SparseFEM-pen.

Figure 6.5: Loadings of the first discriminate axis obtained with (a) the Fisher-EM algorithm, (b) the sparse Fisher-EM algorithm according to the intuitive approach, (c) to the penalized regression approach, (d) to the penalized Fisher's criterion.



(a) Fisher-EM. (b) SparseFEM-int. (c) SparseFEM-reg. (d) SparseFEM-pen.

Figure 6.6: Loadings of the second discriminative axis obtained with (a) the Fisher-EM algorithm, (b) the sparse Fisher-EM algorithm according to the intuitive approach, (c) to the regression approach, (d) to penalized Fisher's criterion.

to be executed than sparseFEM-pen and 2/3 more times than sparseFEM-reg. This can be a limitation for the intuitive sparse procedure if we deal with very large datasets. However, such a behavior is common amongst the selection variable algorithms. Indeed, as comparison, we have computed the elapsed real time and the CPU time of the variable selection algorithms proposed by Witten and Tibshirani [178] on the one hand, and Raftery and Dean [148] on the other hand. It appears that in average, these both algorithms present a high computational cost since the elapsed real time reaches 927.1, respectively 7599.4, and the CPU time 5.81, respectively 14.99. This is explained by the fact that as the number of observations and the dimension of the space increase, the computational time of algorithms increases drastically too.

6.3.3 Comparison with existing approaches on simulated data

In this experiment, we aim to compare the performances of sparse Fisher-EM to several competitors already introduced. In particular, the 3 procedures of sparse Fisher-EM will be compared on simulations with the sparse k-means of Witten and Tibshirani [178], the Bayesian frameworks developed by Raftery and Dean (SU) on the one hand and Maugis *et al.* [121] (SRUW) on the other hand. For each method, we have used the implementation provided by the authors. More precisely, we have used the R package sparcl for the sparse k-means algo-

rithm, the clustvarsel package for the method of Raftery and Dean. For the SRUW algorithm, we have reported the clustering accuracies found in the literature [37].

This experiment aims to consider the same set-up as those proposed by Witten and Tibshirani in [178] and by Celeux *et al.* in [37]. Therefore, for this simulation, 3 Gaussian components of n observations each, which differ only on $q = 5$ features, have been simulated in a $p = 25$-dimensional observation space. In particular, each random vector Y_j conditionally to the class membership follows an univariate Gaussian density function with mean $\mu_{kj} = \mu \times (\mathbf{1}_{k=1,j\leq q}, -\mathbf{1}_{k=2,j\leq q})$ and a unit variance $\sigma_{kj} = 1$. Four different simulations have been run from the following situations: $q = 5$ and $p = 25$ and consist on varying the number of observations ($n = 30$ or $n = 300$) and the parameter μ which is equal to 0.6 or 1.7. Each simulation has been repeated 25 times.

The results are presented in Table 6.2 and stand for the average and the standard deviation of the clustering accuracy of the number of non-zero coefficients obtained from the 7 approaches. Note that the results about the SRUW algorithm corresponds to clustering errors and non-zero variable rates found in the literature [37]. We have added the results of the 3 procedures of sparse Fisher-EM (sparseFEM-int, sparseFEM-reg, sparseFEM-pen) already introduced in the last paragraphs which have been obtained in the same experimental conditions. Moreover, the reported results concerning the 3 sparse Fisher-EM algorithms stand for the $\text{DLM}_{[\Sigma_k\beta_k]}$ model with a sparsity level which corresponds to the highest BIC value obtained at each trial.

Concerning the first scenario which consists on $n = 10$ observations for each class and $\mu = 0.6$, the error rate of sparseFEM-int, sparseFEM-reg and sparseFEM-pen are comparable. Indeed, they reach 47% of error rate in average which is slightly larger than those obtained by the sparse k-means (40%) and SRUW (40%) algorithms but remain smaller that SU (62%). However, in the same manner than SRUW, the sparse Fisher-EM algorithms remain very sparse since they select a small number of discriminative variables (between 4 and 7 variables in average), in particular the penalized Fisher's criterion (sparseFEM-pen) procedure compared to sparse k-means or SU.

In the second scenario, we can observe that all the methods improve their clustering error rates which can be mostly explained by the fact that the centers of simulated cluster are farther as previously ($\mu = 1.7$). In particular the sparse k-means and SRUW algorithms have comparable performances in both clustering accuracies (8% of error rate) and sparsity: indeed, the number of non-zero variables has drastically decreased in the case of sparse k-means since it selects in average 8 non-zeros variables and this number decreases to 6.8 in the case of SRUW.

The two last scenarios deal with a larger number of observations fixed to $n = 300$. In the case of $\mu = 0.6$, we find the same behavior as in the $n = 30$, $\mu = 0.6$ case for the 7 algorithms: indeed, the clustering error rates remain high in average (0.4) and few sparseness is introduced except for the SRUW approach. However, we can note that the sparse Fisher-EM algorithms

Simulation	Method	Clustering error	non-zero variables
$n = 30$ $\mu = 0.6$	Kmeans	0.48 ± 0.05	25.0 ± 0.0
	SparseKmeans	0.47 ± 0.07	19.0 ± 6.6
	SU	0.62 ± 0.06	22.2 ± 1.2
	SRUW	$0.40 \pm 0.03^*$	$8.1 \pm 1.9^*$
	sparseFEM-int	0.47 ± 0.06	5.1 ± 7.1
	sparseFEM-reg	0.49 ± 0.07	11.1 ± 11.0
	sparseFEM-pen	0.47 ± 0.03	3.9 ± 2.5
$n = 30$ $\mu = 1.7$	Kmeans	0.14 ± 10.2	25.0 ± 0.0
	SparseKmeans	0.08 ± 0.06	23.6 ± 0.8
	SU	0.41 ± 0.10	16.6 ± 10.4
	SRUW	$0.08 \pm 0.08^*$	$6.8 \pm 1.4^*$
	sparseFEM-int	0.13 ± 0.11	2.5 ± 0.7
	sparseFEM-reg	0.14 ± 0.10	5.1 ± 1.4
	sparseFEM-pen	0.17 ± 0.11	2.0 ± 0.0
$n = 300$ $\mu = 0.6$	Kmeans	0.43 ± 0.03	25.0 ± 0.0
	SparseKmeans	0.46 ± 0.03	24.0 ± 0.5
	SU	0.42 ± 0.03	25.0 ± 0.0
	SRUW	$0.34 \pm 0.02^*$	$7.0 \pm 1.7^*$
	sparseFEM-int	0.41 ± 0.04	3.0 ± 1.6
	sparseFEM-reg	0.42 ± 0.03	4.0 ± 1.6
	sparseFEM-pen	0.42 ± 0.03	2.0 ± 0.0
$n = 300$ $\mu = 1.7$	Kmeans	0.05 ± 0.06	25.0 ± 0.0
	SparseKmeans	0.05 ± 0.01	15.0 ± 0.0
	SU	0.05 ± 0.01	25.0 ± 2.0
	SRUW	$0.05 \pm 0.01^*$	$5.6 \pm 0.9^*$
	sparseFEM-int	0.04 ± 0.01	3.0 ± 0.2
	sparseFEM-reg	0.05 ± 0.01	7.0 ± 1.7
	sparseFEM-pen	0.04 ± 0.01	2.0 ± 0.0

* results which have been reported from [37].

Table 6.2: Results obtained from 20 simulations with $p = 25$ and $q = 5$

improve their clustering accuracy compared to the similar case with $n = 30$ whereas the 3 other procedures do not. Moreover, in the case of $\mu = 1.7$, it appears that the 3 sparse Fisher-EM algorithms present very good performances. In particular, the sparseFEM-int and sparseFEM-pen procedures have the best clustering accuracies amongst the 7 approaches and in the same times they select 3 discriminative variables amongst the 25.

To conclude, the 3 procedures of sparse Fisher-EM algorithm present comparable clustering accuracies than the existing approaches and they tend to be sparser than the sparse k-means or SU algorithms procedures according to the use of the penalized BIC criterion for model selection.

6.3.4 Comparison with real data set benchmark

This last experimental paragraph will focus on comparing on real-world datasets the efficiency of sparse Fisher-EM algorithms in terms of both clustering accuracies and discriminative variable selection with several existing methods, including the most recent ones. On the one hand, the 3 sparse procedures of Fisher-EM will be compared between them. On the other hand, they will be compared to the sparse k-means introduced by Witten and Tibshirani [178] and also the Bayesian approaches proposed by Raftery and Dean [148] (SU) and extended by Maugis *et al.* [121] (SRUW). The comparison has been made on the 7 different benchmark datasets already presented in Chapter 5 and coming mostly from the UCI machine learning repository. Moreover, for each method, the number of components has been fixed to the true one and the other parameters such as the selection of covariance structures of the level of sparsity have been chosen through the corresponding methods of model selection. In particular, the penalized BIC criterion has been used for the 3 sparse Fisher-EM algorithms to select the model and the level of sparsity. For the three other approaches, we have used their own procedures of model and hyper-parameter selection.

Table 6.3 presents the average clustering accuracies and the associated standard deviations obtained for the 3 sparse procedures of the Fisher-EM algorithm (sparseFEM-int, sparseFEM-reg, sparseFEM-pen) and for the 3 methods already defined: sparse k-means, SU and SRUW. The level of sparsity obtained for each method has been reported through the average number of non-zero variables (in brackets in the table). The results associated to the sparse Fisher-EM algorithms have been obtained by averaging 20 trials with random initializations. SU, SRUW and sparse k-means methods have their own deterministic initialization and this explains the lack of standard deviation for both methods.

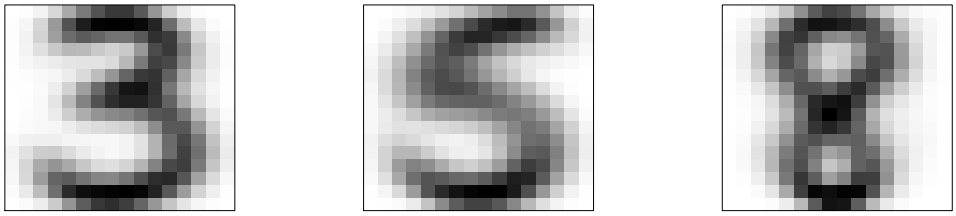
First of all, we can observe that the sparse Fisher-EM algorithms are competitive to existing methods in terms of good clustering performances and discriminative variables selection. More precisely, for a comparable level of sparsity, the 3 sparse Fisher-EM algorithms always outperform the SU algorithm on these datasets except for the zoo data. Moreover, the sparse Fisher-EM algorithms deal with both tasks of sparsity and clustering compared to the sparse k-means algorithm which does not really select variables. In particular, for the iris, wine

Approaches	iris *(p=4)*	wine *(p=13)*	chiro *(p=17)*	zoo *(p=16)*	glass *(p=9)*	satimage *(p=36)*	usps358 *(p=256)*
sparseFEM	96.0±0.0	97.8±0.2	78.2±11	63.0±9.7	51.4±1.3	69.6±0.6	79.3±5.4
-int	*(2.0±0.0)*	*(3.2±2.1)*	*(2.0±0.0)*	*(11±0.6)*	*(6.1±0.6)*	*(30.1±0.6)*	*(47±4.1)*
sparseFEM	88.9±1.4	98.3±0.0	84.1±10	75.4±1.9	51.6±0.9	60.6±3.0	78.8±9.1
-reg	*(4.0±0.0)*	*(3.0±0.0)*	*(2.8±0.79)*	*(13±4.5)*	*(6.0±1.6)*	*(31±1.2)*	*(82±16)*
sparseFEM	97.3±0.0	97.7±0.0	81.2±11	72.7±8.1	53.3±0.7	71.7±2.3	73.1±7.4
-pen	*(3.4±0.9)*	*(2.0±0.0)*	*(4.9±2.7)*	*(14.2±2.5)*	*(7.0±0.0)*	*(29±2.6)*	*(5.0±1.3)*
sparse	90.7	94.9	95.3	79.2	52.3	71.4	74.7
k-means	*(4.0)*	*(13.0)*	*(17.0)*	*(16.0)*	*(6.0)*	*(36.0)*	*(213)*
SU	96.0	92.7	71.1	75.2	48.6	58.7	48.3
	(3.0)	*(5.0)*	*(6.0)*	*(3.0)*	*(3.0)*	*(19.0)*	*(6.0)*
SRUW	96.0	94.4	92.6	92.1	43.0	56.4	36.7
	(3.0)	*(5.0)*	*(8.0)*	*(5.0)*	*(6.0)*	*(22.0)*	*(5.0)*

Table 6.3: Clustering accuracies and their standard deviations (in percentage) on 7 UCI datasets (iris, wine, chironomus, zoo, glass, satimage, usps358) averaged on 20 trials. The average number of nonzero variables is reported in brackets. No standard deviation is reported for SU/SRUW and sparse k-means since their initialization procedure is deterministic and always provides the same initial partition.

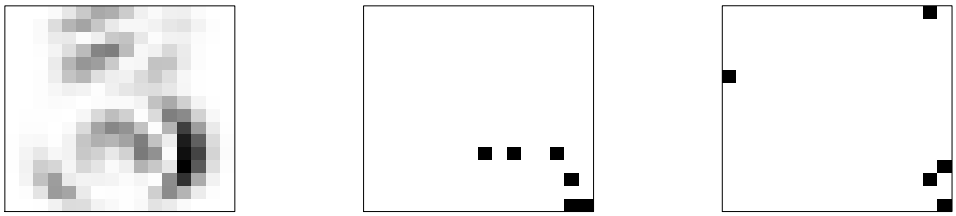
and chironomus datasets, the sparse Fisher-EM algorithms select in average between 2 and 4 variables on respectively 4, 13 and 17 original variables whereas they are all kept in the sparse k-means procedure. Even though the data are high-dimensional as in USPS358 data, the intrinsic procedure for sparse k-means which selects the level of sparsity does not provide a sufficient level of sparsity to make variable selection. In the SU situation, this method performs quite well for low-dimensional data but we find the limitations already observed by several authors such as [37, 120, 121, 178] in the case of high-dimensional data and particularly the high computational costs. This limitation also exists for the SRUW approach.

In terms of axes interpretation, let consider the sparse axes obtained by the 5 algorithms. In particular, Figures 6.8 and 6.9 stand for respectively the selected variables and the loadings of axes obtained by the Fisher-EM algorithm and its sparse versions, the sparse k-means and SU procedures, on the USPS358 dataset. These representations stand for a favorable case in terms of clustering accuracy which has been obtained among the 25 repetitions of this experiment: for the sparse k-means, the variable selection obtained in Figure 6.8a, stands for a situation with an error rate equal to 74.7%, Figures 6.8b and c represent respectively the relevant variables obtained by SU with 48.3% of error rates and SRUW (36.7%). For the 3 sparse Fisher-EM algorithms, we have superimposed in a same figure the loadings of the 2 discriminative axes fitted by sparseFEM-int, sparseFEM-reg and sparseFEM-pen procedures relative to (84.5%), (77.5%) and respectively (81.7%) of clustering error rate. First of all, we can observe that, even though the SU and SRUW procedures are sparser than the sparse k-means or the sparse Fisher-EM procedures, they correspond to a partition which is poor performing. This can be mainly explained by the fact that most of selected variables are irrelevant to discriminate



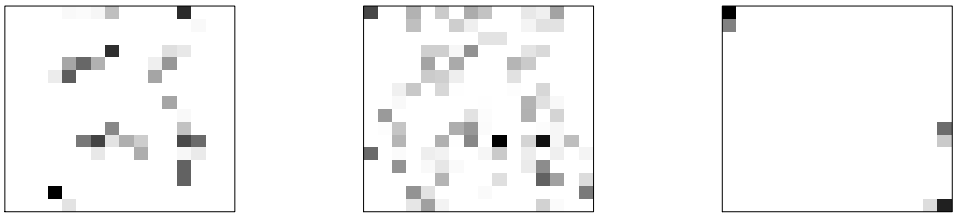
(a) (b) (c)

Figure 6.7: Group means obtained from the true labels in the USP358 datasets.



(a) Sparse k-means. (b) SU. (c) SRUW.

Figure 6.8: Clustering variable selection obtained from (a) the sparse k-means algorithm of Witten and Tibshirani, (b) the SU approach proposed by Raftery and Dean and (c) the SRUW approach from Maugis *et al.*.



(a) SparseFEM-int. (b) SparseFEM-reg. (c) SparseFEM-pen.

Figure 6.9: Clustering variable selection obtained from (a) the sparseFEM-int ($\lambda = 0.3$), (b) the sparseFEM-reg ($\lambda = 0.2$) and (c) the sparseFEM-pen procedures ($\lambda = 0.2$) of the sparse Fisher-EM algorithm with a level of sparsity selected by the penalized BIC.

Approaches:	CPU time	Approaches:	CPU time
sparseFEM-int	3.15 ± 0.03	sparse k-means	5.81
sparseFEM-reg	2.36 ± 0.03	SU	14.99
sparseFEM-pen	1.52 ± 0.05	SRUW	∞

Table 6.4: Computational times computed for the 3 versions of the sparse Fisher-EM, sparse k-means, SU and SRUW on the USPS358 data. No standard deviation is reported for SU/SRUW and sparse k-means as their initialization procedure is deterministic and always provides the same initial partition.

number 3 to numbers 5 and 8. We can observe, for example, in Figure 6.8b, that the black squares located in right bottom corner and selected as discriminant for the clustering task by the SU procedure, do not correspond to any discriminative pixel. Moreover, in Figures 6.8a and 6.8b, we can see that the sparse versions of Fisher-EM select only a few original variables as discriminative ones and correspond to a subset of those selected by the sparse k-means procedure. Besides, we can observe Figures 6.8c the same limitation of the sparse FEM-pen procedure as previously. Indeed, compared to sparse-FEM-int and sparse-FEM-reg, this last procedure only thresholds the loadings of the projection matrix of the Fisher-EM algorithm and do not consider, *a priori,* the task of discriminative variable selection.

Finally, both sparse-FEM-int and sparse-FEM-reg algorithms seem to answer quite well to the tasks of clustering and feature selection. Moreover, in regard to the existing approaches, such a procedure provides generally a sparser representation of the data in a reasonable time computing. Table 6.4 stands for an overview of CPU time averaged on 20 repetitions, on the USPS358 dataset. As we can remark, our procedures are much faster than SU and sparse k-means algorithms. Furthermore, as the SRUW procedure has not finished to run in a convenient time, no information about the computational time is reported. Consequently, the sparse Fisher-EM algorithms appear to be a good compromise, in practice, to cluster the data and select a set of discriminative variables in a reasonable time.

Part II

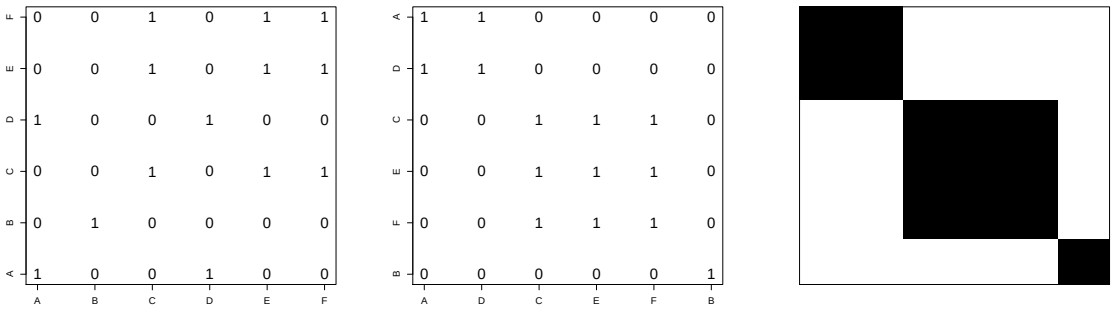
Seriation

Chapter 7

State-of-the-art in seriation

An important issue in datamining is determining and visualizing relational structures between observations. *Seriation* or *sequencing* is an old-fashioned tool for ordering elements and visualizing them. Its origins date back to the end of the 19th century where practical problems occurred in archeology and anthropology: in order to understand and reconstruct the past, found objects, potteries *etc*, are ordered and classified from few criteria such as the material, the symbols or their geographical location for example. Many empirical works used such an intuitive technique, but Sir W. M. Flinders Petrie [145], an Egyptologist proposed, the first, a systematic method of chronological sequencing for excavations of graves in the Nile area. More precisely, he crossed, in a table, the data linked to the geographical location of graves and those associated to the found objects. Then, he rearranged the table, by permuting its rows and its columns, such as the large values were close to the diagonal. It appeared that the similar graves were close to each other, in the rearranged table, and corresponded to a chronological order. This *sequence dating* influenced lots of archaeologists and anthropologists who improved and extended his methodology. However, Petrie did not comprehend the seriation problem through a mathematical background, and in the archeology context, it was necessary to wait 1951, so that Robinson [150] and Brainerd [24], in particular, define a rearranged table through mathematical properties. Many authors, such as [93, 39, 100, 31, 28] were interested to the particular geometry of the reordered matrix. Parallely, the seriation was defined, by some authors, ([122, 123, 140]) as an optimization-type problem based on an ordering criterion. Besides, since the main aim of seriation remains in the definition and the evaluation of the best permutation of a row, or a column, seriation was also comprehended as an algorithmic problem. This point of view was developed, in the literature, by many authors, such as in particular [42, 49, 151, 123, 6, 39, 27]. Even though this reordering method was widely applied and extended in archeology, as in the work of Ihm [95], many seriation approaches were developed, in parallel, in various fields, such as ecology, marketing or sociology for example. An historical overview of reordering methods was recently proposed by Liiv in [112].

In this Chapter, we first introduce the general framework of the seriation problem, before exposing the similarity criteria traditionally used in seriation and also the most well-known



(a) Non-ordered dissimilarity matrix. (b) Reordered dissimilarity matrix. (c) Image of reordered matrix.

Figure 7.1

algorithms. However, the traditional reordering technics present some limits as soon as the number of observations increases. We will expose therefore the alternative measures proposed in the literature which deal with this problem. Finally, we will end up this chapter by the most recent approaches used directly from the data matrix and named block-clustering approaches.

7.1 Seriation

7.1.1 A definition

Seriation is a data analysis tool whose aim is to work directly on data matrices. These matrices can be symmetric and in this case, the rows and columns refer to the same elements; more generally, they can be rectangular and then, the row elements are different from those in columns. Depending on the matrix considered, a specific taxonomy was proposed by Carroll and Arabie in [32]. In particular, the *one-way two mode* clustering refers to the seriation applied on a symmetric matrix (*i.e.* a matrix whose the elements stand for the similarity between pairs of observations) whereas on a rectangular matrix, it is called *two-way two mode* clustering. Many authors worked on this subject, and the reader can refer to [118, 131, 112] for a structured overview of these methods.

The principle of seriation is based on reordering rows and columns of a data matrix by successive permutations, such as the adjacent rows, respectively the adjacent columns, are the most similar. We can borrow the definition of seriation proposed by Liiv [112] which defines it as *"[...] an exploratory data analysis technique which reorders objects into a sequence along a one-dimensional continuum so that it best reveals regularity and patterning among the whole series"*. This situation is illustrated in Figures 7.1: let us consider a dissimilarity matrix, depicted in Figure 7.1a, which defines the similarity between pairs of observations, when its elements are equal to 1, and, at the opposite, their dissimilarity, when its elements are equal to 0. From this binary matrix, its rows, and symmetrically its columns, are permuted, such that

similar elements become pairwise adjacent, as it is illustrated in Figure 7.1b. We can therefore observe, in the same figure, that the reordered matrix forms groups in blocks. In order to highlight this block structure, the rearranged matrix can be considered as an image, as it is depicted in Figure 7.1c: the 1s are colored by black squares whereas the 0s are associated to white pixels. From this observation, we can feel that the seriation can be characterized as a local ordered clustering method. Indeed, conversely to traditional clustering approaches which cluster an observation with respect to its distance with the group means, seriation gives a local information, about similarities between adjacent pairs. Moreover, owing to the representation into blocks, as it is illustrated in Figure 7.1c, such an approach enables also to highlight a global structure of the data. Both remarks correspond to the main stakes of seriation: firstly, this method aims at rearranging rows and columns of a data matrix, in order to highlight a global structure in the data. Secondly, this method looks for identifying clusters amongst the rearranged elements of the matrix. An essential aspect in this method, is the definition of a dissimilarity criterion (or a similarity criterion) which measures the distance, or the closeness, between each pairs of individuals.

Let us consider $(x_1, \ldots, x_n)$ a dataset of n observations described by p variables. Let us also define its $n \times n$ corresponding dissimilarity matrix $D = (d_{ij})_{i,j \in (1,\ldots,n)}$ where each element d_{ij} stands for the dissimilarity between the observations i and j. Let us consider a permutation function Ψ which orders the elements of the matrix D according to an arrangement criterion $\mathcal{C}$. The goal of seriation is, then, to find the permutation function Ψ^* which optimizes $\mathcal{C}$, such as:

$$\Psi^* = \arg\max_{\Psi} \mathcal{C}\left(\Psi\left(D\right)\right). \tag{7.1.1}$$

Note that the dissimilarity measure can be either observed directly from the data, if we dispose of relational datasets for example, or it can be computed from a data matrix. A plethora of dissimilarity measures (*versus* similarity) was developed in the literature for the seriation problem and the next paragraph is going to expose some of them.

7.1.2 Similarity measures for seriation

In the literature, different kinds of measures were developed for the seriation procedure. More precisely, some authors focused on the geometrical properties of the rearranged matrix as Robinson [150], Brainerd [24] or more recently, Hubert *et al.* [93] and Chen [39] for example, whereas others developed similarity measures with regard to pairwises observations. Most of these works were resumed in [79].

7.1.2.1 Measures based on geometrical properties of the rearranged matrix

The interest on the form of the rearranged matrix was inspired by the work of Petrie [145], an Egyptologist, whose the aim was to order potteries according to the geographical context.

Design style	Context 3	6	7	5	1	2	4
beaker					×	×	
blackrim	×				×	×	
bottle	×					×	
flatpot		×		×			
handle	×			×			×
pointed		×	×	×			
spirals				×			×

(a) Non-ordered table

Design style	Context 1	2	3	4	5	6	7
beaker	×	×					
blackrim	×	×	×				
bottle		×	×				
handle			×	×	×		
spirals				×	×		
flatpot					×	×	
pointed					×	×	×

(b) Ordered table after permutations.

Table 7.1: Small example of a table containing design styles observed according to the geographical context and inspired by Petrie's serial ordering of Egyptian potteries.

Table 7.1 stand for a part of the well-known example introduced by Petrie and stand for different design styles found on potteries and their associated context numbered for 1 to 7. In the initial data matrix depicted in Table 7.1a, we can observe for example, that the context number 3 contains three different design styles: blackrim, bottle and handle, and the design style spirals are found in the contexts 4 and 5. By sorting simultaneously the rows and the columns of this table, such as the crosses are found as closed as possible to the diagonal, we obtain a reordered matrix as in Table 7.1a. If the cross symbols are replaced by 1s and the empty case by 0s, then the resulting matrix is called a Robinsonian matrix.

Such structure of the rearranged matrix was introduced, in parallel, by Robinson [150] and Brainerd [24]. In particular, they defined a *Robinson* matrix, from a similarity matrix, whose the highest entries within each row and column are on the diagonal of the matrix and the entries never increase when moving away from the diagonal. A similar definition was done for dissimilarity matrices which were referred to by *anti-Robinson* matrices, by Hubert *et al.* [92]. An anti-Robinson matrix is matrix which has the smallest entries within each row and column on the diagonal of the matrix and the entries never increase when moving away from the diagonal. Therefore, by considering the dissimilarity matrix $D = (d_{ij})_{i,j\in(1,\dots,n)}$, the elements d_{ij} of an anti-Robinson matrix has the following properties:

$$\begin{cases} \text{for} \quad 1 \leq i < j \leq n, \quad d_{i,j} \geq d_{i+1,j} \\ \text{for} \quad 1 \leq j < i < n, \quad d_{i,j} \leq d_{i+1,j}. \end{cases}$$

These constraints on the elements of the matrix D point out an ordering amongst the objects. This definition was widely exploited by Diday [46], who highlighted the importance of the Robinson, or anti-Robinson form, for representing a proximity matrix and its associated graphical display. Many applications exploited Robinsonian dissimilarities, in particular in archeology [150, 106], in psychology [91, 92], in the biological area through the analysis of DNA sequences [135] or in overlapping clustering [53, 46, 134].

From these properties, Caraux *et al.* [31] declined two different seriation criteria, based on

the dissimilarity matrix D:

$$\mathcal{C}_1(D) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}|i-j|^2 \tag{7.1.2}$$

$$\mathcal{C}_2(D) = \sum_{i=1}^{n}\sum_{j=1}^{n} (d_{ij} - \alpha|i-j|^2), \tag{7.1.3}$$

where d_{ij} denotes the dissimilarity between observations i and j. In the criterion $\mathcal{C}_1$, the distance $|i-j|^2$ reflects the distance to the main diagonal of D and the term d_{ij} is used as a weight with regard to this distance. This criterion increases when the highest dissimilarities are far away from the diagonal, leading to maximization. Conversely, $\mathcal{C}_2$ has to be minimized. This last criterion evaluates, indeed, the difference between pairs of dissimilarities and their rank difference ($|i-j|^2$) weighted by a parameter α. Consequently, the closer the observations i and j are, the smaller the dissimilarity d_{ij} and also the difference $(d_{ij} - \alpha|i-j|^2)$ are.

More recently, several authors, such as Hubert *et al.* [93], Chen [39] or also Brusco *et al.* [28] proposed different optimization criteria, which can be formulated through a general formulation:

$$\mathcal{C}_3(D) = \sum_{1\leqslant i<j<k\leqslant n} f(d_{ik}, d_{ij}) + \sum_{1\leqslant i<j<k\leqslant n} f(d_{kj}, d_{ij}),$$

where f is a function which takes different forms according to the authors. For example, if $f(.)$ stands for the sign(.) function defined such as:

$$f(x,y) = \text{sign}(y-x) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x < y \\ -1 & \text{if } x > y \end{cases},$$

then the criterion, proposed by Hubert *et al.* [93], is re-found:

$$\mathcal{C}_{3.1}(D) = \sum_{1\leqslant i<j<k\leqslant n} \text{sign}(d_{ij} - d_{ik}) + \sum_{1\leqslant i<j<k\leqslant n} \text{sign}(d_{ij} - d_{kj}). \tag{7.1.4}$$

This measure, taken back by Brusco *et al.* [28] more recently, quantifies the divergence of a rearranged matrix from a Robinsonian structure. It is based on a gradient index which evaluates the difference between the number of correct correspondences and differences with a Robinson structure. Moreover, by replacing $f(x,y)$ by $|x-y|\,\text{sign}(x-y)$, we refind the second function proposed by Hubert *et al.* [93] which has the following form:

$$\mathcal{C}_{3.2}(D) = \sum_{1\leqslant i<j<k\leqslant n} |d_{ij} - d_{ik}|\,\text{sign}(d_{ij} - d_{ik}) + \sum_{1\leqslant i<j<k\leqslant n} |d_{ij} - d_{kj}|\,\text{sign}(d_{ij} - d_{kj}). \tag{7.1.5}$$

This is a weighted version of the criterion defined in expression (7.1.4). Finally, by letting $f(x,y) = \mathbf{1}_{\{x-y\}}$ or $f(x,y) = |x-y|\,\mathbf{1}_{\{x-y\}}$, both formulations of Chen [39], named the violations anti-Robinson events, are refound.

Many improvements were made on the characterization of a Robinson matrix and we can cite, in particular, the work of Warrens and Heiser [176] who extended the 2-dimensional Robinson matrix to a Robinson cube.

7.1.2.2 Criteria based on a local neighborhood

McCormick *et al.* [122, 123] were the first authors who both formalized the seriation problem as an optimization problem and proposed, to that end, an iterative algorithm. Their main goal was to order and organize the data such as they can visualize, directly in a 2-dimensional table, the data structure. They proposed a similarity measure, well-known under the name "measure of effectiveness" (ME), based on the scalar product between rows and columns.

Let us consider the $n \times n$ dissimilarity matrix $D = (d_{ij})_{i,j\in\{1,\ldots,n\}}$. The measure of effectiveness is then defined as:

$$\mathcal{C}_4(D) = \frac{1}{2}\sum_{i,j=1}^{n} d_{ij}(d_{i,j-1} + d_{i,j+1} + d_{i-1,j} + d_{i+1,j}), \tag{7.1.6}$$

with $d_{0,j} = d_{n+1,j} = d_{i,0} = d_{i,n+1} = 0$. This criterion has to be maximized and when it reaches a maximum, the structure designed is compact in the sense that the rearranged matrix forms blocks. Please, note that, in the original work of McCormick *et al.* [122, 123], such a criterion was proposed in the general case of a $n \times m$ rectangular matrix which allowed to divide the optimization problem in two sub-problems: the first one, by maximizing the measure on the rows in order to find permutations on the columns of the matrix and the second one by considering the opposite situation *i.e.* the row permutations maximizing the measure on the columns. However, for sake of clarity and homogeneity, we rewrote here the criterion from the dissimilarity matrix D.

By remarking that the rearranged matrix had a block-diagonal form, some authors defined such a structure from the neighborhood. In particular, Niermann [140] proposed two criteria to minimize. The first one has the following form:

$$\mathcal{C}_5(D) = \sum_{i,j=2}^{n-1}\left[\sum_{k=i-1}^{i+1}\sum_{\ell=\max(i,j-1)}^{j+1}(d_{ij} - d_{k\ell})^2\right], \tag{7.1.7}$$

where the term in square brackets stands for a neighborhood comprising at most 8 adjacent entries and named the Moore neighborhood. The second criterion proposed, is such that:

$$\mathcal{C}_6(D) = \sum_{i,j=2}^{n-1}\left[\sum_{k=i-1}^{i+1}(d_{ij} - d_{kj})^2 + \sum_{\ell=\max(i,j-1)}^{j+1}(d_{ij} - d_{i\ell})^2\right]. \tag{7.1.8}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \Longrightarrow \quad \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

(a) Binary matrix with C1P. (b) All columns are C1P. (c) Binary matrix without C1P.

Table 7.2: (a) Binary matrix which has the consecutive ones property (C1P) since the permutation of one of its row enables to obtain a matrix (b) whose all columns have consecutive ones. The matrix (c) has not the C1P in columns since no permutation enables to obtain a matrix with consecutive ones in all columns.

It is based on the Neuman neighborhood (term in square brackets) which comprises, at most, 4 adjacent entries. The main difference between both criteria remains in the contributions of rows and columns in the neighborhood. In particular, in the Neuman neighborhood case, these contributions are independent from each others, contrary to the Moore neighborhood.

Finally, the last remark is based on the fact that, a dissimilarity matrix $D = (d_{ij})_{i,j\in\{1,\dots,n\}}$ can be viewed as a finite weighted graph, in which each observation would correspond to a vertex and each element d_{ij} would be relative to a weighted edge. Seriation can, therefore, be viewed as a path in the graph, where each vertex is visited once. This path is named in the literature a *Hamiltonian path*. In this case, the criterion to optimize is based on the minimization of the consecutive dissimilarities and takes the following form:

$$\mathcal{C}_7(D) = \sum_{i=1}^{n-1} d_{i,i+1}.$$

This notion can also be linked to the so-called *Traveling Salesman Problem* (TSP) whose a comprehensive overview can be found in [76, 5].

7.1.2.3 Seriation as consecutive ones property

The seriation problem can also be viewed as the construction of a matrix having the consecutive ones property (C1P). Indeed, by considering a binary matrix *i.e.* a $(0, 1)$-table whose entries are only 1 or 0, the consecutive ones property for columns (respectively for rows) is stated when there exist row permutations (respectively column permutations) such as the 1s are ordered in consecutive positions in all columns (or rows). This property is illustrated in Table 7.2 with the matrix (a) which has the consecutive ones property, as the permutation of its two first rows enables to obtain a matrix (b) whose all its columns have consecutive ones. In the same table, we have also illustrated the opposite case: the matrix (c) haven't got the consecutive ones property as no permutation of its rows allows to obtain consecutive ones in all columns.

Many authors have worked on the characterization of the consecutive ones property such as Kendall [105], Hubert [91], Meidanis *et al.* [132] or more recently Narayanaswamy and Subashini [139], and others, such as Booth and Lueker [20] or Hsu [90], worked on algorithms to

identify it. These two last works focused on the algorithmic of the consecutive ones properties and they will be detailed in the next section.

More recently, in the context of data compression, Johnson *et al.* [100] proposed to minimize a criterion based on a "run" which stands for the number of sequences containing consecutive ones on each row. Therefore, they aimed at minimizing the sum of runs on every rows of the matrix, leading to the following criterion:

$$\mathcal{C}_8(D) = \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n-1} |d_{i,j} - d_{i,j+1}|,$$

where the d_{ij}s are either equal to 1 or 0.
All the presented criteria aims to find a structure in the data matrix such as the similar elements are grouped together. In particular, in the case of symmetric $(0,1)$-tables, this implies that the similar elements are adjacent leading to a sequence of consecutive 1s, and are close to the diagonal. It seems to exist then a certain equivalence between the Robinsonian or the consecutive ones properties. However, such equivalence is not obvious as Warrens [175] showed it, in a very recent work. Indeed, he proved that firstly, such an implication depended on the dissimilarity measures and secondly, amongst them, only a few were concerned and required very strong conditions.

7.1.3 Reordering algorithms for seriation

Seriation is certainly linked to the similarity measures between rows and columns, but it also depends on algorithmic as the optimized permutation needs to be found. This is a *np*-hard problem which supposes the computation of all combinations between row and column permutations. Such a situation can be considered when the dataset is relatively small, but this task becomes difficult and cost computing as soon as the size of the dataset increases.

Different kinds of algorithms were proposed in the literature, aiming at both the efficiency and the speed of its execution. One of the most well-known algorithm and named *branch and bound*, suggests to make an exhaustive research on subsets of data, instead of on the entire dataset. Such methods were introduced in the 50's, by Croes *et al.* [42], Eastman *et al.*[49] and also Rossman *et al.*[151] and were improved more recently by Chen *et al.*[39] or Brusco *et al.*[27]. Other algorithms are based on heuristic research. In particular, the work of McCormick *et al.* [123] improved by Arabie and Hubert [6], used the neighborhood of each observation, in order to define a list of potential candidates which can be considered at each iteration. More precisely, the column (or row) selected at each iteration, is the one which allows the largest increase of their similarity measure (measure of efficiency). In the same perspective, Kirckpatrick *et al.* [108] introduced the concept of *simulated annealing paradigm* in the seriation problem. Such a procedure enables to accept a candidate with a certain probability which can be worse, than the current solution. Then, at the beginning, the probability to accept a candidate is high, but it decreases gradually with the execution of the algorithm. Most of

these methods took their origins in works on the Traveling salesman problem, and most of algorithms dealing with mixed-integer programming, branch-and-bound method, local-search algorithms, genetic algorithms and more, are resumed in [5], in [76] or in [79].

In the case of consecutive ones property, Booth and Lueker [20] introduced a data structure, called PQ-trees, that is able to compactly represent all valid permutations of the rows of a $(0,1)$-table providing the C1P. To build such a structure, they proposed a linear algorithm with respect to the number of rows, of columns and also to the total number of ones in the table, which tests the consecutive ones property in matrices. Such an approach was improved by the works of Hsu [90], who proposed a novel algorithm.

Seriation is an interesting datamining tool since it enables to both determine and visualize relational structures between observations. However, it presents some limitations as the traditional approaches are mainly based on permutations between rows and columns. When the number of observations increases, these combinatorial algorithms become cost-consuming and the associated solution can be therefore untractable.

7.2 Distances, neighbors and density-based connectivity

The notion of similarity between pairs of observations is not only associated to the seriation problem. More generally, this notion is very used in the nonparametric clustering task. In particular, some authors proposed new similarity measures, in order to deal with high-dimensional data. Before exposing these measures, we are going to briefly recall the traditional similarity measures.

7.2.1 Traditional similarity measures

As soon as clustering is defined as a data analysis tool aiming at grouping together data in homogeneous clusters, the least we can do, is to define the notion of homogeneous clusters. In the notion of homogeneous cluster or natural cluster, we consider the observations which are similar between them. The most intuitive and well-known measure of similarity is based on the distance. In particular, we can define the distance $\mathtt{d}$ between a pairwise of observations $(x_i, x_{i'})$ in a p-dimensional space, such as:

$$\mathtt{d}\left(x_i, x_{i'}\right) = \left(\sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^q\right)^{1/q}, \tag{7.2.1}$$

where $q \geq 1$. In particular, if $q = 2$, it gives the traditional Euclidean metric.

An other well-known similarity function ables to compare two vectors x_i and $x_{i'}$ is based

on the normalized inner product:

$$\mathtt{s}\left(x_i, x_{i'}\right) = \frac{x_i^t x_{i'}}{\|x_i\| \, \|x_{i'}\|} \tag{7.2.2}$$

and represents the cosine of the angle formed by the vectors x_i and $x_{i'}$. Note that in the case of binary vectors, such a function measures the features which are shared by both observations. Moreover, in the particular case of $(0,1)$-vectors, several authors proposed similarity measures based on the inner product $x_i^t x_{i'}$, such as the well-known Jaccard and Needham's measure:

$$\mathtt{s}\left(x_i, x_{i'}\right) = \frac{x_i^t x_{i'}}{x_i^t x_i + x_{i'}^t x_{i'} - x_i^t x_{i'}},$$

the Russel and Rao [155] measure:

$$\mathtt{s}\left(x_i, x_{i'}\right) = \frac{x_i^t x_{i'}}{n - x_i^t x_{i'}},$$

or also, the Dice measure:

$$\mathtt{s}\left(x_i, x_{i'}\right) = \frac{x_i^t x_{i'}}{x_i^t x_i - x_{i'}^t x_{i'}}.$$

There exist other measures based on the inner-product in the literature and certain can be found in [48].

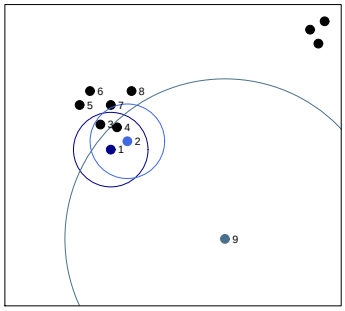
However, most of them become meaningless as soon as the dimension increases (see Chapter 1 for the curse of dimensionality). Several authors proposed indirect measures of neighborhood. They have the particularity to both introduce parsimony in the datasets in order to deal with high-dimensional data, and keep an information on the proximity between pairs of observations.

7.2.2 Similarity measures for high-dimensional data

In this paragraph, we are going to focus, in first, on the *share near neighbors* measure developed by Jarvis and Patrick [97], improved and extended by Guha *et al.* [74], Ertöz *et al.* [51] and Vathy-Fogarassy *et al.* [170]. The works of Gowda and Khrisna [71, 72] based on the mutual nearest neighborhood measure will be also explained in this section. Finally, certain authors defined and worked on the density based notion of clusters in order to develop simple and efficient practical algorithms, instead of working on the notion of similarity between pairwises of observations. We are going to briefly present the work of Ester *et al.* [52] in which the concept of density-connectivity was used, and its extensions [103].

7.2.2.1 The shared nearest neighbors

Jarvis and Patrick [97] introduced a similarity measure based on the g-nearest neighbors (g-nn) and named shared near neighbors. This similarity was stated on two key-ideas: firstly, the



(a) Representation of datapoints.

	1	2	3	4	5	6	7	8	9	...
1	4	4	3	3	·	·	·	·	·	·
2	4	4	3	3	·	·	·	·	·	·
3	3	3	4	·	·	·	·	·	·	·
4	3	3	·	4	·	·	2	·	·	·
5	·	·	·	·	4	4	3	·	·	·
6	·	·	·	·	4	4	3	·	·	·
7	·	·	·	2	3	3	4	2	·	·
8	·	·	·	·	·	·	2	4	·	·
9	·	·	·	·	·	·	·	·	4	·
...	·	·	·	·	·	·	·	·	·	·

(b) Measure of shared neighbors between pairwise of observations.

Figure 7.2: Datapoints in their 2-dimensional space with their 3-nearest neighbors (a) and the associated shared neighbors measure (b).

authors assumed that a pairwise of data points could be considered as similar if they shared a same neighborhood among their g-nearest neighbors. This supposes therefore that their respective list of g-nearest neighbors has to match. Secondly, these data points themselves have to belong to the list of the g-nearest neighbors of the other point. This condition aims at avoiding the association of isolated data points to a group. Indeed, since the similarity measure is based on the g-nearest neighbors, the volume of the sphere containing the g-nn changes according to the compactness of the data points. In particular, an isolated point can, in fact, share the same neighborhood as grouped points, as it is illustrated in Figure 7.2.

Let us rewriting these conditions in a formalism. Let us consider a dataset $\{x_1, \ldots, x_n\}$ of n observations described in a p-dimensional space having a metric. According this certain metric, let $\mathcal{S}_i$ denotes the subset of g-nearest neighbors of an observation i, for $i \in \{1, \ldots, n\}$. Then, the pairwises (x_i, x_j) with $i \neq j$ and $i, j \in \{1, \ldots, n\}$ are shared nearest neighbors if both conditions are satisfied:

$$\begin{cases} \mathcal{S}_i \cap \mathcal{S}_j \neq \phi \\ x_i \in \mathcal{S}_j \text{ and } \quad x_j \in \mathcal{S}_i \end{cases}.$$

In this case, the pairwise (x_i, x_j) is assumed to be shared near neighbors. Consequently, the associated shared near neighbor value is:

$$m_{ij} = |\mathcal{S}_i \cap \mathcal{S}_j|\,, \tag{7.2.3}$$

where $|\mathcal{A}|$ stands for the cardinality of the set $\mathcal{A}$. This measure is computed from the situation depicted in Figure 7.2a, for some pairs of observations and is illustrated in Figure 7.2b. As we can observe, the shared neighbor value introduces sparsity in the dissimilarity matrix and the noisy datapoint (observation 9) is removed. In terms of nearest neighbor graph, edges can be designed between pairs of shared nearest neighbors and, in the same way, a sparse graph is then provided. Jarvis and Patrick also proposed a weighted version of their similarity measure,

by taking into account the position of each observation in the neighborhood of its associated pairwise. In this case, the weight associated to the edge is:

$$\omega_{ij} = (g + 1 - \ell)\,(g + 1 - m) \tag{7.2.4}$$

where $\ell \in \{1, \ldots, k\}$ stands for the position of x_i in the g-nearest neighborhood of x_j, and similarly, m is the rank of x_j, in the g-nearest neighborhood of x_j. From the connected graph, the edges which are below on a certain threshold are removed and the remaining connected ones form clusters. The measure of shared near neighbors is then characterized of sparse. Indeed, in a first hand, it allows to remove outliers and noisy points because of the absence shared neighbors; in a second hand, such a measure enables to keep links in uniform regions, forming thus, the final clusters by removing the ones in the transition region. However, the main drawback of Jarvis and Patrick's method remains in the thresholding of the number of shared neighbors. Indeed, on the one hand, this threshold has to be high enough in order to prevent the merge of two distinct clusters. On the other hand, if it is too high, then, some little clusters can be removed from the clustering or one cluster could be divided in several sub-clusters. The last remark stands for the main drawback of this measure as it was already remarking by [51, 73].

This measure of similarity, denoted by m_{ij} in equation (7.2.3), was reintroduced later in the works of Guha *et al.* [74] through the following clustering function:

$$\mathcal{C}_7 = \sum_{k=1}^{K} n_k \sum_{i,j \in C_k} \frac{m_{ij}}{n_k^{1+2f(\theta)}},$$

where n_k is the number of observations in the cluster k and $f(\theta)$ is a function defined such as $n_k^{f(\theta)}$ stands for approximately the number of neighbors in the cluster k. The problem of such an approach is the existence of two parameters θ and $f(.)$ to calibrate and no information about it was done by the authors.

We can also cite the works of Ertöz *et al.* [51] who improved Jarvis and Patrick's approach and extended their notion of shared nearest neighbors measure. They introduced indeed the concept of "core" or "representation points" to the observations having the most of shared nearest neighbors. They considered, thus, the sum of links (m_{ij}) for each observation x_i:

$$M_i = \sum_{j=1}^{g} m_{ij},$$

and the points having a high M_i become candidate to be representative points, the ones having a low M_i according to a certain thresholds, are considered as noisy points. Consequently, their clustering algorithm introduced sparsity by eliminating noisy points and then, associated data points to representative points. Moreover, the authors showed that such a similarity measure remains efficient in the case of high-dimensional data. However, the calibration of the threshold

was not treated any more in their case, and this remains a major problem in practice.

Finally, Vathy-Fogarassy *et al.* [170] proposed more recently, a standardized version of the Jarvis and Patrick's measure and called it "fuzzy similarity measure" :

$$s_{ij} = \frac{|\mathcal{S}_i \cap \mathcal{S}_j|}{|\mathcal{S}_i \cup \mathcal{S}_j|}, \tag{7.2.5}$$

where s_{ij} stands for the standardized similarity measure between a pair of observations (x_i, x_j) and $\mathcal{S}_i$ stands for the subsets of nearest neighbors of an observation i. Besides, as each subset $\mathcal{S}$ consists of g-nearest neighbors and, according to equation (7.2.3), this similarity s_{ij} can be rewritten as:

$$s_{ij} = \frac{m_{ij}}{2g - m_{ij}}.$$

This standardized measure has the particularity to be comprised in the $[0, 1]$ interval and the more s_{ij} is close to 1, stronger is the similarity between observations i and j. At the opposite, $s_{ij} = 0$ means that the observations i and j are different from each other. From this standardization, the authors introduced the notion of "transitive fuzzy similarity measure" in order to evaluate the degree of spreading of nearest neighbors given an observation i. Moreover, they also introduced a parameter t, which determines the degree of the spreading. In this case, this transitive measure becomes:

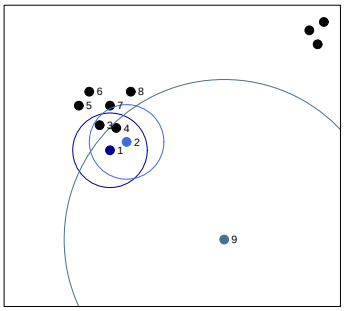
$$s_{ij}^{(t)} = \frac{\left|\mathcal{S}_i^{(t)} \cap \mathcal{S}_j^{(t)}\right|}{\left|\mathcal{S}_i^{(t)} \cup \mathcal{S}_j^{(t)}\right|}.$$

If $t = 1$, then the standardized shared nearest neighbor measure, defined in equation (7.2.5), is re-found. In the case of $t = 2$, the second degree of nearest neighbors is taken into account *i.e.* the nearest neighbors of the nearest neighbors of the observation i, and so one. These different effects are weighted and summed up, in order to have a global measure of similarity. However, as the previous approaches, the proposed algorithm needs to calibrate several hyper-parameters, such as the maximum degree of spreading, or the determination of the vector of weights, and Vathy-Fogarassy *et al.* did not propose any method to fix them.

The major limitation of all these approaches is the existence of hyper-parameters to calibrate. This remains a problem in practice, as the authors do not really discuss about it and do not propose convenient solutions.

7.2.2.2 The nearest neighbors

Gowda and Krishna [71, 72] proposed a measure of similarity between pairs of observations and based on the neighborhood. This measure, called mutual neighborhood, is defined as the sum of the mutual neighbor ranks between two observations. In other words, let us consider $\{x_1, \ldots, x_n\}$, n observations in a p-dimensional space endowed with a metric. If an



(a) Datapoints in their 2-dimensional space.

observations	NN1	NN2	NN3
1	2	4	3
2	4	1	2
3	4	7	1
4	2	3	7
5	6	3	7
6	5	7	3
7	4	3	6
8	7	4	2
9	2	1	4
...	...	...	...

(b) Nearest neighbors of observations 1-9.

	1	2	3	4	5	6	7	8	9	...
1	·	3	6	0	0	0	0	0	0	···
2	3	·	0	2	0	0	0	0	0	···
3	6	0	·	0	0	0	4	0	0	···
4	0	2	0	·	0	0	4	0	0	···
5	0	0	0	0	·	2	0	0	0	···
6	0	0	0	0	2	·	5	0	0	···
7	0	0	4	4	0	5	·	0	0	···
8	0	0	0	0	0	0	0	·	0	···
9	0	0	0	0	0	0	0	0	·	···
...	...	...	...	...	...	...	...	...	...	...

(c) Mutual neighbor value for each pairwise of datapoints.

Figure 7.3: (a) Representation of datapoints in their 2-dimensional space and (b) their associated mutual neighbor values by considering 3-nearest neighbors.

observation i is the ℓth nearest neighbor of observation j and j is the mth nearest neighbor of i, then the mutual neighbor value is $MNV(x_i, x_j) = m + \ell$. Consequently, smaller is the mutual neighbor value, the more the observations will be similar. In particular, if $m = 1$ and $\ell = 1$, $MNV(x_i, x_j) = 2$ and the observations x_i and x_j are the nearest neighbors. However, the computation of such a measure needs to order $n - 1$ observations given the studied one which becomes cost computing as soon as the dataset becomes large. To that end, Gowda and Krishna restricted the computation of such a measure to the g-nearest neighbors of each observation. In particular, let us consider $\mathcal{S}_i$, respectively $\mathcal{S}_j$, the subset of g-nearest neighbors of the observation i, respectively j, and let ℓ (respectively m) denotes the position of observation i (respectively j) amongst the g-nearest neighbor of j (respectively i), it leads then to the following conditions:

$$\begin{cases} x_i \in \mathcal{S}_j \text{ and } & \text{rank}\,(x_i) = \ell \\ x_j \in \mathcal{S}_i \text{ and } & \text{rank}\,(x_j) = m \end{cases},$$

and the mutual neighbor value is $MNV(x_i, x_j) = m + \ell$. If the observation i is not in the neighborhood of j, then, the pairwise (x_i, x_j) has no mutual neighbor and $MNV(x_i, x_j) = 0$. This last condition prevents the grouping of isolated datapoints, as in the Jarvis and Patrick's case (see Figure 7.3a). The computation of such a measure is depicted in Figures 7.3: Figure 7.3a stands for the representation of datapoints in their 2 dimensional-space, Figure 7.3b is an ordered list of 3-nn of each data point and Figure 7.3c stands for the associated mutual neighbor value of each pairwise of observations. In this particular example, the mutual neighbor value between observation 1 and 2 is equal to $2 + 1 = 3$, since the observation 2 is the first nearest neighbor of observation 1 and observation 1 is the 2nd nearest neighbor of observation 2 as we can observe in Figure 7.3b. Conversely, since the data point 8 does not share any neighborhood of other observations, its mutual neighbor values are 0 (see line 8 in

Figure 7.3c). This measure is very sparse and seems to introduce much more zeros than the one of Jarvis and Patrick.

Finally, the notion introduced by Gowda and Khrishna was recently extended by Zhang *et al.* [190] who proposed the estimation of a normalized density derivative from this mutual neighbor value.

7.2.3 Density based-clustering

This notion of similarity between pairwises of observations was exploited, in the identification of clusters, through the construction of neighborhood graphs and the development of simple and efficient clustering algorithms. There exist different kinds of families of neighborhood graphs, according to the similarity measure used. The most well-known families of graphs are, perhaps, the ε-neighborhood graphs and k-nearest neighbor graphs. In the first case, a pair of observations will be connected if their distance is smaller than ε whereas in the second case, an observation will be connected to its k-nearest neighbors.

The authors, who worked in such an approach, were focused on defining geometrical properties of a cluster. Whereas the works of Ester *et al.* [52], Ankerst *et al.* [4] or Kailing *et al.* [103] defined a density connectivity concept, in order to propose efficient clustering algorithms, other authors were focused on a more formal definition of clusters, as connected components of the t-level set of the underlying probability distribution.

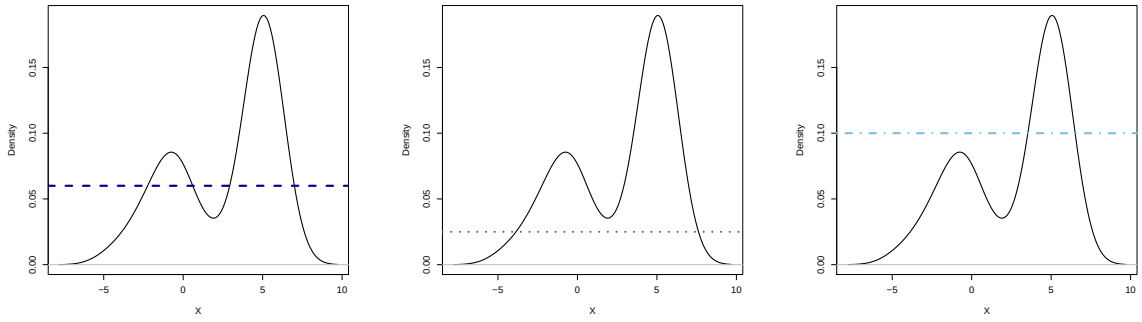
7.2.3.1 Algorithmic approaches for identifying clusters

Instead of focusing on the similarity measures between pairwise of observations, some authors tried to define a cluster through geometrical properties. To that end, the density-based notion of clusters was introduced, in order to develop simple and efficient algorithms of clustering.

In particular, Ester *et al.* [52] based their clustering algorithm on the notion of *density-connectivity* which provides a relational structure between observations. This notion is explained through two key-definitions. Firstly, by denoting $\mathcal{S}_i$ the ε-neighborhood of an observation i, a point j is defined as *directly-density reachable* from i if these two conditions are satisfied:

$$\begin{cases} \|x_j - x_i\| & \leq \varepsilon \\ |\mathcal{S}_i| \geq & \xi. \end{cases}$$

where $\|.\|$ stands for any norm, $|\mathcal{S}_i|$ is the cardinal of the subset $\mathcal{S}_i$ and ξ denotes a minimum number of elements, in the ε-neighborhood of the observation i. Please, note that the *directly-density reachable* notion can be viewed as a restricted ε-neighborhood as, in addition to consider the elements included in the ball centered in x_i with radius ε, this ball has to reach a minimum size. Secondly, an observation j is *density-connected* to a point i, if there is an intermediate point which is both density-reachable by j and i. Consequently, a cluster is defined as a set of points density-reachable and the number of clusters corresponds to the



(a) $t = 0.06$ (b) $t = 0.025$ (c) $t = 0.10$

Figure 7.4: Two clusters are defined as connected components of the t-level set of a density f.

number of connected components. From these both notions, Ester *et al.* proposed an algorithm, called DBSCAN, which was the basis of works such as, those of Kailing *et al.* [103] or Ankerst *et al.* [4].

We can remark that all these methods are intrinsically linked to connectivity graphs approaches. However, their main drawbacks are, perhaps, the lack of statistical justifications and also the absence of methodology for practical issues. Indeed, the definition of a density-connectivity supposes the use of a similarity measure which needs the calibration of at least one parameter. Moreover, it is a well-known fact, that the calibration of such a parameter is totally dependent of the determination of the true number of clusters. These previous works, however, mainly based on algorithmic, do not really provide a solution for the choice of these parameters in practice, nor theoretical justifications.

From a formal definition of a cluster by Hartigan [82], Bito *et al.* [25] , Biau *et al.* [13] and also Maier *et al.* [116] particularly, were focused in deriving theoretical justifications, in order to identify the number of clusters or to choose the kind of neighborhood for building an appropriate graph.

7.2.3.2 Theoretical works on the identification of a cluster

Hartigan [82] has introduced in 1975 a formal definition of a cluster.

Definition 7.2.1. *Given a random variable $X \in \mathbb{R}^p$ with probability f, let $t > 0$ denotes a fixed and positive level set, then the t-level set of the density f is defined as:*

$$\mathcal{L}(t) = \{x \in \mathbb{R}^p | f(x) \geq t\},$$

and we denote $k(t)$ the number of connected components associated to $\mathcal{L}(t)$.

Clusters are then defined as connected components of the t-level set of an underlying probability distribution. This has the main advantage to be easy to interpret geometrically.

Moreover, it enables also to identify the number of connected components, or clusters. This situation is illustrated in Figure 7.4, which stands for the distribution of a mixture of 2 Gaussian densities and different level in a 1-dimensional space and different level sets t. As we can observe, the main problem of such a definition remains in the determination of the level t. Indeed, as we can observe in Figure 7.4, t needs to be sufficiently large to find 2 components but not too large in order to avoid its under-estimation.

Many works were done in this problematic. We can cite, in particular, the first works of Polonik [146] for example, on this question, who proposed to evaluate the number of connected components after having estimated the level sets of the density f. However, performing clustering by first estimating density becomes a difficult task in practice, as soon as the data are high-dimensional or/and are in small sample size. In a practical point of view, several authors tried to avoid such an estimation step and some of them were interested in working on the connectivity graphs. We can cite, in particular, the analysis of Brito *et al.* [25] who studied the connectivity of random mutual g-nn graphs. They were particularly interested in asymptotic results in the case of non-noisy data and proved that for a certain order, the choice of g ensures that connected components of the mutual g-nearest neighbor graph correspond to the true underlying structure. In the same spirit, Biau *et al.* [13] proposed a simple algorithm based on the idea to form a rough skeleton of a certain level set $\mathcal{L}(t)$ and to count the number of connected components of the resulting graph. They provided asymptotic results for the estimation of the connected components of the level set $\mathcal{L}(t)$ in the case of an ε-neighborhood graph. More recently, Maier *et al.* [117] also defined the clusters as connected components of the t-level set of a density, in order to remove noise points from the sample. However, conversely to Biau *et al.*, they worked on the g-nearest neighborhood graphs for which they derived theoretical results to determine for which value of g the probability of cluster identification is maximized.

7.3 Block-clustering in a probabilistic framework

An other kind of approaches, presenting the same goals as seriation, was developed in the literature, and both deals with high dimension and visualization. These methods are named differently in the literature and the most well-known names are block-clustering, bi-clustering, co-clustering or two-mode clustering methods.

Conversely to the seriation, block-clustering aims to cluster, rather than to permute, rows and columns of a matrix but it enables also to interpret directly on the data matrix the results of the clustering. Block clustering was introduced by Hartigan [81] who proposed to simultaneously cluster both the observations and the variables. The first works of Hartigan on this question were based on a measure evaluating the deviation between the rearranged matrix and an "ideal" data matrix. Since then, many authors worked on this question and an overview of the different methods of block-clustering is detailed in [118, 131].

Among all the block clustering methods, this paragraph will focus on approaches developed

for binary data, in a probabilistic framework. In particular, we are going to briefly resume the latent models introduced, in particular by Govaert and Nadif [69, 70] and extended by Wyse and Friel [183].

The underlying idea of the latent block model for binary tables was to extend the finite mixture model, initially used in model-based clustering [59], to partition the rows (observations) and the columns (features) of a matrix. Govaert and Nadif [67, 68, 70], the firsts, proposed a latent block model which aims to find both a partition among the observations into K clusters, and a partition into G clusters among the p variables. They proposed to recast the block clustering issue in terms of mixture model approach.

Let us consider the probabilistic framework of finite mixture model in which the overall population is considered as a mixture of these groups and each component is modeled by a probability distribution. Let us consider a given $n \times p$ binary matrix Y, defined by $Y = \{(y_{ij}) \mid i \in I, j \in J\}$ where $I = \{1, \ldots, n\}$ stands for the indexes of n observations and $J \in \{1, \ldots, p\}$, the set of p variables. The aim of their approach, is to divide both the observations into K homogeneous groups *i.e.* adjoin to each observation y_i a value $z_{ik} = 1$, for $k = 1, \ldots, K$, if the observation y_i belongs to the kth cluster and $z_{ik} = 0$ otherwise, and to divide the set of p variables into G blocks. In the same manner, for each variable, a value $\omega_{jg} = 1$ for $g = 1, \ldots, G$ is adjoined to the jth variable, if it belongs to the block variable g.

Govaert and Nadif assume that the rows and columns of the dataset considered may be reordered, so that the matrix can be represented as a $K \times G$ blocks. The data, in blocks, are modelled by the same density. By assuming moreover, that the rows clustering is independant of the columns one, and by assuming a local independance of the random variable Y_{ij} conditionally to z and ω, then the conditional density function is:

$$f\left(y \mid z, \omega; \theta\right) = \prod_{i,j,k,g} \psi\left(y_{ij}; \alpha_{kg}\right)^{z_{ik}\omega_{jg}},$$

where $\psi\left(.\,; \alpha_{kg}\right)$ stands for a probability density function of parameter α_{kg} and θ the parameter of the mixture model. Besides, since the data are binary, $\psi\left(.\,; \alpha_{kg}\right)$ is assumed to be a Bernoulli density function with parameter $\alpha_{kg} \in [0, 1]$ and by denoting $\theta = (\pi_1, \ldots, \pi_K, \tau_1, \ldots, \tau_G, \alpha_{11}, \ldots, \alpha_{KG})$ where π_k and τ_g stand for the mixing proportions of K clusters of observations and of G blocks of p variables, the Bernoulli latent block model (BLM) is defined according to the following density function:

$$f(y, \theta) = \sum_{z,\omega} \prod_{i,k} \pi_k \prod_{j,g} \tau_g \prod_{i,j,k,g} \psi\left(y_{ij}; \alpha_{kg}\right)^{z_{ik}\omega_{jg}},$$

where $\psi\left(y; \alpha_{kg}\right) = \left(\alpha_{kg}\right)^{y}\left(1 - \alpha_{kg}\right)^{1-y}$. The procedure, used for estimating the parameters of the BLM model, is based on an EM algorithm. However, because of the dependence structure among the rows and columns of the data matrix, the computation of the expectation of log-

likelihood is not directly tractable in the E-step. This leads Govaert and Nadif, to propose an approximation to the joint distribution. The main asset of this approach is that it is more parsimonious than the application of two Bernoulli mixture models, independently, on the sets I and J. Moreover, such a model allows probabilistic justifications and allows explicit modelling of noise in the data. However, a main limitation remains in the BLM model of Govaert and Nadif. Indeed, they assume that the numbers of clusters K and of blocks G are known. It appears that such an hypothesis is very strong, as the determination of these two parameters has a considerable influence on the results of clustering algorithm. Several authors worked on the determination of these both parameters in latent block model approaches, such as the very recent work of Wyse and Friel [183].

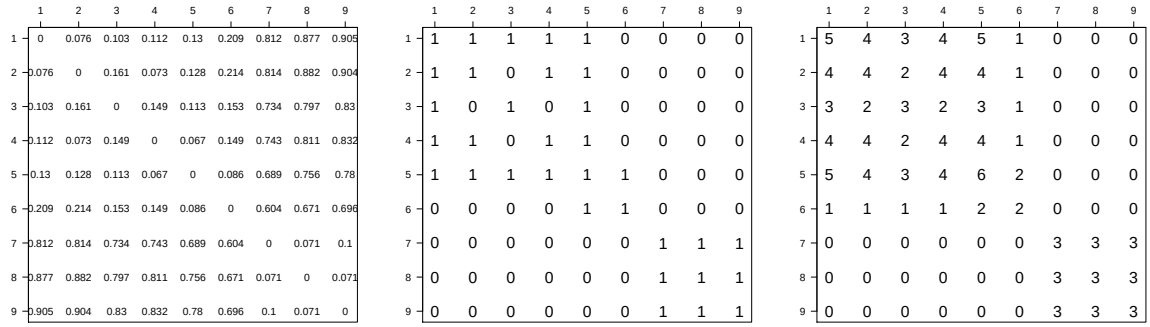
Chapter 8

The PB-Clust algorithm

The determination of the number of clusters, in a dataset, remains a challenging question in a non-parametric clustering task. Many authors proposed, either theoretical, or empirical results, to deal with this issue. We propose, in this manuscript, a visualization method based on the seriation approach.

The key-idea of our approach is to visualize the intrinsic structure of the data by introducing sparsity, *i.e.* zeros, in the dissimilarity matrix. To do that, we introduce the notion of common neighbors in order to build a dissimilarity matrix, and from which a family of $(0, 1)$-tables is declined. This collection, consisting of binary matrices with different degrees of sparsity, are then ordered such as the adjacent rows and symmetrically, adjacent columns, are the most similar. Moreover, we propose a parsimonious block clustering algorithm (PB-Clus) which rearranges rows and columns, according to the similarity measure based on the inner-product. It produces a family of reordered matrices, with different degrees of sparsity. Amongst this collection of rearranged matrices, only one is chosen according to a compactness criterion, such as it best reveals the intrinsic structure of the data.

In this Chapter, the notion of shared neighbors based on an ε-neighborhood will be introduced and described. In particular, geometrical aspects of this common neighborhood will be discussed before comparing it with the existing measures in the literature. Moreover, we construct a collection of binary matrices from the common neighborhood. As we want to visualize the intrinsic structure of the data, we propose a simple seriation algorithm based on a forward stepwise approach, in order to rearrange the collection of binary matrices. As we obtain a collection of reordered matrices, one need a criterion in order to select the *best* visualization among them. This criterion, based on the cluster compactness, will be introduced, in the same paragraph. Finally, some computational aspects of the PB-Clus algorithm will be discussed, such as the initialization strategy, the computational cost of this algorithm and also the calibration of the ε-neighborhood parameter, before ending this section with some links between our approach and the level set issue.



(a) Distance matrix.

	1	2	3	4	5	6	7	8	9
1	0	0.076	0.103	0.112	0.13	0.209	0.812	0.877	0.905
2	0.076	0	0.161	0.073	0.128	0.214	0.814	0.882	0.904
3	0.103	0.161	0	0.149	0.113	0.153	0.734	0.797	0.83
4	0.112	0.073	0.149	0	0.067	0.149	0.743	0.811	0.832
5	0.13	0.128	0.113	0.067	0	0.086	0.689	0.756	0.78
6	0.209	0.214	0.153	0.149	0.086	0	0.604	0.671	0.696
7	0.812	0.814	0.734	0.743	0.689	0.604	0	0.071	0.1
8	0.877	0.882	0.797	0.811	0.756	0.671	0.071	0	0.071
9	0.905	0.904	0.83	0.832	0.78	0.696	0.1	0.071	0

(b) ε-neighborhood matrix.

	1	2	3	4	5	6	7	8	9
1	1	1	1	1	1	0	0	0	0
2	1	1	0	1	1	0	0	0	0
3	1	0	1	0	1	0	0	0	0
4	1	1	0	1	1	0	0	0	0
5	1	1	1	1	1	1	0	0	0
6	0	0	0	0	1	1	0	0	0
7	0	0	0	0	0	0	1	1	1
8	0	0	0	0	0	0	1	1	1
9	0	0	0	0	0	0	1	1	1

(c) Matrix of common neighbors.

	1	2	3	4	5	6	7	8	9
1	5	4	3	4	5	1	0	0	0
2	4	4	2	4	4	1	0	0	0
3	3	2	3	2	3	1	0	0	0
4	4	4	2	4	4	1	0	0	0
5	5	4	3	4	6	2	0	0	0
6	1	1	1	1	2	2	0	0	0
7	0	0	0	0	0	0	3	3	3
8	0	0	0	0	0	0	3	3	3
9	0	0	0	0	0	0	3	3	3

Figure 8.1: Dissimilarity matrix of data based on Euclidean distance (a), its associates ε-neighborhood matrix with $\varepsilon = 0.145$ (b) and its common neighbors matrix (c).

8.1 A family of common neighborhood matrices

We introduce in this paragraph the notion of the common neighborhood which enables us to build a family of sparse matrices.

8.1.1 Common neighborhood

8.1.1.1 Definition of the common neighborhood

In this section, we define the notion of shared neighbors, which consists of a similarity measure between pairs of observations. The intuition behind this measure is that, by considering an ε-neighborhood centered on each dataset, the more the number of common neighbors between two observations is high, the more these ones are similar.

Let us then introduce the following definition:

Definition 8.1.1. *Let $\{x_1, \ldots, x_n\}$ denote a dataset of n observations which is described in a p-dimensional space having a metric. Let $D = (d_{ij})_{i,j\in\{1,\ldots,n\}}$ be its corresponding $n \times n$ dissimilarity matrix. Moreover, let us define $\bar{D}$ a binary ε-neighborhood matrix from D whose elements satisfies $\bar{d}_{ij} = \mathbf{1}_{\{d_{ij}\leq\varepsilon\}}$. Then, the matrix of common neighbors is defined such as:*

$$B = \bar{D}^t\bar{D} = \bar{D}\bar{D}^t.$$

Then, each element b_{ij} of the matrix B stands for the number of neighbors shared by the ε-neighborhood of observations i and j. In particular, the elements b_{ii} stands for the number of neighbors contained in the ε-neighborhood of the observation i.

Different remarks can be done from Definition 8.1.1. Firstly, we can note that the dissimilarity matrix can be based on different similarity metrics. The most common one is of course the Euclidean distance but other distances, or measures, can be used depending on the data

considered. In particular, for DNA sequences, a well-known similarity measure is the Pearson coefficient. The second remark concerns the choice of the smoothing parameter ε which plays a major role in the definition of the neighborhood. Indeed, this parameter has to be small enough to separate the clusters, but sufficiently large to group points belonging to the same cluster and to avoid the over-estimation of the number of clusters. The choice of ε will depend on the dataset studied and we will discuss about its calibration in Section 8.2.2.3. Finally, we can note that the common neighborhood matrix B is symmetric ($b_{ij} = b_{ji}$, $\forall i, j \in \{1, \ldots, n\}$) and its elements are such that $b_{ij} \in \mathbb{N}$. They stand for the number of common elements in the ε-neighborhoods of observations x_i and x_j. An example is given in Figure 8.1a where from a dissimilarity matrix based on the Euclidean distance, a binary ε-matrix is built with $\varepsilon = 0.145$ (Figure 8.1b), and its associated common neighborhood matrix is computed and illustrated in Figure 8.1c.

When the ε-neighborhood is built from a distance, we can interprate geometrically the common neighbors. Indeed, They stand for the cardinality of the intersection of two balls, centered respectively in x_i and x_j with radius ε. In this case, an other definition for the common neighborhood which is equivalent to the previous one, can then be stated:

Definition 8.1.2. *Let $\{x_1, \ldots, x_n\}$ be a dataset of n observations described in a p-dimensional space having a metric. For $\varepsilon > 0$, the common neighborhood between a pairwise of observations (x_i, x_j) is defined as:*

$$b_{ij} = card\left(\mathcal{B}\left(x_i, \varepsilon\right) \cap \mathcal{B}\left(x_j, \varepsilon\right)\right),$$

where $\mathcal{B}\left(x, \varepsilon\right)$ denotes the ball centered in x with radius fixed to ε and $card\left(A\right)$ the cardinality of A.

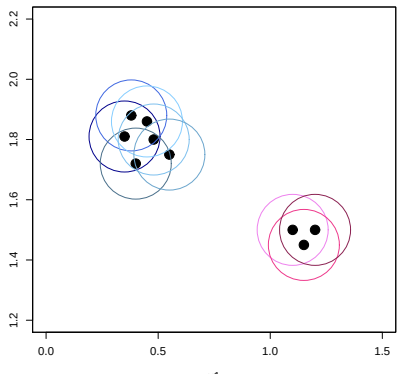
This definition is illustrated in Figure 8.2a; data points are plotted in their 2-dimensional space, with their respective ε-neighborhood determined by balls centered on each data point, with radius $\varepsilon = 0.145$. The resulting common neighborhood matrix is depicted in Figure 8.2b.

8.1.1.2 A family of binary matrices

The common neighbors matrix gives the number of neighbors shared in the ε-neighborhood of a pair of observations. Consequently, this supposes that higher this number is, the more the observations are similar. From this, the level of common neighbors can be viewed as a level of sparsity inside the data. Indeed, by thresholding the common neighborhood matrix for a certain level of common neighbors, the resulting matrix tends to keep the data points which are, in terms of graph theory, well-connected with the rest of the data and to stress the intrinsic structure of the data.

From this idea, we are going to introduce the collection of binary neighborhood matrices according to this definition:

Definition 8.1.3. *Let $B = \left(b_{ij}\right)_{i,j \in \{1,\ldots,n\}}$ denotes a common neighborhood matrix built from*



(a) Plot of data.

	1	2	3	4	5	6	7	8	9
1	5	4	3	4	5	1	0	0	0
2	4	4	2	4	4	1	0	0	0
3	3	2	3	2	3	1	0	0	0
4	4	4	2	4	4	1	0	0	0
5	5	4	3	4	6	2	0	0	0
6	1	1	1	1	2	2	0	0	0
7	0	0	0	0	0	0	3	3	3
8	0	0	0	0	0	0	3	3	3
9	0	0	0	0	0	0	3	3	3

(b) Matrix of common neighbors.

Figure 8.2: Plot of data points in a 2-dimensional space (a) and the associated common neighbors matrix (b) for a fixed value of ε ($\varepsilon = 0.145$).

$X \in \mathbb{R}^{n\times p}$. Then, B_λ defines a binary matrix with level of sparsity $\lambda \in \{1, \ldots, \lambda_m\}$ and $\lambda_m = \max_{i,j}(b_{ij})$, noted as λ-matrix, if its elements b_{ij}^λ satisfy the following conditions:

$$b_{ij}^\lambda = \begin{cases} 1 & \text{if} \quad b_{ij} \geq \lambda \\ 0 & \text{otherwise} \end{cases}. \tag{8.1.1}$$

In this case, $(B_1, \ldots, B_{\lambda_m})$ stands for a collection of binary neighborhood matrices whose the level of sparsity depends on the number of common neighbors.

This definition enables to introduce sparsity among the data. Indeed, according to the threshold λ, we can note that the number of pairs of observations satisfying the condition, in Definition 8.1.1, decreases with the increase of the threshold λ. Consequently, the λ-matrix is filled up with zeros, with respect to the λ-level and becomes then sparsier.

Proposition 8.1.1. *Let consider $(B_1, \ldots, B_{\lambda_m})$ a family of λ-matrices. There exists inclusion relations $\subseteq$ between the collection of λ-matrices:*

$$B_{\lambda_m} \subseteq B_{\lambda_m - 1} \subseteq \cdots \subseteq B_\lambda \subseteq \cdots \subseteq B_1, \tag{8.1.2}$$

where $\lambda \in \{1, \ldots, \lambda_m\}$ stands for the level of common neighbors and $\lambda_m = \max_{i,j}(b_{ij})$ denotes the maximum number of shared neighbors.

Proof. Let us consider $B = (b_{ij})_{i,j\in\{1,\ldots,n\}}$ the common neighborhood matrix and its corresponding family of λ-matrices. In the case $\lambda = 1$, the 1-matrix B^1 is equivalent to the ε-neighborhood graph. Indeed, by using the definition of the common neighborhood graph, its elements b_{ij} are defined such as $b_{ij} = \text{card}\left(\mathcal{B}(x_i, \varepsilon) \cap \mathcal{B}(x_j, \varepsilon)\right)$ which can be rewritten as $b_{ij} = \text{card}\left(\|x_i - x_j\| \leq \varepsilon\right)$. Consequently, according to Definition 8.1.1, the elements of the 1-matrix associated to B satisfy $b_{ij}^1 = \mathbf{1}\{b_{ij} \geq 1\}$ *i.e.* $b_{ij}^1 = \mathbf{1}\{\|x_i - x_j\| \leq \varepsilon\}$ which corresponds to the definition an ε-neighborhood. This B^1 matrix is the most complex form of

3	0	0	0	3	0	0	0	3
0	3	0	0	0	3	0	3	0
0	0	2	0	0	0	2	0	0
0	0	0	1	0	0	0	0	0
3	0	0	0	3	0	0	0	3
0	3	0	0	0	3	0	3	0
0	0	2	0	0	0	2	0	0
0	3	0	0	0	3	0	3	0
3	0	0	0	3	0	0	0	3

(a) Common neighborhood matrix.

1	0	0	0	1	0	0	0	1
0	1	0	0	0	1	0	1	0
0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	1
0	1	0	0	0	1	0	1	0
0	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0
1	0	0	0	1	0	0	0	1

(b) λ-matrix with $\lambda = 1$.

1	0	**0**	**0**	1	0	**0**	0	1
0	1	**0**	**0**	0	1	**0**	1	0
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	0	**0**	**0**	1	0	**0**	0	1
0	1	**0**	**0**	0	1	**0**	1	0
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	1	**0**	**0**	0	1	**0**	1	0
1	0	**0**	**0**	1	0	**0**	0	1

(c) λ-matrix with $\lambda = 3$.

Table 8.1: Example of a family of λ-matrix for $\lambda = 1$ and $\lambda = 3$ from a common neighborhood matrix.

λ-matrices since all the possible connections between pairwises of observations are marked in a fixed ε-neighborhood.

By considering the B^2 matrix *i.e.* the λ-matrix with $\lambda = 2$, only the elements b_{ij} superior to 2 are kept:

1. case $b_{ij} = 0$, then $b^1_{ij} = b^2_{ij} = 0$.
2. case $b_{ij} = 1$, then $b^1_{ij} = 1$ and $b^2_{ij} = 0$.
3. case $b_{ij} > 1$, then $b^1_{ij} = b^2_{ij} = 1$.

Hence, since case 1. and case 3. are similar between the λ-matrices B^1 and B^2, and according to case 2., the elements of the matrix B^2 form subsets of the matrix B^1 and we can note that $B^2 \subseteq B^1$. This argument can be generalized for $\lambda > 2$ and it follows that the remaining pairwises of observations are subsets of the 1-matrix. Consequently, there exists inclusion relations between the collection of λ-matrices and this enables us to conclude. □

The interest of this property is that outliers and noisy data disappear with the increase of the λ-level, as the connections between pairwises become sparsier, with the increase of the threshold λ.

Before making some links with existing approaches, we illustrate the family of λ-matrices with the example depicted in Table 8.1. From a common neighborhood matrix in Table 8.1a., a 1-matrix is built by thresholding the matrix 8.1a. with a level $\lambda = 1$. All the connection points linked to the common neighbors matrix are kept conversely to the case with level $\lambda = 3$ where 3 observations were removed. The 3-matrix is thus much sparsier.

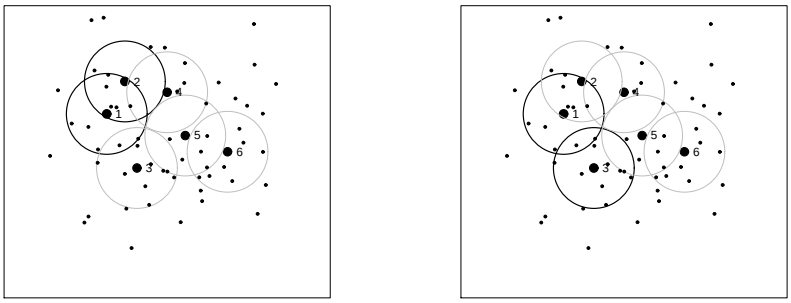
8.1.2 Link with existing neighborhood

At this point, some links can be done with the different kinds of neighborhoods existing in the literature. In particular, the notion of common neighbors introduced, is close to the shared nearest neighbors of Jarvis and Patrick [97]. In particular, conversely to our approach, in

which an ε-radius is fixed to determine a neighborhood, Jarvis and Patrick considered the k-nearest list of neighbors of each data. The number of shared neighbors is then defined in a pair of observations, as the number of correspondences between both lists of k-nearest neighbors. The main advantage of using the k-nn instead of an ε-distance, is perhaps the adaptation of the maximum distance between the observation i and its kth nearest neighbor. It enables in particular to deal with cases where the distances intra-clusters are different. However, in some situations, since the volume of the sphere containing the k-nn changes according to the compactness of the data points, an isolated point can share the same neighborhood as grouped points. To avoid this situation, Jarvis and Patrick added a condition on the pairwises of data points such as the shared neighborhood is computed only if both observations belong to the list of k-nearest neighbors of each of them. However, it appears that such cautions, in certain cases, are non-efficient. In particular, in the case of Gaussian samples with different covariance matrices, Ertörz *et al. [51]* showed that the shared nearest neighbors measure tended to overestimated the number of clusters.

The common neighborhood matrix can also be compared to the transitive fuzzy similarity measure proposed by Vathy-Fogarassy *et al.* [170]. Their measure is a normalized version of the shared neighborhood which, in addition, takes into account the "spreadness" of the nearest neighbors (the nearest neighbors of the nearest neighbors, ... , of the nearest neighbors of the pairs of observations). However, an issue remains in their approach since no method is proposed to calibrate the degree of spreadness. In our approach, such a spreadness is already included in the common neighbor measure according to the ε-neighborhood. Indeed, when the ε-neighborhood is defined according to a distance, our measure stands for the number of elements contained in the intersect of balls with radius ε. Thus, in the same manner as the shared neighbors, we dispose of information about the close neighbors of observations x_i on the one hand, *i.e.* those which belong to the intersection of their balls as it is illustrated in Figure 8.3a. On the other hand, the measure provides also information about the farest neighborhood of x_i as it is depicted in 8.3b. We can remark that, according to the ε-neighborhood, there is no need to have the same restriction on the neighborhood as the shared common neighbors defined by Jarvis and Patrick. Consequently, the common neighbors measure provides an idea of the spreadness of these neighbors. It presents also the advantage that no additional parameter needs to be calibrate contrary to the approach of Vathy-Fogarassy having to calibrate both k and the degree of spreadness.

Therefore, in our approach, the common neighbors matrix is defined through an ε-neighborhood. This enables to both consider the closest points, as they correspond to high level of the common neighbors, but also the farest ones inside the ε-neighborhood, which are associated to a low level of common neighbors. This gives a certain idea of the spreadness of the neighborhood. Moreover, conversely to the existing approaches, our idea is to introduce sparsity inside the dataset in order to visualize the intrinsic structure of the data determining thus, the number of clusters. We have introduced a collection of λ-matrices and differently to



(a) $x_1 \in \mathcal{B}(x_2, \varepsilon)$ and $x_2 \in \mathcal{B}(x_1, \varepsilon)$. The common neighborhood measure is 7.

(b) $x_1 \notin \mathcal{B}(x_2, \varepsilon)$ and $x_2 \notin \mathcal{B}(x_1, \varepsilon)$. The common neighborhood measure is 2.

Figure 8.3: Two situations of the ε-neighborhood which illustrates the notion of spreadness.

the existing approaches, our work proposes to use these structures and their different degrees of sparsity in order to stress and visualize the structure of the data. This is done according to a seriation approach.

8.2 The PB-Clus algorithm

Since we dispose of a collection of binary symmetric matrices with different degree of sparsity, we are going to use a seriation approach to reorder the rows and the columns of each matrix in order to produce a block-representation. A collection of reordered matrices is then created. Besides, the level of sparsity λ is chosen from the quality of the block-visualization according to a compactness criterion and allows to assess the number of clusters in the data.

8.2.1 Seriation on the collection of λ-matrices

8.2.1.1 Reordering criterion

As the common neighborhood and its associated family of λ-matrices reveal both a local structure, between pairs of observations, and a global structure (group clusters), we are going to use such an information according to a seriation approach, in order to obtain a block representation of the λ-matrices. As we saw in Chapter 7, there exist several criteria in the literature to realize the reordering of a matrix. All the measures of similarity introduced previously can be applied, but the reordering criterion, that we chose here, is the well-known inner-product already used in the seriation problem by [6] and [123] in particular.

Let us consider the common neighborhood matrix $B = (b_{ij})_{i,j \in \{1,\ldots,n\}}$ and its family of the λ-binary matrices $(B_1, \ldots, B_{\lambda_m})$ with $\lambda_m = \max_{i,j}(b_{ij})$. Then, the permutation function Ψ which aims to optimize the sum of consecutive scalar is as follows:

$$\forall \lambda \in \{1, \ldots, \lambda_m\} : \quad \Psi^* = \arg\max_{\Psi} \sum_{i=1}^{n-1} \frac{(b^{\lambda}_{\Psi(i)})^t b^{\lambda}_{\Psi(i+1)}}{||b^{\lambda}_{\Psi(i)}||.||b^{\lambda}_{\Psi(i+1)}||}.$$

The use of the inner-product from the λ-matrices enables to group the observations whose share a same neighborhood. Indeed, the λ-matrices are made of $(0,1)$ n-dimensional row vectors and each of them indicates if two observations have in common some neighborhood for a certain level λ. The difference in terms of number of common neighbors is then erased and it remains only a binary information about the neighborhood. Hence, given a λ level, two observations are similar if they share the same neighborhood, implying that, the inner product will be equal or almost equal to 1. At the opposite, if 2 observations have no common neighbor, then the inner product will tend to 0. Consequently, the criterion based on the sum of the consecutive inner products is maximum when the adjacent rows and columns share the same neighborhood.

Besides, according to the links established between the common neighborhood and the clusters, the inner-product have interesting geometrical properties to define a cluster.

Definition 8.2.1. *A cluster $\mathcal{G}$ is a subset of points whose their vector of common neighbors are correlated between them.*

Consequently, in the easiest case, clusters can be simply defined as connected components which leads to the following fact: the vectors of a same cluster will be correlated between them and conversely, the vectors associated to different clusters will be pairwise orthogonal. If the groups are not well-separated, no orthogonal components will be detected and groups structure will not appear clearly. We can then foresee the interest of the collection of λ-matrices which enables to introduce sparsity in the data, in order to reveal subsamples forming well-separated groups.

8.2.1.2 The algorithm

The parsimonious block clustering algorithm that we propose and named PB-Clus, is based on the previous definition. PB-Clus is a forward stepwise algorithm which aims to rearrange the rows and columns of the collection of λ-matrices such as the adjacent rows/columns are the most similar.

Initially, we dispose of a family of λ-matrices, built from a common neighborhood matrix. Let us consider a certain λ-level and its associated λ-matrix. Initially, an observation i is chosen amongst the n observations and we denote by b^{λ}_i the boolean vector associated to the ith column of the λ-matrix. From this vector b^{λ}_i, the algorithm will create three different subsets: a first subset noted $\mathcal{S}_\ell$ which lists the elements whose the associated binary vectors are colinear to b^{λ}_i, the correlated ones are grouped in the subset $\mathcal{S}_c$ and the last subset $\mathcal{S}_o$ consists of the elements whose the associated binary vectors are orthogonal to the vector b^{λ}_i. The subset $\mathcal{S}_\ell$ has the priority and its elements are placed next to the observation i, then the

algorithm considers the element in $\mathcal{S}_c$ whose the linked binary vector of common neighbors is the more correlated with b_i^λ. If the subsets $\mathcal{S}_\ell$ and $\mathcal{S}_c$ are empty for a current b_i^λ, then a vector in $\mathcal{S}_o$ is randomly selected to be placed next to i. Then, new subsets $\mathcal{S}_\ell, \mathcal{S}_c, \mathcal{S}_o$ are formed from the last observation ordered and this scheme is iterated until all the observations are rearranged. We can detail the algorithm in this manner for the case of a λ-matrix:

1. Let $\mathcal{L}_o$ the set which will contain the list of reordered observations and $\mathcal{L}_i$ the list of all observations. Initially, $\mathcal{L}_o = \emptyset$ and $\mathcal{L}_i = \{1, \ldots, n\}$.
2. Let i the current observation and $\mathcal{L}_i$ stands for the list of remaining non-ordered observations.
3. Create the sets $\mathcal{S}_\ell^i, \mathcal{S}_c^i, \mathcal{S}_o^i$ from $\mathcal{L}_i$.
4. a. If $\mathcal{S}_c \neq \emptyset$, then select the observation j whose the binary λ-vector is the most correlated to i.

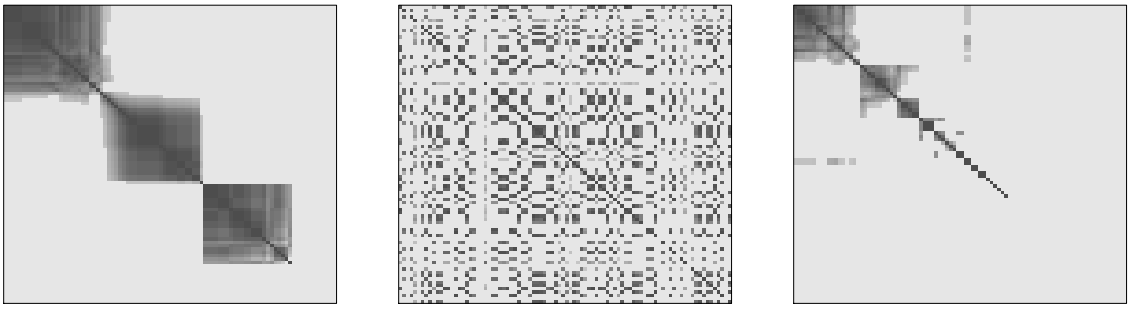
   b. If $\mathcal{S}_c = \emptyset$ and $\mathcal{S}_\ell = \emptyset$, then select randomly an observation j in $\mathcal{S}_o$.
5. Complete $\mathcal{L}_o \leftarrow i, \mathcal{S}_\ell$.
6. Replace $\mathcal{L}_i \leftarrow \mathcal{L}_i \setminus \{i, \mathcal{S}_\ell, j\}$ and $i \leftarrow j$.

This procedure is iterated until all the elements in the list $\mathcal{L}_i$ are arranged. The final ordering list $\mathcal{L}_o$ is then applied on the λ-matrix, and this enables us to visualize it. It is also possible to visualize the matrix from the rearranged matrix containing the inner-product of pairs of binary vectors which makes the visualization softer. Besides, we can remark that this algorithm is a forward stepwise algorithm as it considers locally the rearrangement of the observations from different subsets. This implies that the position of observations, already ordered, remains unchanged for a current permutation. Consequently, the present algorithm proposes an approximative solution for the reordering of a matrix and is not optimal in comparing to some existing approaches in the literature.

The PB-Clus algorithm is applied for several values of λ. This creates a collection of reordered λ-matrices. Therefore the level of sparsity λ needs to be selected, in order to propose a visualization, such as the intrinsic structure of the data appears clearly. This criterion is detailed in the following paragraph.

8.2.1.3 A compactness criterion

For a fixed ε-neighborhood, the main issue of this collection of λ-matrices remains in the determination of the level λ. The seriation criterion can not be used in our context to select the λ-level since it would favour small values of λ *i.e.* little sparsness. Indeed, a high λ-level would imply very sparse matrices with lots of zero columns. Their inner products would be equal to zero, which would lead to decrease the seriation criterion.



(a) Selected λ-rearranged matrix with $\lambda = 14$ and $\mathcal{C} = 12.5$.

(b) Random λ-matrix with $\lambda = 14$ and $\mathcal{C} = 111.2$.

(c) Selected λ-rearranged matrix with the non-standardized criterion ($\lambda = 22$).

Figure 8.4: Influence of the structure selected by the compactness criterion.

In order to tackle this issue, we propose a criterion based on the quality of the visualization of the structure of the data since we want to determine the number of clusters in the dataset. Let us introduce the following compactness criterion:

Definition 8.2.2. *Let us consider the family of rearranged λ-matrices noted as $(\mathcal{B}_\lambda)_{\lambda\in\{1,\dots,\lambda_m\}}$. Then, the λ-level which corresponds to the best visualization of the intrinsic structure of the data is:*

$$\lambda^* = \arg\min_{\lambda} \quad \frac{\sum_{i=1}^{n}\sum_{j=1}^{n-1}\left|b_{i,j}^{\lambda} - b_{i,j+1}^{\lambda}\right|}{\min\left(\sum_{i,j=1}^{n} b_{ij}^{\lambda}, \sum_{i,j=1}^{n}\left(1 - b_{ij}^{\lambda}\right)\right)},$$

where $\left(b_{ij}^{\lambda}\right)_{i,j\in\{1,\dots,n\}}$ denotes the elements of the λ-rearranged matrix $\mathcal{B}_\lambda$.

We can remark that this criterion is linked, in a certain way, to the consecutive ones property. Indeed, this criterion is minimized when the number of switches between the 0s and 1s on the rows of the matrix is weak. This implies that the rearranged matrix will have a compacter representation. Indeed, the compactness criterion is mainly defined by the quantity $\sum_{j=1}^{n-1}\left|b_{i,j}^{\lambda} - b_{i,j+1}^{\lambda}\right|$ which corresponds to the number of changes between zeros and ones on ith row. Moreover, this quantity is minimum when the 1s are ordered in consecutive positions in the row i and is maximum in the case of a randomized binary matrix. Such a situation is illustrated in Figure 8.4a which stands for the λ-rearranged matrix selected by the minimum of the compactness criterion and Figure 8.4b which corresponds to the same λ-matrix but in a randomized order. We can observe that, for a same λ-level ($\lambda = 14$), the compactness criterion is minimum for the block representation in Figure 8.4a for a value of $\mathcal{C} = 12.1$ whereas it is ten times larger in the randomized case since $\mathcal{C} = 111.2$.

However, the criterion is standardized by the minimum between the total number of zeros ($\sum_{i,j=1}^{n}\left(1 - b_{ij}^{\lambda}\right)$) and of ones ($\sum_{i,j=1}^{n} b_{ij}^{\lambda}$) in the rearrange matrix considered. This additional term aims to prevent the case of an infinite sparsity. Indeed, from the same situation in

Figure 8.4a, the non-normalized criterion selects a sparser λ-matrix ($\lambda = 22$) which is however useless to determine the number of clusters (see Figure 8.4c).

8.2.2 Computational considerations

8.2.2.1 Initialization

The PB-Clus algorithm is a finite and non-iterative algorithm meaning that there is no convergence problem which occurs. The solution obtained by the algorithm depends only on the initial row chosen to start the sequential selection process. Even though the algorithm is not sensitive to the starting point since the local neighborhood is preserved, the visualization of the reordered matrix can however suffer from a bad initialization. Indeed, since the proposed algorithm reorders objects into a sequence along a one-dimensional continuum, a "bad" initialization can lead to separate a same cluster in two parts because of the one-dimensional ordering. This situation is reinforced by the fact that PB-Clus is a forward algorithm which considers only the best similarity between to the current observation i and its following one $i+1$ and does not take into account the order of the i first observations. Consequently, in order to tackle this issue, we propose a deterministic strategy for initializing the algorithm. It is based on the notion of "core" point or "representation" point introduced by Ertöz *et al.* [51] or the notion of "high connectivity" in graphs theory. The underlying idea is to select an observation whose the probability to belong to a cluster is very high. In particular, this situation occurs when the element share lots of neighbors with other observations. Thus, the more the number of common neighbors is high for an observation, the more the probability of this observation to belong to a cluster is high. Consequently, we propose to initialize the PB-Clus algorithm by selecting for the first row of the rearrange matrix, the observation which has the highest number of common neighbors. This guarantees that this observation remains in a high density area.

8.2.2.2 Computational cost of the PB-Clus algorithm

The computation cost of the PB-Clus algorithm depends on several parameters: the size of the common neighborhood matrix, the collection of λ-matrices and those of each of λ-matrix. In particular, the construction of the common neighborhood matrix is obtained according to the ε-neighborhood. Indeed, the computation of the common neighbors can be comprehended through the computation of an adjacency matrix noted $\bar{\mathtt{D}}$ whose its entries are either 0 or 1 depending if an observation i belongs to the ε-neighborhood of an observation j (cf. Definition 8.1.1). The computation of the common neighborhood matrix is then obtained by multiplying the adjacency matrix with itself. The time complexity of a naive approach computing this adjacency matrix is at worst $o\left(n^3\right)$. However, we can expect that the number of common neighbors for each point will be small compared to n, leading to a sparsier common neighborhood matrix.

Moreover, the PB-Clus algorithm is admittedly more time consuming than the traditional seriation algorithm since it is executed not only on one matrix but on a collection of binary matrices. However, on the one hand, since PB-Clus is a forward algorithm, it is linear, and on the other hand, the thresholding of the common neighbors matrix by the λ-values, introduces lots of sparsity. The collection of binary matrices which refers to different degrees of sparsity becomes very sparse as the λ-value increases: the more the λ-level is high, the more the λ-matrix is filled up with zeros removing thus column and row vectors of the study. Therefore, the number of ordering elements decreases drastically which consequently drops the computational time.

8.2.2.3 Choice of the ε-neighborhood

The PB-Clus algorithm is based on two parameters which are the λ-level and the ε-neighborhood. The first parameter aims to introduce a level of sparsity in the binary dissimilarity in order to highlight a subset of elements which characterize the intrinsic structure of the data. This λ-value allows to ease the visualization of the data structure and is determined according to the compactness criterion introduced in paragraph 8.2.1.3. The second parameter charaterizes the ε-neighborhood considered among the data. This parameter is important in the algorithm since it determines the neighborhood of each observation. In particular, this parameter needs to be small enough to separate the clusters but sufficiently large to group points belonging to the same cluster by avoiding the building of subclusters. The ε parameter is consequently indirectly linked to the quality of the visualization of a rearranged matrix. Consequently, in the same manner as the λ-level, the selection of the value of ε could be done according to the compactness criterion.

However, we propose, in practice, to use the distribution of distances of pairs of observations since the ε-distance is linked to the studied dataset. A list of different values of ε is created from quantiles varying between 0.1 and 0.5. We can expect that an interesting neighborhood remains below 50% of the distance between pairwises of observations. Besides, we noticed empirically that an ε-value fixed to the value associated approximatively with the first quartile of the distribution of pairwise distances, gave good results for the construction of the common neighbors matrix.

8.3 Links with level sets

The family of λ-matrices, presented in this chapter, aims to find the intrinsic structure of the data *i.e.* estimating the number of clusters in a dataset. From very recent readings, this family of λ-matrices could be viewed as several subsets of observations which could be related to the modes of an underlying density in the feature space. The aim of this paragraph is to suggest some links which could be done with the definition of level sets.

In particular, let us introduce the formal definition of cluster proposed by Hartigan [82]:

Given a random variable $X \in \mathbb{R}^p$ with probability f, let $t > 0$ denotes a fixed and positive level set, then the t-level set of the density f is defined as:

$$\mathcal{L}(t) = \{x \in \mathbb{R}^p | f(x) > t\},$$

and $k(t)$ denotes the number of connected components associated to $\mathcal{L}(t)$.

In addition to this definition, Hartigan showed that this collection of $k(t)$ clusters has a hierarchical structure. In particular, he pointed out that for any two clusters G_1 and G_2 with different levels or not, then $G_1 \subset G_2$ or $G_2 \subset G_1$ or $G_1 \cap G_2 = \emptyset$.

From this definition, many authors used this notion to tackle statistical issues such as density estimation, multi-modality tests or density contour clusters estimation and particularly through the notion of *excess mass approach* (see [146] for more details). In particular, most of the works consist principally of estimating the density contour clusters *i.e.* the level set $\mathcal{L}(t)$, before determining the number of connected components of the resulting set estimate.

More recently, some authors used this approach to cluster tree estimation as in the works of Struezle and Nugent [162] whereas others ([25, 43, 13, 116], *etc*) re-considered it, in the context of graphs theory, in order to tackle the issue of determining the number of components or of choosing a neighborhood graph.

The main work, which is of our interest, is the one of Biau *et al.* [13] which proposed an asymptotic graph-based estimator of the number of clusters by bypassing the estimation of the level set. In particular, they proposed to first construct a set $J(t) = \{i \in \{1, \ldots, n\} : f_n(X_i) \geq t\}$ where f_n stands for any estimation of the probability density f, before considering a sequence of real and positive numbers $\varepsilon \geq 0$ from which they define an ε-neighborhood matrix with binary entries. Then, from the set $J(t)$, a graph $\mathcal{G}(t)$ is produced, in which two elements x_i and x_j are assumed to belong to the same cluster, if there exists a chain of intermediate elements which are connected between them. Biau *et al.* showed that the true number of clusters $k(t)$ can be estimated by the number of connected components in the graph $\mathcal{G}(t)$. They pointed out that this estimator was consistent.

From this, we are tempted to draw a parallel between the graph $\mathcal{G}(t)$ and the collection of λ-matrices that we proposed in this section. Indeed, the λ-matrix stands for a connection graph which stresses pairs of observations sharing a same neighborhood with a level λ of sparsity. In the same manner as Biau *et al.,* who look for a chain of elements connected between them to form a cluster, the PB-Clus algorithm reorders objects into a sequence along a one-dimensional continuum. In our case, a cluster is formed when vectors of common neighbors are correlated between them and the number of clusters is estimated by the number of connected components *i.e.* the number of blocks obtained according to the visualization of the reordered matrix. For a further work, it would be interesting to show that our approach enables to compute an estimator of the number of clusters and to prove its possible consistency.

Chapter 9

Experiments

This section presents experiments, on simulated and real datasets, in order to highlight the main features of the PB-Clus algorithm from different situations. The aim of the first paragraph is to assess, on simulations, the behavior of the compactness criterion, with respect to the ε-neighborhood and the λ-level of common neighbors. On the second experiment, the PB-Clus algorithm is applied on simulated dataset where the problem of unbalanced groups occurs while in the second section, the case of overlapping groups will be considered. The third section will deal with noisy datasets. In particular, the behavior of the PB-Clus algorithm will be tested on noisy data and will be then compared with three other methods of seriation. Finally, the two last sections will focus on comparing on real datasets the efficiency of the PB-Clus algorithm on seriation benchmark datasets on the one hand and on clustering ones, on the other hand.

Note that, for all these experiment, the shared neighbors has been computed from a dissimilarity matrix which has been built using the Euclidean distance. However, the use of the Euclidean distance is not necessary, the practitioner can use the metric which seems to be the most appropriate for the considered datasets.

9.1 Choice of the ε-neighborhood

As the PB-Clus algorithm is entirely defined by the ε-neighborhood and the λ-level, the least we can do, is to assess the compactness criterion to select a visualization which clearly represents the intrinsic structure of the data. For this simulation, 300 observations are simulated: they consist of 3 balanced groups and each group is modeled by a Gaussian density in 2 dimensions for which the means vector of each cluster is as follows:

$$m_1 = (-0.4, -0.3),\ m_2 = (0.4, -0.3),\ m_3 = (0, 0.3).$$

Their covariance matrix is isotropic, common to the three clusters and fixed to $S = \sigma^2 \mathbf{I}_2$ with $\sigma^2 = 0.04$.

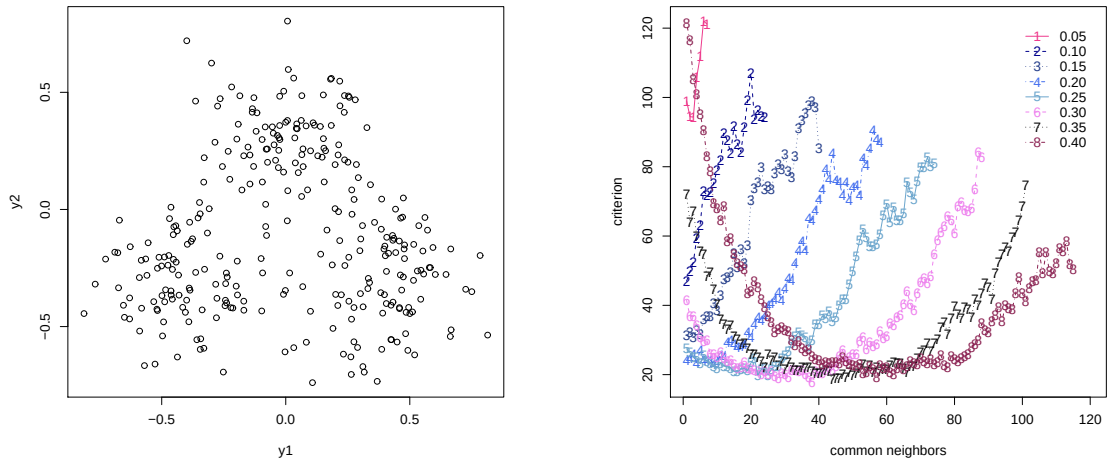
The projection of the data in their 2-dimensional space is illustrated in Figure 9.1a. We can notice that the three clusters overlap. Besides, the PB-Clus algorithm is executed from this simulation for different values of the ε-neighborhood and of common neighbors which varying with respect to ε. More precisely, we consider the distribution of pairwise Euclidean distances from which the values of ε is obtained for the quantiles 0.05, 0.10, 0.15, *etc*, until 0.40. Furthermore, the set of λ-values depends on the ε-neighborhood and the PB-Clus algorithm is executed on all its values. Finally, this simulation is repeated 20 times and the reported results is averaged on these repetitions.

Figure 9.1b stands for the evolution of the compactness criterion, averaging on the 20 trials, with respect to the ε-neighborhood and the λ-level of common neighbors. Please, note again that the values, given for ε, stand for the quantiles of the pairwise distances distribution of data. First of all, we can notice in Figure 9.1b the size of each ε-curves are different and as the ε-value increases, the minimum of the compactness criterion is reached for a larger number of common neighbors. This phenomenon is explained as, when ε increases, the volume of each ball, centered in a data point to build the near neighborhood, increases. This leads automatically to a rise of different levels of sparsity (λ-values) to consider. The effects of different sizes of the family of λ-matrices, can be viewed both in Figure 9.1b in a first hand and in Figures 9.2 in a second hand. Let us consider the case $\varepsilon = 0.05$ which corresponds to a very small ε-neighborhood. As we can observe in Figure 9.2a, the set of common neighbors is not large enough to provide a good visualization. In the opposite case, if we consider a too large ε-neighborhood, the collection of λ-matrices is larger and it appears that the PB-Clus is time-consuming. In average, the minimum of the compactness criterion is reached by the 6th curve, as it is depicted in Figure 9.1b. It corresponds to an ε-neighborhood which is associated to the quantile 0.30. Furthermore, we can note that the associated visualization depicted in Figure 9.2e. is very clear and stresses an intrinsic structure of the data of 3 groups comparing to Figures 9.2a, 9.2b, 9.2f. In addition, we can note that in average, for $\varepsilon \in [0.2; 0.3]$, the visual quality of the selected reordered matrices are similar (see Figures 9.2c, 9.2d and 9.2e) and this is also verified by the associated minimum value of compactness criterion which remain very close to each other ($\mathcal{C}_\lambda = 19.3$, $\mathcal{C}_\lambda = 18.9$ and $\mathcal{C}_\lambda = 17.7$ for $\varepsilon \in \{0.20, 0.25, 0.30\}$). Consequently, for the following experiments, we are going to fix the ε-value to 0.25.

9.2 Seriation on unbalanced datasets

The goal of this second experiment is to evaluate the efficiency of the PB-Clus algorithm, in the case of unbalanced data. In this paragraph, the data are simulated from a mixture of three Gaussians, in a 2-dimensional space, with different size and compactness. In particular, we consider the case where two clusters overlap. These clusters consist in, respectively, $n_1 = 100$, $n_2 = 200$ and $n_3 = 15$ observations each. Their means clusters are:

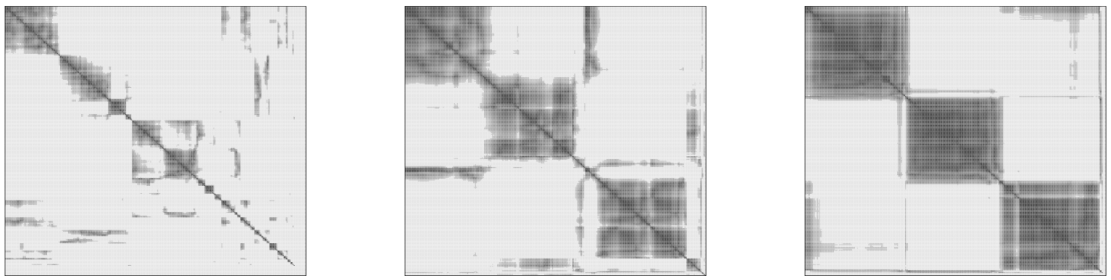
$$m_1 = (-12, -9)\,,\; m_2 = (0, 3)\,,\; m_3 = (5, 9)\,.$$



(a) Plot of data.

(b) Evolution of compactness criterion.

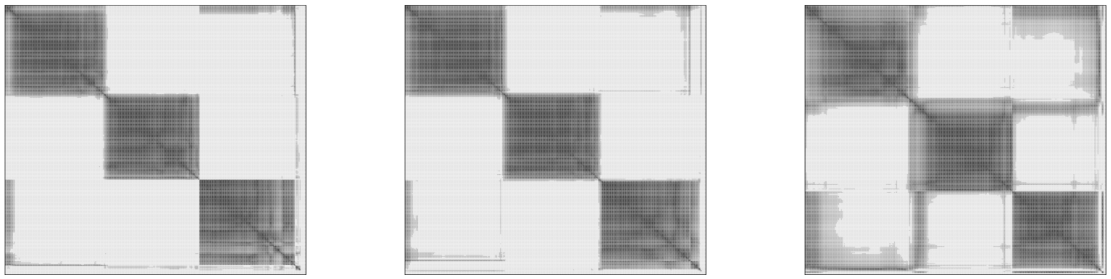
Figure 9.1: (a) Plot of simulated data in their 2-dimensional space and (b) evolution of the behavior of the compactness criterion with respect to the ε-neighborhood and the λ-level of common neighbors.



(a) Rearranged λ-matrix with $\varepsilon = 0.05$ selected by the compactness criterion ($\lambda = 2$).

(b) Rearranged λ-matrix with $\varepsilon = 0.10$ selected by the compactness criterion ($\lambda = 1$).

(c) Rearranged λ-matrix with $\varepsilon = 0.20$ selected by the compactness criterion ($\lambda = 10$).

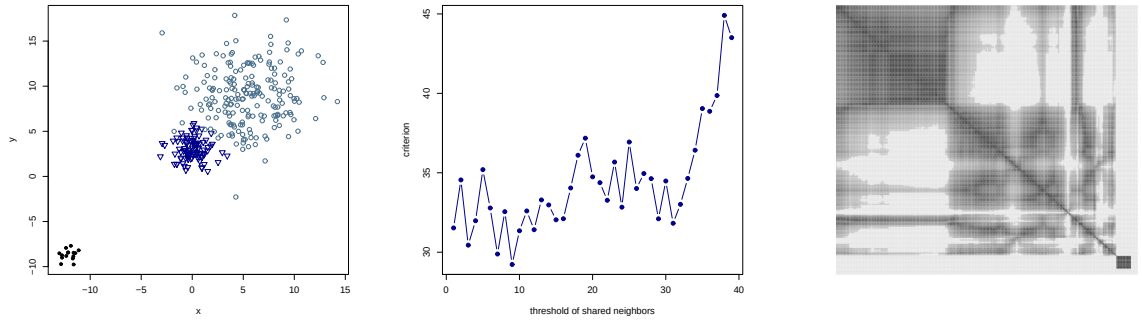


(d) Rearranged λ-matrix selected with an ε-neighborhood fixed $\varepsilon = 0.25$ ($\lambda = 25$).

(e) Selected rearranged λ-matrix by the compactness criterion ($\varepsilon = 0.3, \lambda = 38$).

(f) Rearranged λ-matrix with $\varepsilon = 0.40$ selected by the compactness criterion ($\lambda = 57$).

Figure 9.2: Rearranged matrices selected by the compactness criterion for each ε-neighborhood.



(a) Data plot. (b) Evolution of the selection criterion. (c) Rearranged matrix.

Figure 9.3: Unbalanced data: Projection of the simulated data in their 2-dimensional datasets (a), evolution of the selection criterion with respect to the nomber of common neighborhood (b) and the associated reordored matrix obtained from the minimum criterion (c).

Their covariance matrix have an isotropic shape, such as $\Sigma_i = \sigma_i \mathbf{I}_2$ for $i \in \{1, 2, 3\}$ with parameters $\sigma_1 = 1.5$, $\sigma_2 = 10$ and $\sigma_3 = 0.5$. Such a situation is illustrated in Figure 9.3a where the data are projected in their 2-dimensional space. The PB-Clus algorithm is applied on this simulated dataset for a neighborhood range varying between 1 (no sparsity) to 39 common neighbors. The maximum threshold of common neighbors is fixed from the empirical distribution of common neighbors shared between each pairwise of observations and stand for 75% of the population. We can see in Figure 9.3b, that the criterion of visualization is minimum for a shared neighborhood of 9 neighbors. Finally, Figure 9.3c illustrates the rearranged matrix for a level of shared neighbors of 9. As we can observe, the intrinsic structure is well-defined in this matrix since the 3 clusters can be visualized. In addition, this visualization gives information about the proximity of the two first clusters as we can see there exist connections between them which explain the overlapping clusters.

9.3 Influence of overlapping groups in visualization

In this second experiment, we would like to assess the influence of overlapping clusters, on the vizualisation of the number of clusters, in the PB-Clus algorithm. In order to measure it, we simulate 3 Gaussians in a 2-dimensional space such as their respective means satisfy:

$$m_1 = (\frac{\delta}{2}, 0),\ m_2 = (\delta, -\delta),\ m_3 = (0, -\delta),$$

where the parameter δ varies between 0 and 3. For each simulation, the position of clusters means fluctuates. This reflects the degree of overlapping clusters. Figures 9.4 stand for six different studied situations where the overlapping parameter $\delta = \{0, 1, 1.5, 2, 3\}$. Consequently, when the case $\delta = 0$ occurs, all the means are equal to 0 ($m_1 = m_2 = m_3 = 0$) and the 3

Simulation x	Simulation y	nb. of shared neighbors	per. of excluded data	criterion value
0	0	28.3±16.8	0.22±0.12	40.4±8.1
1	1	23.7±6.9	0.12±0.04	37.1±2.1
1.5	1.5	22.3±15.6	0.07±0.06	34.4±7.0
2	2	26.6±5.9	0.06±0.02	19.8±2.6
3	3	14.3±9.6	0.03±0.03	9.8±1.4
4	4	6.4±6.4	0.01±0.02	7.2±0.4

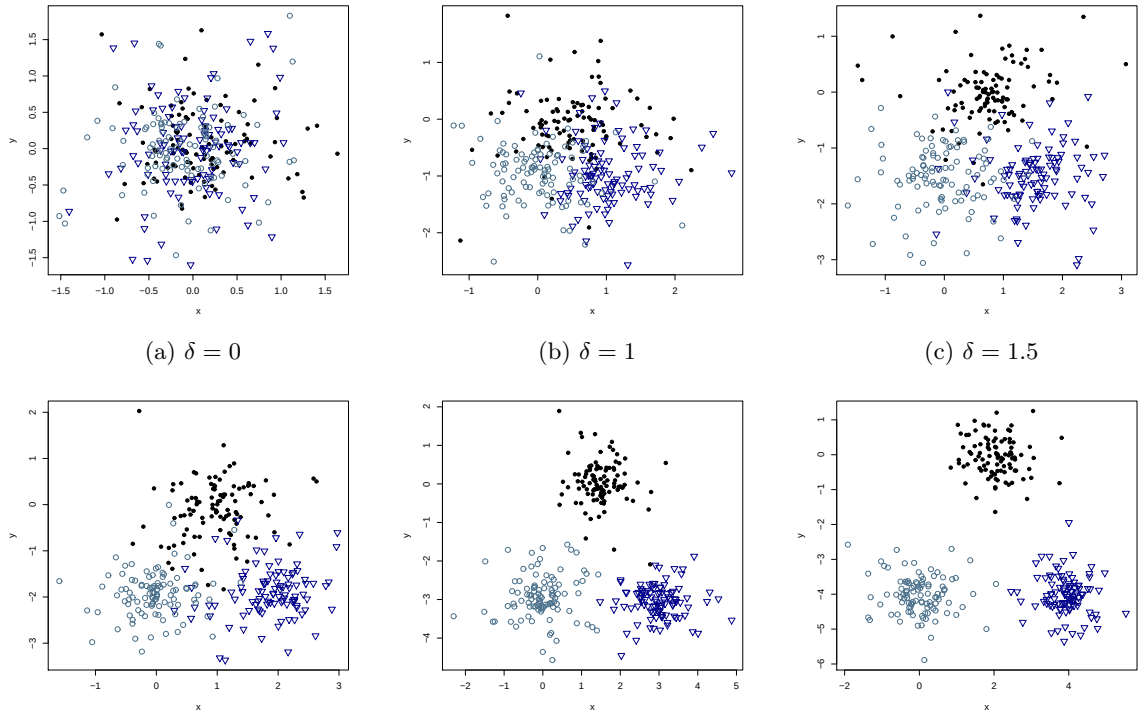
Table 9.1: Results obtained from 10 simulations.

groups are totally merged. In the opposite case, *i.e.* $\delta = 3$, the groups are well-separated and it corresponds to the situation where $m_1 = (1.5, 0)$, $m_2 = (3, -3)$ and $m_3 = (0, -3)$. The PB-Clus algorithm is executed on these different situations and each simulation is repeated 20 times.

The main results are reported in Table 9.1: the averages of the evolution of the number of shared neighbors, the percentage of excluded data and also, the compactness criterion with their standard deviations, according to the values of the overlapping parameter δ. As expected, we can observe that, according to the degree of overlapping of clusters, the compactness criterion chooses a sparser representation. In particular, smaller δ is, the more the percentage of excluded data increases. Indeed, if δ is small which depicts a situation where clusters are overlapping, then the intrinsic structure of the data becomes fuzzy which leads to exclude more and more data. The extreme cases is when the situations $\delta = 0$ or $\delta = 1$ occur. There is then no more structure in the data. Indeed, in these both cases, even though the algorithm removes many data since it represents 12% to 22% of observations, the structure of 3 clusters is not visible anymore. Such a situation is illustrated in Figures 9.5a and 9.5b, which depict the rearranged matrices obtained for $\delta = 0$ and $\delta = 1$ from one simulation. As we can observe in Figure 9.4b, the clusters are so much superimposed that they form a unique cluster. The PB-Clus algorithm does not detect any structure at all, as Figure 9.5b illustrates it. However, as a structure appears in the datasets ($\delta \geq 1.5$), then the PB-Clus algorithm enables to stress it, as in Figures 9.5c–f.

9.4 Noisy data

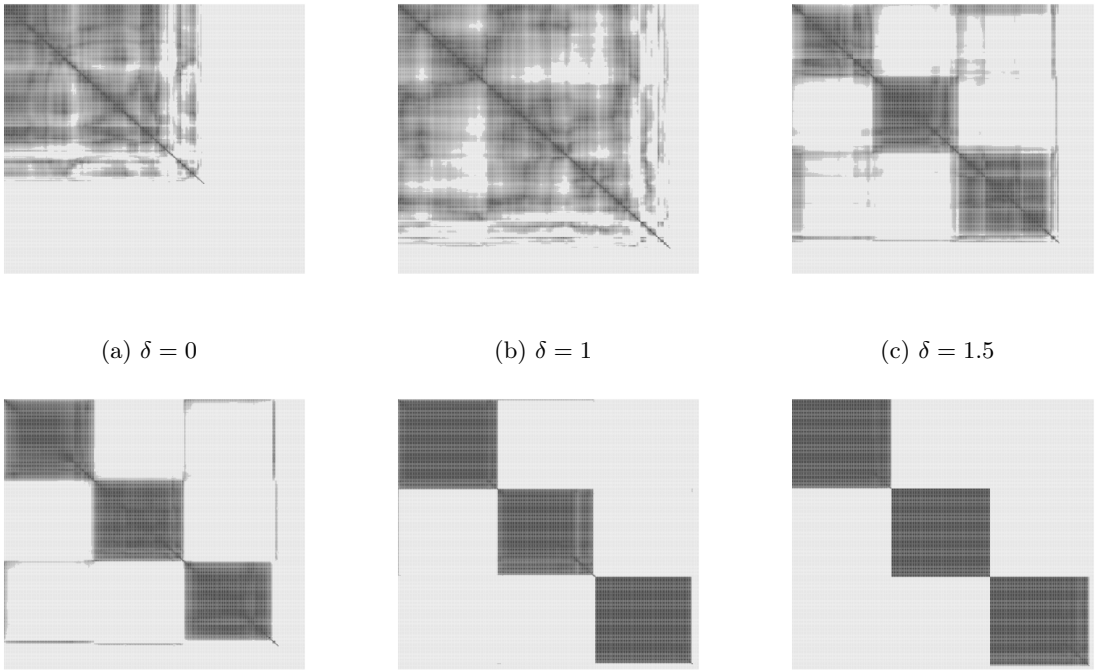
This section presents three experiments on noisy data: in the first experiment, the behavior of the PB-Clus algorithm according to noisy data is assessed. The aim of the second paragraph is to compare the results obtained by our algorithm with several existing seriation



(a) $\delta = 0$ (b) $\delta = 1$ (c) $\delta = 1.5$

(d) $\delta = 2$ (e) $\delta = 3$ (f) $\delta = 4$

Figure 9.4: Plots of the simulated datasets according to different situations of simulations.



(a) $\delta = 0$ (b) $\delta = 1$ (c) $\delta = 1.5$

(d) $\delta = 2$ (e) $\delta = 3$ (f) $\delta = 4$

Figure 9.5: Rearranged matrix according to the different situations of simulations.

methods. Finally, the last experiment deals with the efficiency of the PB-Clus agorithm when the dimension of the data increases.

9.4.1 Behavior of PB-Clus according to noisy data

In this first experiment, we assess the behavior of the PB-Clus algorithm in the case of noisy data. Three Gaussians are simulated in a 2-dimensional space and consist of 100 observations each. The cluster means are well-separated and the variance of clusters is supposed to be isotropic. The parameters used for the simulations were the following:

$$m_1 = (1.5, 0),\ m_2 = (3, -3),\ m_3 = (0, -3),$$

for the mean vectors of the three clusters and their covariance matrix has been fixed to $\Sigma = \sigma \mathbf{I}_2$ with $\sigma = 0.5$. These groups are voluntary well-separated, in order to test the efficiency of our proposed method to noisy data. Therefore, we generate noisy data according to a uniform density on an hypercube of dimension 2. In order to assess the sensitivity of the visualization to noise, the percentage of noisy data, in the simulated sample and noted δ, varies between 10% to 100% of the observed data. For each level of noise δ, we repeat the simulation twenty times. Table 9.2 stands for, respectively, the averages of the number of shared neighbors, the percentage of excluded data *i.e.* the level of sparsity, and also the minimum value of the criterion and their standard deviation.

First of all, we can observe that the more the percentage of noisy data increases, the more the sparsity level increases. Indeed, both the percentage of excluded data and the number of shared neighbors tend globally to raise with δ. The introduction of such a sparsity, in the dataset, allows to preserve a relative good visualization of the intrinsic structure. However, the quality of the visualization of the rearranged matrix decreases and this is explained by the increase of the compactness criterion.

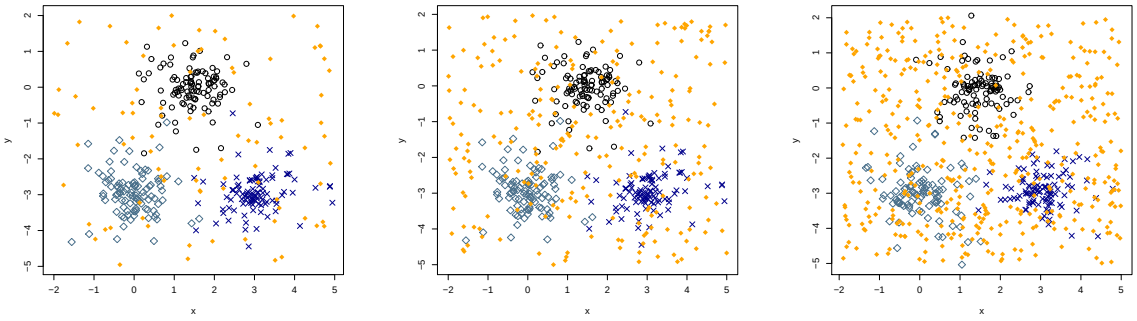
Furthermore, we represent in Figures 9.6 and 9.7 the simulated data and their associated rearranged matrix for 3 different levels of noise ($\delta = \{0.4, 0.8, 1\}$). We can observe that, admittedly the visualization deteriorates just a little, the visualization of the number of clusters remains clear. The quality of these visualizations is explained as the sparsity level increases, the number of excluded data increases. Consequently, only the strong-connected data are kept which enables to stress the intrinsic structure of the studied data.

9.4.2 A comparative study between seriation methods

The main goal of this experiment is to compare the visualization provided by the PB-Clus algorithm with those obtained by three other seriation methods. We consider a hierachical clustering algorithm (HC), an algorithm developed by Chen [39] and based on the Robinsonian property (Chen) and also an algorithm based on the anti-Robinsonian property and proposed by Brusco *et al.* [26]). These methods were already cited in Chapter 7 and are also detailed

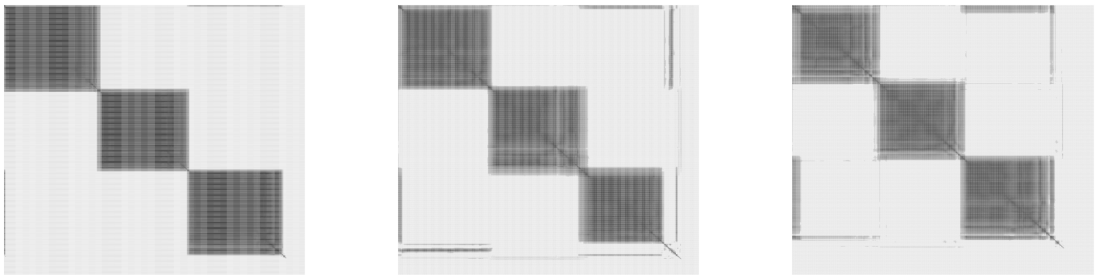
Parameter	nb. of shared neighbors	per. of excluded data	criterion value
$\delta = 0.10$	23.4±6.3	0.05±0.01	12.35±1.21
$\delta = 0.20$	27.2±8.4	0.06±0.02	14.63±2.11
$\delta = 0.40$	44.8±9.5	0.09±0.02	20.78±3.34
$\delta = 0.80$	46.4±17.3	0.13±0.04	38.02±3.48
$\delta = 1.00$	65.1±23.3	0.10±0.05	44.87±1.81

Table 9.2: Results obtained from 20 simulations.



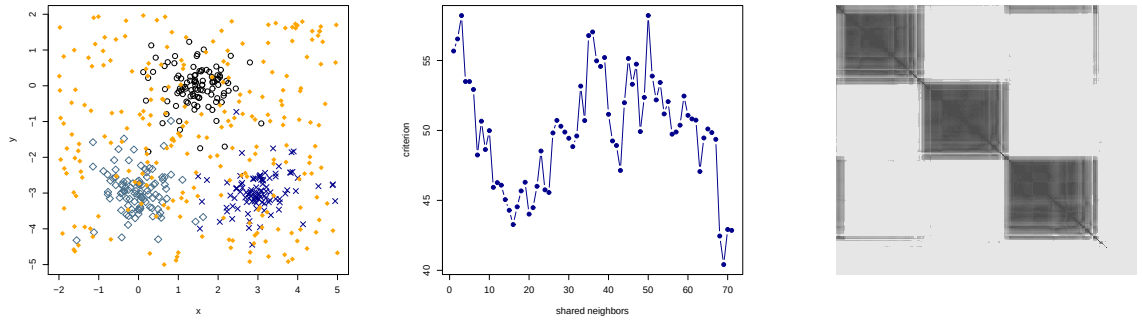
(a) 40% of noisy data. (b) 80% of noisy data. (c) 100% of noisy data.

Figure 9.6: Plots of the simulated dataset in the cases of 40%, 80% and 100% of additional noisy data.



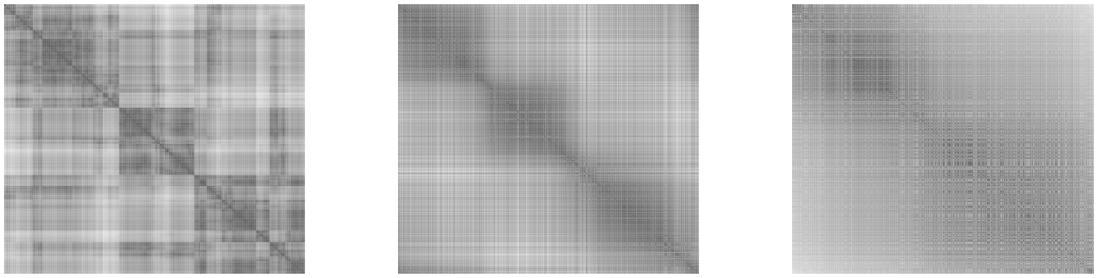
(a) 40% of noisy data. (b) 80% of noisy data. (c) 100% of noisy data.

Figure 9.7: Rearranged matrices in the cases of 40%, 80% and 100% of additional noisy data.



(a) 100% of noisy data. (b) PB-Clus criterion. (c) Rearranged matrix by PB-Clus.

Figure 9.8: Visualization of the simulated data on their 2-dimensional space (a), of the criterion of the PB-Clus algorithm selecting a level of common neighbors equal to 69 (b) and the associated rearranged matrix (c).



(a) Rearranged matrix by HC. (b) Rearranged matrix by Chen. (c) Rearranged matrix by MDS.

Figure 9.9: Rearranged matrices obtained by hierarchical clustering (HC) (a), Chen method (b) and a method based on anti-Robinsonian matrix (c) on the noisy dataset.

in [79]. The implementation of these 3 methods are available in the package 'seriation' of R (R development Core Team 2004) which is used to obtain these results.

For this experiment, we use the same simulation as previously. In particular, we consider the mixture of 3 balanced Gaussians whose parameters have been defined in the previous paragraph. Moreover, half of the dataset are noisy data. This situation is illustrated in Figure 9.8a. The evolution of the compactness criterion, according to the different level of sparsity *i.e.* the number of common neighbors, is depicted in Figure 9.8b. The best visualization, selected by the criterion, is in its minimum. This corresponds to a number of shared neighbors equal to 69. This sparsity implies that 13% of the data have been excluded and enables, in the same time, to obtain a clear representation of the structure. The visualization, associated to this best criterion, is illustrated in Figure 9.8c. From the ε-neighborhood graph, we execute three methods of seriation which are provided by the library seriation of the software R. Figure 9.9a stands for the visualization of the rearranged matrix obtained according to the HC method and Figures 9.9b and 9.9c illustrate respectively the rearranged matrix obtained by Chen's

Parameters:		nb. of shared neighbors	per. of excluded data	criterion value
$\mu = 1.6$	$p = 5$	2.7±1.3	0.009±0.02	6.61±0.72
	$p = 10$	2.6±1.1	0.011±0.02	8.69±1.32
	$p = 20$	3.1±1.1	0.025±0.04	11.05±1.57
	$p = 40$	3.4±1.6	0.042±0.06	15.62±2.06
	$p = 100$	4.4±1.7	0.086±0.08	20.04±2.71
	$p = 200$	4.6±1.4	0.109±0.06	21.39±2.85
	$p = 500$	5.5±1.0	0.145±0.05	22.84±2.61
$\mu = 2.6$	$p = 5$	2.2±0.6	0.005±0.01	4.8±0.63
	$p = 10$	2.2±0.6	0.006±0.01	4.5±0.51
	$p = 20$	2.6±1.0	0.009±0.02	5.0±0.67
	$p = 40$	2.9±1.4	0.01±0.02	5.4±0.72
	$p = 100$	2.8±1.2	0.01±0.02	8.2±1.31
	$p = 200$	3.5±1.4	0.01±0.04	12.9±1.93
	$p = 500$	4.5±1.5	0.03±0.07	18.5±3.32

Table 9.3: Means and standard deviation of the number of shared neighbors, the percentage of excluded variables and the compactness criterion obtained on 20 simulations.

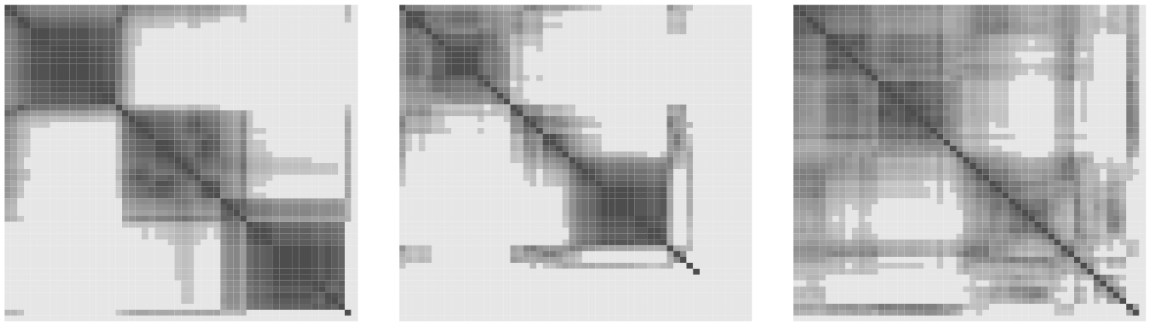
method and by Brusco's one. As we can observe, the block diagonal form, in these 3 rearranged matrices, exists particularly in the 2 first figures. However, such a structure does not appear clearly comparing to the visualization obtained by the PB-Clus algorithm in Figure 9.8c. Such a visualization is possible according to the family of sparse binary matrices. It allows, indeed, to remove most of noisy data.

9.4.3 Impact of noisy variables

In this last paragraph, we want to evaluate the behavior of the PB-Clus algorithm, in the case of high-dimensional data. Therefore, for this simulation, 3 Gaussian components of n observations each, which differ only on $q = 5$ features, are simulated in a p-dimensional observation space. In particular, each random vector Y_j conditionally to the class membership follows an univariate Gaussian density function with mean $\mu_{kj} = \mu \times (\mathbf{1}_{k=1,j\leq q}, -\mathbf{1}_{k=2,j\leq q})$ and a variance $\sigma_{kj} = 0.5$. For this experiment, the dimension p of observations varied between $p = q = 5$ to $p = 100$ and the number of obervations has been fixed to $n = 18$. Moreover, we consider two different cases: $\mu = 1.6$ and $\mu = 2.6$. Each simulation is run 20 times and the results, averaged, are presented in Table 9.3.

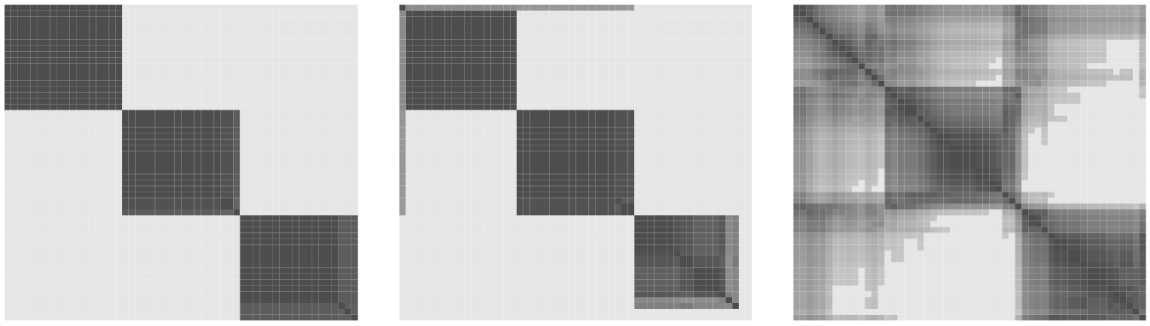
Concerning the first scenario ($\mu = 1.6$), we can observe that the compactness criterion increases very quickly with the dimension compared to the second scenario ($\mu = 2.6$). This is explained as the clusters are not well-separated, in the first simulation. Consequently, when the dimension of the observation space increases, the data structure is not strong enough to remain dominant in the dissimilarity matrix. As the dimension p becomes then higher than 10, the structure is lost whatever is the sparsity introduced in the data.

On the contrary, in the case of $\mu = 2.6$, we can observe that the increase of the compactness



(a) $p = 10, \mu = 1.6$ (b) $p = 40, \mu = 1.6$ (c) $p = 100, \mu = 1.6$

Figure 9.10: Rearranged matrices obtained amongst 20 simulations and for different values of p in the case of $\mu = 0.6$.



(a) $p = 10, \mu = 2.6$ (b) $p = 40, \mu = 2.6$ (c) $p = 100, \mu = 2.6$

Figure 9.11: Rearranged matrices obtained amongst 20 simulations and for different values of p in the case of $\mu = 0.6$.

criterion is much slower than previously which supposes that the information relative to the intrinsic structure remains strong in the dissimilarity measure. In the evolution of the criterion, a gap appears from $p = 100$ meaning that the visualization is going to deteriorate. Hence, even though $n < p$, the PB-Clus algorithm remains efficient to detect a structure as soon as the clusters are relatively well-separated.

Besides, we can visualize the effect of the dimension on the rearranged matrices. Indeed, Figures 9.10 and 9.11 present some reordered matrices obtained amongst the 20 simulations and for different values of p and μ. In particular, we can visualize the different remarks previously done: in the case $\mu = 1.6$, as the observation dimension exceeds $p = 10$, the block clusters representation become fuzzy, and the intrinsic struture does not appear clearly anymore whereas in the case $\mu = 2.6$, the intrinsic structure remains well-defined, even when $p = 100$.

dataset:	shared neighbors	compactness criterion	clustering accuracy
Lsun	35	17.47	0.93
TwoDiamonds	9	10.43	0.99
Target	6	23.11	0.98

Table 9.4: Means and standard deviation of the number of shared neighbors, the percentage of excluded variables and the compactness criterion obtained on 20 simulations.

9.5 Non Gaussian clusters

In this fourth experiment, we consider 3 different benchmark datasets coming from the FCPS repository[1] [167]:

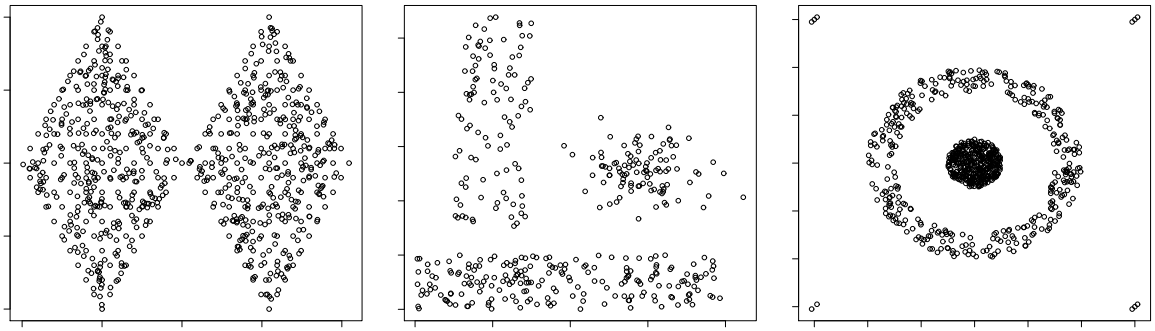
- The **Lsun** dataset is made of 3 different groups, consisting of 400 observations and described by 2 variables. This dataset is characterized by the fact that each cluster has a different within covariance matrix. Figure 9.12a stands for the plot of this dataset.
- The **TwoDiamonds** dataset contains 800 observations which are split up into 2 different clusters and described by 2 attributes. As Figure 9.12b illustrates it, the density of each cluster is non-Gaussian.
- Finally, the last dataset is the **Target data** which stands for 770 observations divided on 6 groups. The main difficulty of this dataset is that it depicts a situation of unbalanced clusters (see Figure 9.12c).

Admittedly, all these datasets are low-dimensional, but they present particularities which make the clustering task difficult. For example, in certain cases, groups of a same mixture do not follow the same distribution; others have a distance which varies within each cluster, and there are also data whose the clusters are not linearly separable.

In this experiment, the efficiency of the PB-Clus algorithm, in finding both the intrinsic structure and the partition of the data, is assessed. Table 9.4 stands for the number of shared neighbors selected by the compactness criterion, the percentage of excluded variables, the value of the compactness criterion and also a clustering accuracy computed from the true labels.

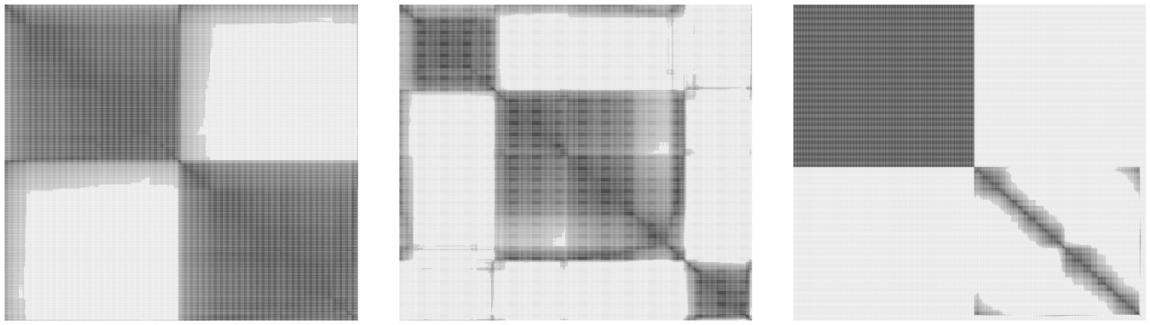
Figures 9.13 stand for the rearranged matrices selected by the compactness criterion for the 3 benchmark datasets. We can observe that, for all these data, the visualization of the intrinsic structure remains clear whatever is the studied dataset. This quality is mainly explained as the data are low-dimensional and the clusters are relatively well-separated. We can observe that the visualization of clusters, in these reordered matrices, changes a little with respect to the form of the clusters. In particular, in the case of the Target dataset, the ring in Figure 9.12c

[1]These datasets are available in the following website: *http://www.informatik.uni-marburg.de/fb12/databionics.*



(a) Two diamands dataset. (b) Lsun dataset. (c) Target dataset.

Figure 9.12: Plots of 3 FCPS datasets.



(a) Two diamands dataset. (b) Lsun dataset. (c) Target dataset.

Figure 9.13: Rearranged matrices obtained by the PB-Clus algorithm and associated to 3 FCPS datasets.

is represented by an oblong form in Figure 9.13. However, in this same figure, we can note a limitation of the PB-Clus algorithm on unbalanced data. In particular, the 4 little clusters are not taken into account in the visualization since the selected number of share neighbors is higher than the number of observations in each cluster ($n = 3$).

9.6 Comparison with seriation algorithms

The aim of this last experiment is to compare the efficiency of the PB-Clus algorithm in terms of visualization and classification, with 2 other methods already introduced previously: the hierarchical clustering in a first hand, and the Chen's method based on an anti-Robinsonian property [39] in a second hand. This comparison is made on 4 traditional datasets which are available in the R software. The 4 datasets considered in this experiment are:

- The **iris** dataset which is made of 3 different groups of irises (*setosa*, *virginica* and *versicolor*) and described by 4 variables. This dataset is described in detail in Section 5.1. Besides, these data are very interesting as most of clustering algorithms do not usually

			PB-Clus		HC		AR	
data	nb.	compactness	Moore	Neumann	Moore	Neumann	Moore	Neumann
Iris	2	8.67	1371.2	471.1	31728.8	10893.1	19357.8	7304.0
Ruspini	3	5.92	1290.1	442.2	8724.9	3036.4	6503.7	2277.1
Geyser	9	9.50	2514.9	850.4	68205.3	2302.1	12866.8	4501.4
Faithful	2	12.9	2634.1	889.4	34045.5	11503.5	23390.0	9894.2
Township	1	5.14	244.5	91.8	1109.9	441.5	849.0	342.0

Table 9.5: Seriation criteria (Moore, Neumann) computed for the PB-Clus, Hierarchical Clustering (HC) algorithms and a seriation method based on an anti-Robinsonian (AR) property on 5 benchmark datasets. The value of compactness criterion and its associated number of shared neighbors (nb.) obtained by PB-Clus are also reported.

select an intrinsic structure to 3 clusters, but only to 2 clusters. This is due to the difficulty to distinguish the *virginica* and *versicolor* irises.

- The **ruspini** dataset comes from the works of Ruspini [154] on clustering: it consists of 75 datapoints described on 2 dimensions. It is made of 4 groups which are easy to cluster.

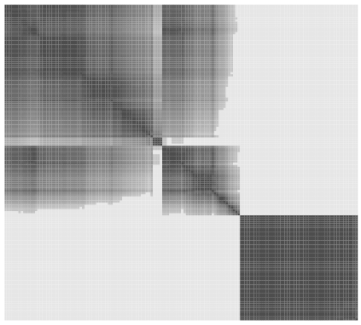
- The **faithful** data come from the works of [78] and represent the times and the duration between 2 geyser eruptions, of the national parc of Yellowstone (Wyoming - USA). This dataset consists of 272 observations and 2 classes.

- The **Geysers** datasets stand for a complete version of the previous data collected by [7]. In this dataset, 299 eruptions are studied between the 1st and the 15th August 1985. They are described by the same features as previously.

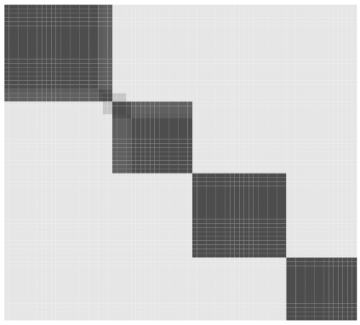
- The **township** dataset stands for a boolean matrix in which 16 cities are described from the presence or absence of 9 characteristics such as universities, police station, railway station, *etc.*

For the 4 first datasets, we compute the dissimilarity matrices by using the Euclidean distance. Besides, as the Township dataset is already in a $(0,1)$-table, we directly create the shared neighbors matrix by considering the matrix TT^t where $T \in \mathbb{R}^{16\times 9}$ stands for the township dataset.

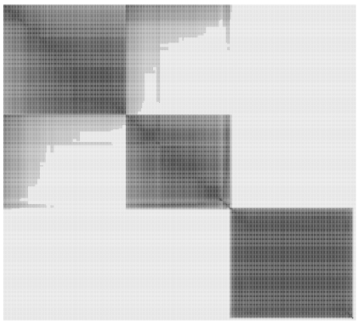
In order to evaluate the quality of the visualization obtained by the PB-Clus algorithm, we use two other criteria defined by Neumann and Moore, in addition to the compactness criterion. Both criteria, already introduced in Chapter 7, are based on the close neighborhood and have to be minimized. Moreover, the Moore and Neumann criteria are computed for two other methods of seriation available in the package seriation of R software: the hierarchical clustering (HC) approach and a method based on the anti-Robinsonian property of a matrix (AR). These results are reported in Table 9.5 for each criterion computed on each dataset. The number of shared neighbors and the compactness criterion provided by the PB-Clus



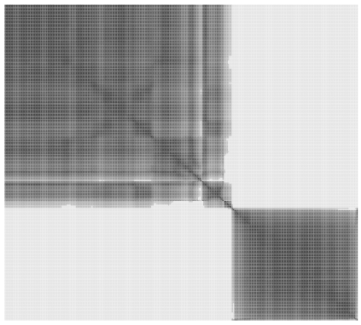
(a) Fisher's irises.



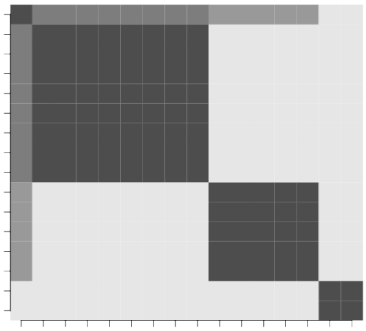
(b) Ruspini.



(c) Geyser.

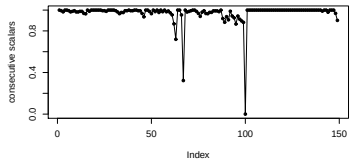


(d) Faithful.

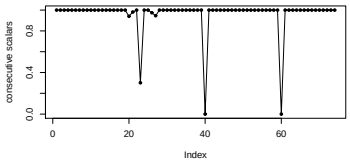


(e) Townships.

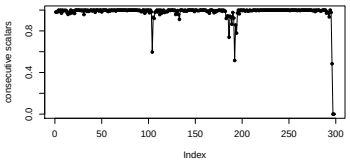
Figure 9.14: Rearranged matrices obtained with the PB-Clus algorithm.



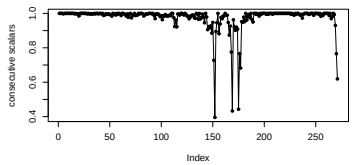
(a) Fisher's irises.



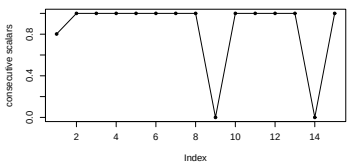
(b) Ruspini.



(c) Geyser.



(d) Faithful.



(e) Faithful.

Figure 9.15: Consecutive scalars resulting from the rearranged matrices.

algorithm are also reported in this table. However, the percentage of excluded observations is not reported in the table, as none observation has been removed. Besides, we can observe that the Moore and Neumann criteria are both minimum with the PB-Clus algorithm, for the 5 datasets. This suggests a certain quality of the visualizations obtained by the PB-Clus algorithm compared to the two other methods. The improvement of the Moore and Neumann criteria, in the case of PB-Clus algorithm, is mainly explained by the use of sparse dissimilarity matrices, contrary to the HC and AR methods. Indeed, the measure based on the common neighbors introduces many zeros, in the studied dissimilarity matrix, which makes clearer its visualization.

Furthermore, Figures 9.14 stand for the rearranged matrices obtained by the PB-Clus algorithm. As we can observe, the structure of each dataset appears clearly. In particular, it is interesting to notice that, in the case of the irises data, a 3-group structure appears clearly whereas on this particular dataset, most of existing clustering algorithms select a number of groups equal to 2. It is particularly true for the models based on Gaussian mixtures, as it is developed in [148] for example. For the 4 other datasets, the block-representation of each reordered matrix is very stressed. This suggests that the clusters are relatively compact and well-separated in the considered datasets. This is highlighted by the fact that the number of shared neighbors, selected by the PB-Clus algorithm, is very low, as it has been already noted in Table 9.5.

Finally, we also report, in this table, the clustering accuracies of the partition obtained by the PB-Clus algorithm. In particular, this rate is computed using the true labels when they are known and in the opposite case, with the labels estimated by the k-means algorithm. Since, this last method supposes the knowledge of the number of groups, we choose to use the number of clusters detected by the PB-Clus algorithm, in order to obtain comparable partitions. The partition of the PB-Clus algorithm is obtained according to the consecutive scalar between adjacent pairwises of observations. The consecutives scalars can represent the similarity between adjacent pairwises of observations and they can also indicate when two adjacent observations do not belong to the same cluster. To that end, the consecutive scalars of rearranged matrix in Figures 9.14 are presented in Figures 9.15. As we can observe, the breakings between two clusters are really well emphasized which has ease the building of partitions.

Finally, Table 9.5 stands for the clustering accuracies: they are computed according to the true label in the case of the Fisher's irises and according to the partition obtained by the k-means algorithm for the other datasets. As we can observe, on the Fisher's irises, the PB-Clus algorithm mis-classifies 12% of the data. This high error rate is mainly explained by the non-separability between the versicolor and virginica species. Indeed, as we can observe in Table 9.5a, several virginica are classified in the versicolor class and this corresponds to the 2 overlapped block-clusters illustrated in Figure 9.14a. Concerning the 4 other datasets, we can note that the partitions obtained by the PB-Clus and the k-means algorithms perfectly

	1	2	3
Versicolor	50	17	0
Virginica	0	33	0
Setosa	0	0	50
Error rate:	*0.12*		

(a) Partition estimated by the PB-Clus algorithm on the Fisher's Irises.

	1	2	3	4
k-means 1	13	0	0	0
k-means 2	0	35	0	0
k-means 3	0	0	15	0
k-means 4	0	0	0	20
Error rate:	*0.00*			

(b) Partition estimated by the PB-Clus algorithm on the Ruspini's data.

	1	2	3
k-means 1	88	2	7
k-means 2	0	105	0
k-means 3	0	0	97
Error rate:	*0.03*		

(c) Partition estimated by the PB-Clus algorithm on Geyser data.

	1	2
k-means 1	168	4
k-means 2	0	100
Error rate:	*0.02*	

(d) Partition estimated by the PB-Clus algorithm on Faithful dataset.

	1	2	3	4
k-means 1	8	0	0	0
k-means 2	0	4	0	0
k-means 3	0	0	2	0
k-means 4	0	1	0	1
Error rate:	*0.06*			

(e) Partition estiamted by the PB-Clus algorithm on the Townships data.

Table 9.6: Partitions estimated by the PB-Clus algorithms for the 5 datasets and cross-validated with either the true labels (irises) or a k-means' partition.

match as the difference rates are less than 3%.

Part III

Application

Chapter 10

Application to cervical cancer detection

This part is not included in the public version because of confidential results.

Chapter 11

Conclusion

11.1 Overview of the contributions

In a first part, we have introduced a new family of probabilistic models which both clusters the data and finds a discriminative subspace chosen such as it best discriminates the groups. This method, named the discriminative latent mixture (DLM) model, aims to find a parsimonious and discriminative fit for the data in order to ease the clustering and the visualization of the clustered data in a mixture model context. This family of models is based on two key-ideas which assume that firstly, actual data live in a latent subspace with an intrinsic dimension lower than the dimension of the observed space and, secondly, a subspace of $K-1$ dimensions is theoretically sufficient to discriminate K groups. We also proposed an estimation procedure named Fisher-EM which improves, most of the time, clustering performance owing to the use of a discriminative subspace. We showed that the Fisher-EM algorithm is an EM like algorithm in the case of the $\text{DLM}_{[\alpha\beta]}$ model. Besides, it appears nevertheless that the interpretation of the estimate discriminative axes is *a priori* quite difficult since each axis of the discriminative subspace is a linear combination of all original variables. The understanding of axes could be facilitated if only some loadings in each discriminant axis were selected. We therefore proposed 3 different methods based on a penalized criterion which enables to introduce sparsity directly in the loadings of the projection matrix. It implies that, in addition to produce sparse loadings, the penalty terms enables to make variable selection.

In a second part, we proposed in the seriation context, a dissimilarity measure based on a common neighborhood which enables to introduce different degrees of sparsity in a dissimilarity matrix. This dissimlarity measure is based on the fact that, the more the number of common neighbors between two pairs of data is high, the more these observations are similar. From the common neighborhood matrix, a collection of binary matrices is constructed and each of them corresponds to different degrees of sparsity. The main interest of this approach is that the collection of binary matrices which refers to different degrees of sparsity λ becomes very sparse as the λ-value increases. In particular, the more the λ-level is high, the more the

associated matrix (λ-matrix) is filled up with zeros, removing thus column and row vectors of the study. Consequently, the introduction of such a family enables to tackle the problem of highly noisy data and overlapping, in the seriation case. Moreover, we proposed a forward-stepwise seriation algorithm, called the PB-Clus algorithm, which rearranges the matrices such as the adjacent rows, and symetrically columns, of the family of λ-matrices are the most similar. A family of sparse rearranged matrices is created and the one which presents the best block diagonal form is selected, according to a compressing criterion. This tool enables to both identify clusters even in the case of noisy data, outliers, overlapping and non-Gaussian groups, and reveals the intrinsic structure of data.

Finally, the last part presented an application on a biological dataset in which both approaches have been applied. The dataset, provided by the Novacyt company, consisted of cervical cells coming from different smears, and was described by 42 morphological and photometrical variables. The main aim of this application was to both cluster the cells in 2 different classes (normal and atypical cells) and find a subset of variables which describe and best-discriminate the atypical cells. The sparse Fisher-EM procedure was used, in one hand, to cluster and select discriminative variables; in an other hand, the PB-Clus algorithm was implemented to check the relevance of the obtained partition. As a result, the discrimination of atypical nuclei was mainly operated by 4 morphological variables and 3 photometrical ones. In addition, this subset of variables allowed to improve the detection of pathological cells.

11.2 Works in progress

11.2.1 Supervised and semi-supervised versions of the Fisher-EM algorithm

In the supervised classification framework, learning a classifier requires the knowledge of labels in the dataset and this supposes that the labels of the learning set are true. However, it appears that in many applications, the labeling task is made by humans and their expertise can be difficult, expensive or sometimes imprecise, leading to mistakes in the labels. Consequently, when some labels are wrong, since most of existing methods give a full confidence to the labels of the learning dataset, the classification prediction gives poor results. For example, in the case of the traditional FDA, recurring problems occur such as its sensitivity to label noise or to sparse labels.

We then propose to rewrite the DLM, that we have defined in an unsupervised context, in a supervised and a semi-supervised contexts in order to both deal with label noise data and/or sparse labels. On preliminary simulations, it appears that a supervised version of the DLM model presents very promising results regarding the robustness to label noise. In particular, let us consider the following introductory example on Irises dataset: let τ the percentage of false labels in the learning set which varies between 0 and 0.9. At each trial, the Irises dataset

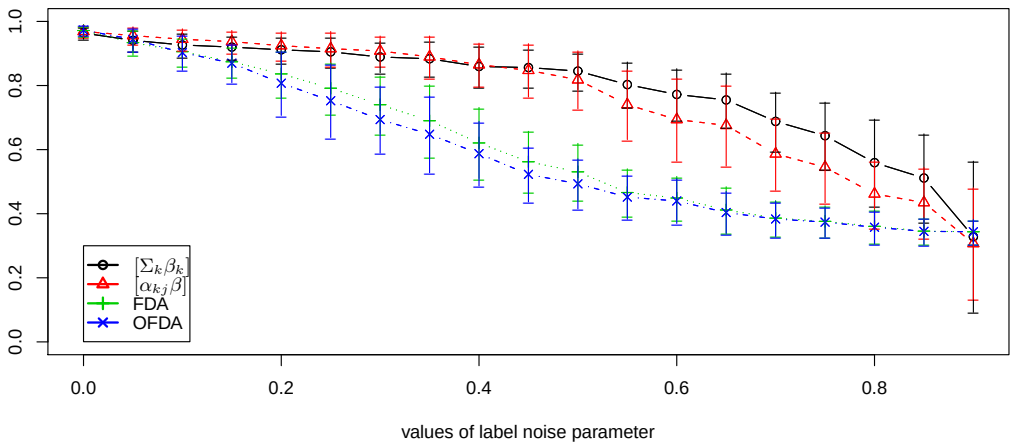


Figure 11.1: Effect of label noise in the learning dataset on the prediction effectiveness: correct classification rate according to the percentage of noisy labels.

is randomly divided into 2 balanced samples: a learning set in which a percentage τ of the data is mislabeled and a test set on which the prediction performances are evaluated. This process has been repeated 50 times for each value of τ in order to monitor both the average performances and their variances. The predictive performances have been assessed on each test set and then averaged. Besides, we have compared 2 DLM models of the supervised approach with the traditional ones such as FDA and its orthonomalized version (OFDA) [80].

Figure 11.1 presents the evolution of correct classification rate computed on the test set for the 4 methods according to τ. As we can observe, our approach appears more robust to label noise than FDA and OFDA. Indeed, the correct classification rates of the DLM models remain larger than 0.8 for a label noise up to $\tau = 0.6$. Conversely, the FDA and OFDA methods have their classification rates which lower drastically and linearly as $\tau = 0.15$ and from $\tau = 0.7$, their prediction performances are comparable to those of a random classifier. These improvements can be explained by the probabilistic framework of the DLM models which takes into account an error term and this avoids to overfit the embedding space on the labeled data and remains generally enough to be robust on label noise contrary to FDA and OFDA.

11.2.2 Convergence in the heteroscedastic case

In this manuscript, the convergence of the Fisher-EM algorithm was proved only for the isotropic case of the $\text{DLM}_{[\alpha\beta]}$ model. In order to prove that the Fisher-EM algorithm is a GEM algorithm for the eleven DLM models, we need to show that the quantity $\Delta(U^{(q+1)}, \theta^{(q+1)} \mid U^{(q)}, \theta^{(q)}) = Q(U^{(q+1)}, \theta^{(q+1)}) - Q(U^{(q)}, \theta^{(q)})$ is positive, with $\theta^{(q)}$ the set of parameters of the DLM models estimated at iteration q, $U^{(q)}$ the projection matrix of the latent subspace and $Q(\theta)$ the expectation of the complete log-likelihood.

The main step to obtain such a result, is to prove that in the F-step, the inequality $Q\left(U^{(q+1)}, \theta^{(q)}\right) \geq Q\left(U^{(q)}, \theta^{(q)}\right)$ holds. By considering, in first, that in the current iteration q, we dispose of the estimates $\hat{U}^{(q)}$, $\hat{\theta}^{(q)} = \left\{\hat{\mu}^{(q)}, \hat{\mathbf{\Sigma}}^{(q)}, \hat{\beta}^{(q)}, \hat{\pi}^{(q)}\right\}$ of all the parameters of the DLM model and we suppose that we dispose of the posterior probabilities $t_{ik}^{(q+1)}$ computed in the E step at the following iteration $q+1$. Let us then introduce the quantity:

$$\Delta_1 = \Delta\left(\hat{U}^{(q+1)}, \hat{\theta}^{(q)} \mid \hat{U}^{(q)}, \hat{\theta}^{(q)}\right) = Q(\hat{U}^{(q+1)}, \hat{\theta}^{(q)}) - Q(\hat{U}^{(q)}, \hat{\theta}^{(q)}).$$

where the $Q\left(\hat{U}^{(q)}, \hat{\theta}^{(q)}\right)$ computed at the current iteration q and conditionally to the $q+1$th E-step is:

$$\begin{aligned} Q(\hat{U}^{(q)}, \hat{\theta}^{(q)}) = -\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}^{(q+1)}\Big[-2\log(\hat{\pi}_k^{(q)}) + \operatorname{trace}\left(\left(\hat{\Sigma}_k^{(q)}\right)^{-1}\hat{U}^{(q)t}C_k^{(q+1)}\hat{U}^{(q)}\right) + \log\left|\hat{\Sigma}_k^{(q)}\right| \\ + (p-d)\log(\hat{\beta}_k^{(q)}) + \frac{1}{\hat{\beta}_k^{(q)}}\operatorname{trace}(C_k^{(q+1)} - \hat{U}^{(q)t}C_k^{(q+1)}\hat{U}^{(q)}) + \gamma\Big], \end{aligned}$$

and:

$$\begin{aligned} Q(\hat{U}^{(q+1)}, \hat{\theta}^{(q)}) = -\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}^{(q+1)}\Big[-2\log(\hat{\pi}_k^{(q)}) + \operatorname{trace}\left(\left(\hat{\Sigma}_k^{(q)}\right)^{-1}\hat{U}^{(q+1)t}C_k^{(q+1)}\hat{U}^{(q+1)}\right) \\ + \log\left|\hat{\Sigma}_k^{(q)}\right| + (p-d)\log(\hat{\beta}_k^{(q)}) + \frac{1}{\hat{\beta}_k^{(q)}}\operatorname{trace}(C_k^{(q+1)} - \hat{U}^{(q+1)t}C_k^{(q+1)}\hat{U}^{(q+1)}) + \gamma\Big], \end{aligned}$$

where $\gamma = p\log(2\pi)$ is a constant term. We recall that $C_k^{(q+1)}$ stands for the empirical covariance matrix of the kth group computed at iteration $q+1$ and $\hat{\Sigma}_k^{(q)}$, respectively $\hat{\beta}_k^{(q)}$, stands for the estimation of the maximum likelihood of the covariance matrix of the kth group in the latent subspace, respectively in its orthogonal complement at iteration q:

$$\Delta_1 = \frac{1}{2}\left[\sum_{k=1}^{K}\operatorname{trace}\left(B_k^{(q)}\left(A_k^{(q)} - A_k^{(q+1)}\right)\right)\right],$$

where:

$$\begin{aligned} A_k^{(q)} &= \hat{U}^{(q)t}n_k^{(q+1)}C_k^{(q+1)}\hat{U}^{(q)} \\ A_k^{(q+1)} &= \hat{U}^{(q+1)t}n_k^{(q+1)}C_k^{(q+1)}\hat{U}^{(q+1)} \\ B^{(q)} &= \hat{\Sigma_k}^{(q)^{-1}} - \frac{1}{\hat{\beta}_k^{(q)}}\mathbf{I}_d. \end{aligned}$$

A first issue remains in the fact that the quantity $B_k^{(q)} = \hat{\Sigma}_k^{(q)^{-1}} - 1/\hat{\beta}_k^{(q)}\mathbf{I}_d$ is dependent of k. The second issue is that the Fisher's criterion is an average criterion. This implies that, *a priori*, we dispose of no guarantee that $\forall k \in \{1, \ldots, K\}$ the condition $\operatorname{trace}(A_k^{(q+1)}) \leq \operatorname{trace}(A_k^{(q)})$

holds. Consequently the sign of this quantity:

$$\sum_{k=1}^{K} \text{trace}\left(B_k^{(q)}\left(A_k^{(q+1)} - A_k^{(q)}\right)\right), \tag{11.2.1}$$

remains unknown. One possible way, to consider the Fisher-EM algorithm as a GEM algorithm for the family of the 12 DLM models, is to modify the Fisher's criterion. In particular, instead of considering an average criterion, as the traditional Fisher's one, we could switch the within covariance matrix by the maximum of the within-class pairwise distances over all classes, as Zhang and Yeung [191] proposed it, very recently, in the supervised case. In this case, the associated Fisher's criterion is not anymore a weighted sum of K covariance matrices as it is based on one distance only. The matrix $A_k^{(q+1)} - A_k^{(q)}$ becomes therefore independent of k leading to consider the following quantity $\text{trace}\left(\left(A^{(q+1)} - A^{(q)}\right)\sum_{k=1}^{K} B_k^{(q)}\right)$. Under some conditions on $\sum_{k=1}^{K} B_k^{(q)}$, the convergence for all the family of DLM models could be perhaps stated.

11.3 Prospects

11.3.1. Model selection criteria for sparse clustering

In the GMM context, different works combining variable selection and clustering have been based on the introduction of penalties in the log-likelihood function. These penalties depend on an hyper parameter which stands for the level of sparsity. Several authors, who worked on ℓ_1-penalized log-likelihood function in the GMM context ([142, 184, 63, 185]), used the penalized BIC to select the hyperparameter, by evaluating the model complexity in regard to the non-zero values. Although Zou *et al.* [195] showed that the number of non-zero coefficients is an unbiased estimate of the degrees of freedom and is asymptotically consistent in the case of penalized regression problem. Nevertheless, this result is not *a priori* true in a penalized GMM context and no theoritical justification was made by its users ([63, 142, 184, 185]) in the penalized GMM context. It would be interesting to obtain theoritical guarantees of such a result in the penalized GMM context. In the same way, since ICL is also used to select the number of components, it would be a natural extension to consider a penalized ICL in regard to the penalized BIC. Moreover, since the main difference between BIC and ICL criteria remains in the presence of an additional entropy term which favors well-separated clusters in the ICL criterion, the ICL criterion would seem to be more adapted than BIC to select the sparsity level.

11.3.2. Visualization in a 2-dimensional space

The Fisher-EM algorithm enables to find a discriminative and a common low-dimensional subspace for all groups of the mixture. It is easy to project and visualize the clustered data

into the estimated discriminative latent subspace if $K < 4$. In a first hand, if the estimated value of d is at most equal to 3, the data can be visualized by projecting them onto the d first discriminative axes and no discriminative information loss is to be deplored in this case. In a second hand, if the estimated value of d is strictly larger than 3, the visualization becomes much more difficult even though it is still possible to project the data in the 3 first discriminative axes. However, even though these 3 first axes are the most discriminative ones among the provided axes, an information loss can occur. To that end, it would be interesting to work on this visualization problems. In particular, conversely to the existing methods such as VAT [12], MDS [21] or t-sne [168] which provides a visualization of data according to a pairwise similarity measure, we would be interested to work on a 2-dimensional visualization of the data by using the characteristics given by our framework (proportions, means and covariance matrices of clusters), or more generally from the GMM context.

11.3.3. Non-linear extension of the Fisher-EM algorithm

We proposed, in this manuscript, the Fisher-EM algorithm which simulatenously reduces the dimension of continuous data and clusters them. A linear projection matrix is estimated such as its column vectors span the discriminative latent subspace. We envisage to reformulate the linear problem of the Fisher's criterion with a kernel one. One of the main asset of such an extension would be to allow the clustering of categorical data, which occurs very often in the biological field or for text classification. In particular with the use of a string kernel, it would allow us to handle sequencial data.

Appendix A

List of publications

Journal articles

Bouveyron C., Brunet C., *Simultaneous model-based clustering and visualization in the Fisher discriminative subspace*, Statistics and Computing, 2011 (in Press).
Brunet C., Villman T.,Vigneron V., *Une famille de matrices sparses pour une modélisation multi-échelle par blocs*, Revue des Nouvelles Technologies de l'Information, 2011 (in Press).

Refeered international conferences

Bouveyron C., Brunet C., *Discriminative variable selection for clustering with the sparse Fisher-EM algorithm,* 14th International Conference on applied Stochastic Models and Data Analysis, Roma, Italy, 2011.
Bouveyron C., Brunet C., *Probabilistic Fisher Discriminant Analysis,* 19th European Symposium on Artificial Neural Networks, Belgium, 2011.
Bouveyron C., Brunet C., *Clustering in Fisher Discriminant Subspace*. Proceedings of the XIIIth International Conference on Applied Stochastic Models and Data Analysis, Lituania, 2009.
Brunet C., *Probabilistic classification for cervical cancer detection*, 8th International Workshop on Operations Research, Havana, 2009.
Bouveyron C., Brunet C., Vigneron V., *Classification of high-dimensional data for cervical cancer detection*, 17th European Symposium on Artificial Neural Networks, Belgium, 2008.

Refeered national conferences

Bouveyron C. and Brunet C., Clustering et visualisation dans le sous-espace discriminant de Fisher : quelques avancées récentes, 43èmes Journées de Statistique de la Société Française de Statistique, Tunis, Tunisie, 2011.
Brunet C., Villman T., Vigneron V., *Une famille de matrices sparses pour une modélisation par blocs*, Actes de la 12ème conférence francophone de l'apprentissage automatique, CAp2010, France, 2010.

Bouveyron C., Brunet C., *Classification automatique dans le sous-espace discriminant de Fisher*. 41èmes Journées de Statistique, SFdS, Bordeaux, France, 2009.

Bibliography

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high-dimensional data for data mining application. In *ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

[2] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[3] E. Anderson. The irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

[4] M. Ankerst, M.M. Breunig, H-P. Kriegel, and J. Sander. Optics : Ordering points to identify the clustering structure. *ACM SIGMOD*, 1999.

[5] D.L. Applegate, R.E. Bixby, V. Chvàtal, and W.J. Cook. *The Traveling Salesman Problem*. Princeton University Press, 2007.

[6] P. Arabie and L.J. Hubert. The bond energy algorithm revisited. *IEEE transactions on systems, man, and cybernetics*, 20(1):268–274, 1990.

[7] A. Azzalini and A.W. Bowman. A look at some data on the old faithful geyser. *Applied Statistics*, 39:357–365, 1990.

[8] J. Baek, G. McLachlan, and L. Flack. Mixtures of Factor Analyzers with Common Factor Loadings: Applications to the Clustering and Visualisation of High-Dimensional Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–13, 2009.

[9] J. Banfield and A. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821, 1993.

[10] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[11] J.D. Benfield and A.E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821, 1993.

[12] J.C. Bezdek and R.J. Hathaway. VAT: A tool for visual assessment (cluster) tendency. *Neural Networks*, pages 2225–2230, 2002.

[13] G. Biau, B. Cadre, and B. Pelletier. A graph-based estimator of the number of clusters. *ESIAM: Prob. and Stat.*, 11:272–280, 2007.

[14] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2001.

[15] C. Biernacki, G. Celeux, and G. Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics and Data Analysis*, 41:561–575, 2003.

[16] C. Biernacki, G. Celeux, G. Govaert, and F. Langrognet. Model-based cluster and discriminant analysis with the mixmod software. *Computational Statistics and Data Analysis*, 51:587–600, 2006.

[17] C. Bishop. *Pattern recognition and machine learning.* Springer, New York, 2006.

[18] C. Bishop and M. Svensen. The Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, 1998.

[19] C. Bishop and M.E. Tipping. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[20] K.S. Booth and G.S. Lueker. Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms. *Journal of Computer System Science*, 1976.

[21] I. Borg and P. Groenen. *Modern Multidimensional Scaling: theory and applications (2nd ed.).* New York: Springer-Verlag, 2005.

[22] C. Bouveyron, S. Girard, and C. Schmid. High-Dimensional Data Clustering. *Computational Statistics and Data Analysis*, 52(1):502–519, 2007.

[23] C. Bouveyron, S. Girard, and C. Schmid. High Dimensional Discriminant Analysis. *Communications in Statistics : Theory and Methods*, 36(14):2607–2623, 2007.

[24] G.W. Brainerd. The place of chronological ordering in archeological analysis. *Am Antiq*, 16(4):301–313, 1951.

[25] M.R. Brito, E.L. Chavez, A.J. Quiroz, and J.E. Yukich. Connectivity of the mutual k-nearest neighbor graph in clustering and outlier detection. *Statist. Probab. Lett.*, 35:33–42, 1997.

[26] M. Brusco, H. F. Kohn, and S. Stahl. Heuristic implementation of dynamic programming for matrix permutation problems in combinatorial data analysis. *Psychometrika*, 2008.

[27] M. Brusco and S. Stahl. *Branch and Bound applications in combinatorial data analysis.* Springer, 2005.

[28] M. Brusco and D. Steinley. Inducing a blockmodel structure of two-mode binary data using seriation procedures. *Journal of Mathematical Psychology*, 50:468–477, 2006.

[29] J. Cadima and Ian T. Jolliffe. Loadings and correlations in the interpretation of the principal components. *Journal of Applied Statistics*, 22:203–214, 1995.

[30] B.S. Caffo, W.S. Jank, and G.L. Jones. Ascent-based Monte Carlo EM. *Journal of the Royal Statistical Society, Series B*, 67:235–252, 2005.

[31] G. Caraux and S. Pinloche. Permutmatrix: a graphical environment to arrange gene expression profiles in optimal linear order. *Bioinformatics*, 21(7), 2005.

[32] D.J. Carroll and P. Arabie. Multidimensional scaling. *Annu Rev Psychol*, 31:607–649, 1980.

[33] R. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):145–276, 1966.

[34] G. Celeux and J. Diebolt. The SEM algorithm: a probabilistic teacher algorithm from the EM algorithm for the mixture problem. *Computational Statistics Quaterly*, 2(1):73–92, 1985.

[35] G. Celeux and G. Govaert. A Classification EM Algorithm for Clustering and Two Stochastic versions. *Computational Statistics and Data Analysis*, 14:315–332, 1992.

[36] G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28:781–793, 1995.

[37] G. Celeux, M.-L. Martin-Magniette, C. Maugis, and A. Raftery. Letter to the editor. *Journal of the American Statistical Association*, 106(493), 2011.

[38] W.C. Chang. On using principal component before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistical Society, Series C*, 32(3):267–275, 1983.

[39] C.H. Chen. Generalized association plots: Information visualization via iteratively generated correlation matrices. *Statistica Sinica*, 12:7–29, 2002.

[40] D.A. Clausi. K-means iterative fisher (KIF) unsupervised clustering algorithm applied to image texture segmentation. *Pattern Recognition*, 35:1959–1972, 2002.

[41] L. Clemmensen, T. Hastie, and B. Ersboll. Sparse discriminant analysis. Technical report, Department of Informatics and Mathematical Modelling Technical University of Denmark, 2008.

[42] G.A. Croes. A method for solving the traveling-salesman problems. *Operations Research*, 6:791–812, 1958.

[43] A. Cuevas, M. Febrero, and R. Fraiman. Estimating the number of clusters. *Can. J. Statist*, 28:367–382, 2000.

[44] A. Dasgupta and A.E. Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93:294–302, 1998.

[45] A. Dempster, N. Laird, and D. Robin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[46] E. Diday. Orders and overlapping clusters in pyramids. *In: De Leeuw J, Heiser WJ, Meulman JJ, Critchley F (eds) Multidimensional data analysis. DSWO Press, Leiden*, pages 201–234, 1986.

[47] C. Ding and T. Li. Adaptative dimension reduction using discriminant analysis and k-means clustering. *ICML*, 2007.

[48] R. Duda, P. Hart, and D. Stork. *Pattern classification*. John Wiley & Sons, 2000.

[49] W.L. Eastman. Linear programming with pattern constraints. Master's thesis, Ph.D. Dissertation, Harvard University, Cambridge, Mass., 1958.

[50] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32:407–499, May 2004.

[51] L. Ertoz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes and densities in noise. *Second SIAM international conference on data mining, Arlington*, 2002.

[52] M. Ester, H-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovery clusters in large spatial databases with noise. *Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[53] B. Fichet. Sur une extension de la notion de hiérarchie et son équivalence avec quelques matrices de robinson. *Actes des Journées de Statistique de la Grande Motte*, 1984.

[54] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[55] D.H. Foley and J.W. Sammon. An optimal set of discriminant vectors. *IEEE Transactions on Computers*, 24:281–289, 1975.

[56] M. Forina, C. Armanino, M. Castino, and M. Ubigli. Multivariate data analysis as a discriminating method of the origin of wines. *Vitis 25*, pages 189–201, 1986.

[57] C. Fraley. Algorithms for model-based Gaussian Hierarchical Clustering. *SIAM Journal on Scientific Computing*, 20:270–281, 1998.

[58] C. Fraley and A. Raftery. MCLUST: Software for Model-Based Cluster Analysis. *Journal of Classification*, 16:297–306, 1999.

[59] C. Fraley and A. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458), 2002.

[60] H.P. Friedman and J. Rubin. On some invariant criteria for grouping data. *Journal of the American Statistical Association*, 62:1159–1178, 1967.

[61] J.H. Friedman. Regularized discriminant analysis. *The Journal of the American Statistical Association*, 84:165–175, 1989.

[62] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic. Press, San Diego, 1990.

[63] G. Galimberti, A. Montanari, and C. Viroli. Penalized factor mixture analysis for variable selection in clustered data. *Computational Statistics & Data Analysis*, 53(12):4301–4310, October 2009.

[64] Z. Ghahramani and G.E. Hinton. The EM algorithm for factor analyzers. Technical report, University of Toronto, 1997.

[65] S. Girard. A non-linear PCA based on manifold approximation. *Computational Statistics*, 15(2):145–167, 2000.

[66] G. Golub and C. Van Loan. *Matrix Computations. Second ed.* The Johns Hopkins University Press, Baltimore, 1991.

[67] G. Govaert and M. Nadif. Clustering with block mixture models. *Pattern Recognition*, 36(2):463–473, 2003.

[68] G. Govaert and M. Nadif. Fuzzy clustering to estimate the parameters of block mixture models. *Soft Computing*, 10(5):415–422, 2006.

[69] G. Govaert and N. Nadif. Algorithms for model-based block Gaussian clustering. *The 4th international Conference on datamining*, pages 536–272, 2008.

[70] G. Govaert and N. Nadif. Latent block model for contingency table. *Communication in Statistics: Theory and Methods*, 39:416–425, 2010.

[71] K. Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighborhood. *Pattern Recognition*, 10:105–112, 1978.

[72] K. Gowda and G. Krishna. The condensed nearest neighbor rule using the concept of mutual nearest neighborhood. *IEEE Transactions on Information Theory*, 25:488–490, 1979.

[73] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. *In Proceedings of ACM SIGMOD*, 98:73–84, 1998.

[74] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. *ICDE*, 15, 1999.

[75] Y-F. Guo, S-J. Li, J-Y. Yang, T-T. Shu, and L-D. Wu. A generalized Foley-Sammon transform based on generalized fisher discriminant criterion and its application to face recognition. *Pattern Recognition letters*, 24:147–158, 2003.

[76] G. Gutin and A.P. Punnen. *The traveling salesman problem and its variations*, volume 12. Springer, 2002.

[77] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[78] W. Haerdle. *Smoothing Techniques with Implementation in S.* Springer, New York, 1991.

[79] M. Hahsler, K. Hornik, and C. Buchta. Getting things in order: an introduction to the r package seriation. Technical Report 58, R, 2009.

[80] Y. Hamamoto, Y. Matsuura, T. Kanaoka, and S. Tomita. A note on the orthonormal discriminant vector method for feature extraction. *Pattern Recognition*, 24(7):681–684, 1991.

[81] J.A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67:123–129, 1972.

[82] J.A. Hartigan. *Clustering Algorithms.* New-York: John Wiley, 1975.

[83] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23:73–102, 1995.

[84] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.

[85] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning.* Springer, New York, second edition, 2009.

[86] R.J. Hathaway. A constrained formulation of maximum-likelihood estimation. *Annals of Statistics*, 13:795–800, 1985.

[87] N.J. Higham. *Matrix nearness problems and its applications*, chapter 1, pages 1–27. Oxford University Press, 1989.

[88] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[89] P. Howland and H. Park. Generalizing discriminant analysis using the generalized singular decomposition. *IEEE transactions on pattern analysis and machine learning*, 2004.

[90] W.L. Hsu. A simple test for the consecutive ones property. *Journal of Algorithms*, 42:1–16, 2002.

[91] L. Hubert. Some applications of graph theory and related nonmetric techniques to problems of approximate seriation: the case of symmetric proximity measures. *Journal Math Stat Psychol*, 27:133–153, 1974.

[92] L. Hubert, P. Arabie, and J. Meulman. Graph-theoretic representations for proximity matrices through strongly-anti-robinsonian or circular strongly-anti-robinsonian matrices. *Psychometrika*, 63:341–358, 1998.

[93] L. Hubert, P. Arabie, and J. Meulman. Combinatorial data analysis: Optimization by dynamic programming. *Society for industrial and Applied Mathematics*, 2001.

[94] J. J. Hull. A database for handwritten text recognition research. *IEEE PAMI*, 16(5):550–554, 1994.

[95] P. Ihm. A contribution to the history of seriation in archaeology. In C. Weihs and W. Gaul, editors, *Studies in classification, data analysis and knowledge organization*, pages 307–316, 2005.

[96] W.S. Jank. The EM algorithm, its stochastic implementation and global optimization: some challenges and opportunities for or. *Perspectives in Operations Research: Papers in honor of Saul Gass' 80th birthday*, pages 367–392, 2006.

[97] R.A. Jarvis and E.A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, 22:1025–1034, 1973.

[98] R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. *Technical report*, arXiv:0909.1440, 2009.

[99] Z. Jin, J.Y. Yang, Z.S. Hu, and Z. Lou. Face recognition based on the uncorrelated optimal discriminant vectors. *Pattern Recognition*, 10(34):2041–2047, 2001.

[100] D. Johnson, S. Krishnan, and J. Chhugani. Compressing large boolean matrices using reordering techniques. *Proceedings of the Thirtieth international conference on Very large data bases*, 30:13–23, 2004.

[101] I. Jolliffe. *Principal Component Analysis*, volume 1737-17 of *2*. Springer-Verlag, New York, 2002.

[102] M. Kachour, J. Fadili, C. Chesneau, C. Dossal, and G. Peyre. The "degrees of freedom" of the lasso for underdetermined systems of linear equations. *Preprint*, 2011.

[103] K. Kailing, H-P. Kriegel, and P. Kroger. Density-connected subspace clustering for high dimensional data. *SIAM int. conf. on data mining*, pages 246–257, 2004.

[104] N. Kambhatla and T. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9:1493–1516, 1997.

[105] D. Kendall. Incidence matrices, interval graphs and seriation in archaeology. *Pacific Journal of Mathematics*, 1969.

[106] D. Kendall. Seriation from abundance matrices. *In: F. Hodson, D. Kendall, P. Tautu (eds) Mathematics in the archaeological and historical sciences. University Press, Edinburgh*, pages 215–252, 1971.

[107] G. Kimeldorf and G Wahba. Some results on Tchebycheffian Spline Functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

[108] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:681–690, 1983.

[109] T. Kohonen. *Self Organizing Maps*. Springer, Berlin, Heidleberg edition, 1995.

[110] F. De la Torre Frade and T. Kanade. Discriminative cluster analysis. *ICML*, pages 241–248, 2006.

[111] M. Law, M. Figueiredo, and A. Jain. Simultaneous Feature Selection and Clustering Using Mixture Models. *IEEE Trans. on PAMI*, 26(9):1154–1166, 2004.

[112] I. Liiv. Seriation and matrix reordering methods: an historical overview. *Wiley interscience*, 3(2):70–91, March 2010.

[113] J. Liu, J.L. Zhang, M.J. Palumbo, and C.E. Lawrence. Bayesian clustering with variable and transformation selection. *Bayesian Statistics*, 7:249–276, 2003.

[114] K. Liu, Y-Q. Cheng, and J-Y. Yang. A generalized optimal set of discriminant vectors. *Pattern Recognition*, 25(7):731–739, 1992.

[115] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L.M. Cam and J. Neyman, editors, *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. University of California Press, 1967.

[116] M. Maier, M. Hein, and U. von Luxburg. *Cluster identification in nearest-neighbor graphs*, volume 4754 of *Lecture Notes in Artificial Intelligence*, chapter In M.Hutter, R. Servedio, and E. Takimoto, editors, Proceedings of the 18th Conference on Algorithmic Learning Theory, pages 196–210. Springer, Berlin, 2007.

[117] M. Maier, M. Hein, and U. von Luxburg. Optimal construction of k-nearest neighbor graphs for identifying noisy clusters. *Theoretical Computer Science*, 410(19):1749–1764, 2009.

[118] F. Marcotorchino. Block seriation problems: a unified approach. *Applied Stochastic Models and Data Analysis*, 3:73–91, 1987.

[119] C. Maugis, G. Celeux, and M. Martin-Magniette. Variable selection in model-based discriminant analysis, *INRIA, RR-7290*. Technical report, INRIA, RR-7290, 2010.

[120] C. Maugis, G. Celeux, and M.-L. Martin-Magniette. Variable selection for Clustering with Gaussian Mixture Models. *Biometrics*, 65(3):701–709, 2009.

[121] C. Maugis, G. Celeux, and M.-L. Martin-Magniette. Variable selection in model-based clustering: A general variable role modeling. *Computational Statistics and Data Analysis*, 53:3872–3882, 2009.

[122] W.T. McCormick, S.B. Deutsch, J.J. Martin, and P.J. Schweitzer. Identification of data structures and relationships by matrix reordering techniques. TR 512, Institute for defense analyses, Arlington, 1969.

[123] W.T. McCormick, P.J. Schweitzer, and T.W. White. Problem decomposition and data reorganization by a clustering technique. *Operations Research*, 20:993 –1009, 1972.

[124] C.E. McCulloch. Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American Statistical Association*, 92:162:170, 1997.

[125] G. McLachlan and K.E. Basford. *Mixture models : inference and applications to clustering*. New York: Marcel Dekker, 1988.

[126] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley Interscience, New York, 1997.

[127] G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Interscience, New York, 2000.

[128] G. McLachlan, D. Peel, and R. Bean. Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics and Data Analysis*, (41):379, 2003.

[129] P. McNicholas and B. Murphy. Parsimonious Gaussian mixture models. *Statistics and Computing*, 18(3):285–296, 2008.

[130] P. McNicholas and B. Murphy. Model-based clustering of microarray expression data via latent gaussian mixture models. *Bioinformatics*, 26(21):2705–2712, 2010.

[131] I. Van Mechelen, H-H. Bock, and P. De Boeck. Two-mode clustering methods: a structured overview. *Statistical Methods in Medical Research*, 13:363–394, 2004.

[132] O.P.J. Meidanis and G. Telles. On the consecutive ones property. *Discrete Applied Mathematics*, 88:325–354, 1998.

[133] X-L. Meng and D. Van Dyk. The EM algorithm - an old folk song sung to a fast new tune. *Journal of the Royal Statistical Society, Series B*, 59(3):511–567, 1997.

[134] B. Mirkin. *Mathematical classification and clustering*. Kluwer, Dordrechti, 1996.

[135] B. Mirkin and S. Rodin. *Graphs genes*, chapter 7. Springer, Berlin, 1984.

[136] A. Montanari and C. Viroli. Heteroscedastic Factor Mixture Analysis. *Statistical Modeling: An International journal (forthcoming)*, (to appear), 2010.

[137] T. B. Murphy, N. Dean, and A. E. Raftery. Variable selection and updating in model-based discriminant analysis for high-dimensional data with food authenticity applications. *Annals of Applied Statistics*, 4(1):396–421, 2010.

[138] F. Murtagh and A.E. Raftery. Fitting straight lines to point patterns. *Pattern Recognition*, 17:479–483, 1984.

[139] N.S. Narayanaswamy and R. Subashini. A new characterization of matrices with the consecutive ones property. *Discrete Applied Mathematics*, 157:3721–3727, 2009.

[140] S. Niermann. Optimizing the ordering of tables with evolutionary computations. *The American Statistician*, 59(1):41–46, 2005.

[141] C.C. Paige and M.A. Saunders. Towards a generalized singular value decomposition. *SIAM J. Numer. Anal.*, 18(3):398–405, 1981.

[142] W. Pan and X. Shen. Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8:1145–1164, 2007.

[143] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high-dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):69–76, 1998.

[144] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572, 1901.

[145] F.W.M. Petrie. Sequences in prehistoric remains. *Journal of the Anthropological Institute of Great Britain and England*, 29(3):295–301, 1899.

[146] W. Polonik. Measuring mass concentration and estimating density contour clusters: an excess mass approach. *Annals of Statistics*, 23(3):855–881, 1995.

[147] Z. Qiao, L. Zhou, and J.Z. Huang. Sparse linear discriminant analysis with applications to high dimensional low sample size data. *International Journal of Applied Mathematics*, 39(1), 2009.

[148] A. Raftery and N. Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006.

[149] B.D. Ripley. SVD, PCA and metric scaling. *Background material of multivariate analysis*, 2004.

[150] W.S. Robinson. A method for chronologically ordering archealogical deposits. *Am Antiq*, 16(4):293–301, 1951.

[151] M.J. Rossman, R.J. Twery, and F.D. Stone. *A Solution to the Traveling Salesman Problem by combinatorial programming.* Peat, Marwick, Mitchell and Co., Chicago, 1958.

[152] S.T. Roweis. EM algorithms for PCA and SPCA. In *NIPS*, 1997.

[153] D. Rubin and D. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, 1982.

[154] H.E. Ruspini. Numerical methods for fuzzy clustering. *Information Sciences*, 2(3):319–350, July 1970.

[155] P.F. Russel and T.R. Rao. On habitat and association of species of anopheline larvae in south-eastern madras. *Journal of Malaria Institute India*, 3:153–178, 1940.

[156] B. Schölkopf, A. Smola, and K. Müller. Non linear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[157] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[158] A.J. Scott and M.J. Symons. Clustering methods based on likelihood ratio criteria. *Biometrics*, 27:387–397, 1971.

[159] D. Scott and J. Thompson. Probability density estimation in higher dimensions. In *Fifteenth Symposium in the Interface*, pages 173–179, 1983.

[160] B. Silverman. *Density estimation for statistics and data analysis.* Chapman and Hall, 1986.

[161] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101, 1904.

[162] W. Stuetzle and R. Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Technical Report, Department of statistics, University of Washington*, 514, 2007.

[163] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):667–288, 1996.

[164] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society, Series B*, 32(2):411–423, 2001.

[165] E. Tipping and C. Bishop. Mixtures of Probabilistic Principal Component Analysers. *Neural Computation*, 11(2):443–482, 1999.

[166] N. Trendafilov and I. T. Jolliffe. DALASS: Variable selection in discriminant analysis via the LASSO. *Computational Statistics and Data Analysis*, 51:3718–3736, 2007.

[167] A. Ultsch and L. Herrmann. The architecture of emergent Self-Organizing Maps to reduce projection errors. *Proceeding ESANN, Brugges*, 2005.

[168] L. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[169] L. van der Maaten, E. Postma, and J. van den Herik. Dimensionality reduction: a comparative review. *Tilburg University Technical Report, TiCC-TR*, 2009.

[170] A. Vathy-Fogarassy, A. Kiss, and J. Abonyi. Hybrid minimal spanning tree and mixture of gaussians based clustering algorithm. *Lecture Notes in Computer Science, Foundations of Information and Knowledge Systems*, pages 313–330, 2007.

[171] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. *IWANN*, 2005.

[172] S. Vines. Simple principal component. *Applied Statistics*, 49:441–451, 2000.

[173] S. Wang and J. Zhou. Variable selection for model-based high dimensional clustering and its application to microarray data. *Biometrics*, 64:440–448, 2008.

[174] J.H. Ward. Hierarchical groupings to optimize an objective function. *Journal of the American Statistical Association*, 58:234–244, 1963.

[175] M.J. Warrens. On Robinsonian dissimilarities, the consecutive ones property and latent variable models. *Advances in Data Analysis and Classification*, 3:169–184, 2009.

[176] M.J. Warrens and W.J. Heiser. Robinson cubes. *In P Brito, P Bertrand, G. Cucumel and F. de Caravalho (eds) Selected Contributions in Data Analysis and Classifcation. Heidelberg: Springer*, pages 515–523, 2007.

[177] G.C.G. Wei and M.A. Tanner. A Monte Carlo implementation fo the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(699704), 1990.

[178] D.M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

[179] D.M. Witten and R. Tibshirani. Penalized classification using Fisher's linear discriminant. *Journal of the Royal Statistical Society, Series B*, (In Press), 2011.

[180] D.M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistic*, 10(3):515–534, 2009.

[181] J.H. Wolfe. Object cluster analysis of social areas. Master's thesis, University of California, Berkeley, 1963.

[182] C. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11:95–103, 1983.

[183] J. Wyse and N. Friel. Block clustering with collapsed latent block models. *Statistics and Computing*, 2010.

[184] B. Xie, W. Pan, and X. Shen. Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electrical Journal of Statistics*, 2:168–212, 2008.

[185] B. Xie, W. Pan, and X. Shen. Penalized mixtures of factor analyzers with application to clustering high-dimensional microarray data. *Bioinformatics*, 26(4):501–508, 2010.

[186] J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, 6:483–502, 2005.

[187] J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. *Advances in Neural Information Processing Systems 20*, pages 1649–1656, 2007.

[188] R. Yoshida, T. Higuchi, and S. Imoto. A mixed factor model for dimension reduction and extraction of a group structure in gene expression data. *IEEE Computational Systems Bioinformatics Conference*, 8:161–172, 2004.

[189] R. Yoshida, T. Higuchi, S. Imoto, and S. Miyano. Array cluster: an analytic tool for clustering, data visualization and model finder on gene expression profiles. *Bioinformatics*, 22:1538–1539, 2006.

[190] J. Zhang. *Visualization for information retrieval*. Springer, 2007.

[191] Y. Zhang and D-Y. Yeung. Worst-case linear discriminant analysis. *NIPS 2011*, 2011.

[192] Z. Zhang, G. Dai, and M.I. Jordan. A flexible and efficient algorithm for regularized fisher discriminant analysis, 2009.

[193] H. Zou and R. Hastie, T.and Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, June 2006.

[194] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. statist. soc.*, 67:301–320, 2005.

[195] H. Zou, T. Hastie, and R. Tibshirani. On the degrees of freedom of the Lasso. *Annals of Statistics*, 35(5):2173–2192, 2007.