Models and estimation algorithms for nonparametric finite mixtures with conditionally independent multivariate component densities

Vy-Thuy-Lynh Hoang

UNIVERSITÉ D'ORLÉANS

ÉCOLE DOCTORALE MATHÉMATIQUES, INFORMATIQUE, PHYSIQUE THÉORIQUE ET INGÉNIERIE DES SYSTÈMES

LABORATOIRE : MAPMO

Thèse présentée par :

Vy-Thuy-Lynh HOANG

soutenue le : 20 Avril 2017

pour obtenir le grade de : Docteur de l'Université d'Orléans

Discipline/ Spécialité : Mathématiques

Models and estimation algorithms for nonparametric finite mixtures with conditionally independent multivariate component densities

Thèse dirigée par :

Didier CHAUVEAU	Professeur, Université d'Orléans, Directeur de thèse

RAPPORTEURS :

Christophe BIERNACKI	Professeur, Université de Lille 1
Laurent BORDES	Professeur, Université de Pau et Pays de l'Adour

---

JURY :

Christophe BIERNACKI	Professeur, Université de Lille 1
Laurent BORDES	Professeur, Université de Pau et Pays de l'Adour
Didier CHAUVEAU	Professeur, Université d'Orléans
Richard EMILION	Professeur Émérite, Université d'Orléans
Catherine MATIAS	Directrice de Recherche CNRS, Université Pierre et Marie Curie
Marguerite ZANI	Professeur, Université d'Orléans

Remerciements

Ce manuscrit conclut trois années de travail, je tiens en ces quelques lignes à exprimer ma reconnaissance envers tous ceux qui, de près ou de loin, y ont contribué.

Je voudrais exprimer ma plus profonde gratitude à mon directeur de thèse, le Professeur Didier CHAUVEAU, pour son encadrement et son soutien. Sa personnalité, ses connaissances, son dévouement, son enthousiasme, sa patience, et ses encouragements durant toutes les années de thèse ont rendu mon expérience à l'Université d'Orléans enrichissante et inoubliable.

Je remercie Professeurs Laurent BORDES et Christophe BIERNACKI d'avoir accepté d'être rapporteurs de cette thèse.

Je voudrais aussi souligner le soutien financier de l'Université d'Orléans et la remercier de m'avoir donné l'occasion de travailler sur le projet de ma thèse de doctorat.

Je ne peux également oublier de remercier mon laboratoire de recherche MAPMO qui ne fait jamais défaut à ses doctorants pour leur procurer un soutien logistique et matériel.

Je souhaite adresser également mes remerciements à Madame Marie-France GRESPIER, Madame Anne LIGER, Madame Marie-Laurence PONCET, Monsieur Romain THERON, Monsieur Christian LAGUERRE, les secrétaires et les techniciens de laboratoire pour leurs aides précieuses et leurs conseils.

Je voudrais remercier mes collègues de m'avoir beaucoup aidée pendant ces 3 années de thèse et tout spécialement: Manon BAUDEL, Mathilde LEGRAND, Remi BUFFE, Julie GAUTHIER, Grégoire VECHAMBRE, Nhat VO, Binh LE, Hieu HO.

Je remercie mes amis Dung NGUYEN, Thao DO, Binh NGUYEN, Tuyen TA, TuaNAnh DOTRAN, pour leurs conseils, leur soutien indéfectible et leurs qualités de cœur au cours de ces trois années éloignée de ma famille.

Un sincère remerciement à Nicole pour m'avoir aidée à apprendre le français.

Je tiens à remercier mes parents, mon petit frère, ma belle-famille pour leur soutien et encouragements tout au long de ces longues années d'études, et pour leur amour sans fin et leur fierté à mon égard. Enfin, je souhaiterais exprimer ma reconnaissance à mon mari, le Dr Huu Quan Do pour avoir consacré son temps avec moi dans la préparation et la rédaction de cette thèse. Merci à ma petite fille d'amour.

Encore une fois, je remercie tous ceux qui ont contribué à cette thèse.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Populations of individuals may often be divided into subgroups. Examining a sample of measurements to discern and describe subgroups of individuals, even when there is no observable variable that readily indexes into which subgroup an individual properly belongs, is sometimes referred to as "unsupervised clustering" in the thesis, and in fact mixture models may be generally thought of as comprising the subset of clustering methods known as model-based clustering.

Most of the density functions that are usually considered in common statistical models, are unimodal, that is, they have at most one peak. However, often we want to be able to represent densities with multiple modes. A common way to do this is to create a mixture model. A finite mixture model is a convex combination of two or more probability density functions. Mixture models have been used in many applications in statistical analysis and machine learning such as modeling, clustering, classification and latent class and survival analysis. Consequently, finite mixture models are a powerful and flexible tool for modeling complex data. We refer to Chapter 2, Section 2.1 for an overview of mixture models.

Finite mixtures give a flexible way to model a wide variety of random observations (see, e.g.,McLachlan and Peel [2000]). Finite mixture models may be used in situations beyond those for which clustering of individuals is of interest. For one thing, finite mixture models give descriptions of entire subgroups (called *components*), rather than assignments of individuals to those subgroups. Indeed, even the subgroups may not necessarily be of interest; sometimes finite mixture models merely provide a means for adequately describing a particular distribution, such as the distribution of residuals in a linear regression model where outliers are present. Much of the theory of these models involves the assumption that the subgroups are distributed according to a particular parametric shape and quite often this parametric family is univariate or multivariate normal.

Mixture models are of parametric or semi/non-parametric form. Parametric approaches impose a structure on the data and limits the capacity in fitting multidimensional data (see Section 2.1.2), whereas non-parametric methods infer the underlying structure from the data itself. There is a growing literature on nonparametric identification of finite mixtures. Univariate mixtures are generally not identified nonparametrically unless adding some restrictions (Bordes et al. [2006d], Hunter et al. [2007]). Models and algorithms for nonparametric estimation of finite multivariate mixtures have been proposed,

where it is usually assumed that coordinates are independent conditional on the subpopulation from which each observation is drawn. These approaches has appeared in a growing body of literature on non- and semiparametric multivariate mixture models with the earlier proposals of Hettmansperger and Thomas [2000], Hall and Zhou [2003], Elmore et al. [2004], Hall et al. [2005], Allman et al. [2009]. The nonparametric multivariate mixtures which have computational procedures akin to the Expectation-Maximization (EM) algorithm (Dempster et al. [1977]) are applicable more generally (Benaglia et al. [2009a], Levine et al. [2011]). Several authors have addressed this conditionally i.i.d. (independent and identically distributed) finite mixture model and proposed extensions to semi- and non-parametric mixtures preserving the identifiability property (Section 2.1.3).

As well as providing a framework for building more complex probability distributions, mixture models can also be used to cluster data. In many applications, the parameters of mixture models are determined by maximum likelihood (Section 2.2.1). A general technique for finding maximum likelihood estimators in latent variable models is the EM algorithm. The association of EM algorithms with mixture models has a long history. We therefore reserve Section 2.2 to introduce the EM algorithm (Dempster et al. [1977]). The EM algorithm can be treated as a special case of the MM algorithm whose general principle behind was first enunciated by Ortega and Rheinboldt [1970]. However, in the EM algorithm conditional expectations are usually involved, while in the MM algorithm convexity and inequalities are the main focus, and it is easier to understand and apply in most cases (Section 2.2.3). There are many EM algorithms that have been proposed to estimate the parameters of parametric and semi-/non-parametric mixture models (Bordes et al. [2007], Benaglia et al. [2009a], Levine et al. [2011], Chauveau et al. [2015], Shen et al. [2016]).

The conditional independence assumption for nonparametric multivariate finite mixture models is the subject of an increasing number of theoretical and algorithmic developments in the statistical literature. In these models the dependence comes only from the mixture (Section 2.2.7). However, there are more and more real data which have dependence within the structure; for instance an example of breast cancer dataset introduced in Section 3.1. In such data the existing nonparametric multivariate mixture models do not work and then they need extensions. For instance, Zhu and Hunter [2015] propose an extension of nonparametric multivariate finite mixture models using ideas based on independent components analysis (ICA). Our first contribution in this work consists, in Chapter 3, to relax this assumption and allowing for independent multivariate *blocks* of coordinates, conditional on the subpopulation from which each observation is drawn. Otherwise the density functions of these blocks are completely multivariate and nonparametric. Hence this nonparametric finite mixture model with multivariate blocks is more flexible and useful (Section 3.1). We then propose an EM-like algorithm, called `mvnpEM`, extended from the `npEM` algorithm proposed by Benaglia et al. [2009a] for this model, and derive some strategies for selecting the bandwidth matrix involved in the nonparametric estimation step of the algorithm. We evaluate also in Chapter 3 the performance of this algorithm through several numerical simulations and experiment a real dataset of reasonably large dimension on this new model and algorithm to illustrate its potential from the model based, unsupervised clustering perspective.

The question about the convergence of the sequence of parameter estimates generated

by an EM algorithm have been studied (Wu [1983]). Redner and Walker [1984] show that EM has linear rate of convergence. Jordan and Xu [1995] mentioned convergence properties of the EM Algorithm for Gaussian Mixtures. Each EM iteration can only improve the likelihood, guaranteeing convergence to a local maximum. Our `mvnpEM` as EM-like `npEM` algorithm, successes in practice but has not any definite theoretical proofs of consistency. In Chapter 4, we extend the idea of establishing an algorithm satisfying a descent property with respect to a log-likelihood objective function and proving that the algorithm converges to a minimizer of such an objective function suggested in Levine et al. [2011] and Shen et al. [2016] to smooth our multivariate model and define an alternative algorithm, namely `mvnpMSL`. The detailed introduction of the smoothed model and `mvnpMSL` algorithm is addressed together with the performance in implementation on the same simulated examples and real data of Chapter 3. This smoothed model and its algorithm own the monotony property and display the similar results in empirical experiments.

In the framework of multiple testing, the p-values under H_0 are uniformly distributed on $[0, 1]$ while the distribution of the p-values associated to H_1 is unknown. The idea to mix parametric and nonparametric estimates is not new (Olkin and Spiegelman [1987], Efron et al. [1996], Priebe and Marchette [2000], Di Marzio and Taylor [2004]). Parametric models have been used with Beta distribution for the p-values (Allison et al. [2002], Liao et al. [2004]) or Gaussian distribution of the probit transformation of the p-values (McLachlan et al. [2006]). Hoti and Holmström [2004] has a new idea in using nonparametric estimate for the unknown distribution in the mixture model of p-values. Robin et al. [2005] proposed a procedure where the unknown part is estimated with a weighted kernel function. Bordes et al. [2006b] consider a two-component mixture model where one component distribution is known while the mixing proportion and the other component distribution are unknown in the environment of relaxing the assumption that the unknown distribution belongs to a parametric family. Following the line of Levine et al. [2011], Nguyen and Matias [2014] constructed an iterative estimator sequence of the unknown density that relies on the maximization of a smoothed likelihood. In the context of having more and more data from multiple testing (genomics, microarrays analysis, neuro-imaging,...) we found a very few references (Chi et al. [2008]-the first one) associated to the FDR control with multivariate p-values. In Chapter 5, our motivation is to design multivariate mixture models adapted to the distribution of multivariate $\boldsymbol{p}$-values from hypothesis tests. A constrained version of the `npEM` algorithm from Benaglia et al. [2009a] succeed in the very simple examples with 2 components in term of comparing with univariate FDR control but does not for $m \geq 3$-component mixtures. We thus propose an alternative constrained version of our algorithm from Chapter 3 and study its potential performance in Chapter 5. Once again, our mixture models and algorithms can be evaluated effectively through a high dimensional real dataset.

Finally, we present the discussions together with potential perspectives of our models and algorithms in the last Chapter.

Chapter 2

Mixture models and EM algorithms

In this chapter, we present an overview of the finite mixture models and the EM algorithms to estimate the parameters of these models.

2.1 Mixture models

Finite mixture models have gained a popularity in many fields of sciences and are being increasingly exploited as a convenient tool because of their flexibility. In statistics, mixture models have a long history which goes back to over a century ago. Beginning with the idea about the possibility of resolving the normal distribution into several other normal distributions in Quetelet [1846] and with the classic paper of Pearson [1894] on his moments based fitting of a mixture of two univariate normal components. The moment of a mixture is convex combination of the moments of the component densities. Pearson's approach is generally thought of as the starting point of the analysis of mixtures. The contribution of Charlier [1906] was made in the early part of the 20th century to improve this method of moments. After 30 years, Charlier and Wicksell [1923] continued to extend this method to the mixture of bivariate normal distributions. Cohen [1967] approached to the case of equal variances where the estimates depend uniquely on the negative root of a cubic equation. Bhattacharya [1967] and Roeder [1994] proposed methods based on graphical procedures to determine the number of components which is also a very significant issue in mixture model. The most popular mixture model is the one consisting of Gaussian components (see McLachlan et al. [1999], Pearson [1894]). A heavy-tailed alternative to Gaussian mixtures is to use mixtures of t-distributions in McLachlan and Peel [2000]. Since the appearance of the monograph of McLachlan and Basford [1988] on finite mixtures, the literature has expanded enormously. We refer to McLachlan and Peel [2000] for a comprehensive survey on the history and applications of finite mixture models. Other helpful resources on the theory, applications and developments in the field are Lindsay et al. [1983], Lindsay [1995], Krishnan and McLachlan [1997], Böhning et al. [1998], Frühwirth-Schnatter [2006], Schlattmann [2009]

The most general model for mixtures is as follows: suppose the vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are a simple random sample from a finite mixture of $m > 1$ arbitrary distributions. The

density of each $\boldsymbol{X}_i$ may be written

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j f_j(\boldsymbol{x}_i), \tag{2.1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^r$, and λ_j denotes the proportion (weight) of component j in the population; the λ_j's are thus positive and $\sum_{j=1}^{m} \lambda_j = 1$. The f_j's are the component densities, drawn from some family of density functions $\mathcal{F}$ absolutely continuous with respect to Lebesgue measure. We write $\boldsymbol{\theta} = (\boldsymbol{\lambda}, \boldsymbol{f})$ for the parameter vector.

2.1.1 The label switching problem

For any permutation ν of $1, \cdots, m$ define the corresponding permutation of the parameter vector $\boldsymbol{\theta}$ by

$$\nu(\boldsymbol{\theta}) = \nu(\boldsymbol{\lambda}, \boldsymbol{f}) = ((\lambda_{\nu(1)}, \cdots, \lambda_{\nu(m)}), (f_{\nu(1)}, \cdots, f_{\nu(m)})).$$

Given a mixture model with m components, there are $m!$ symmetric modes of the distribution with respect to the permutation of the component labels. If we have no information that distinguishes between the components of the mixture, the distribution $g_{\boldsymbol{\theta}}$ is the same for all permutation of $\boldsymbol{\theta}$. This symmetric property can cause problems when we try to estimate quantities which relate to individual components of the mixture. For example, assume a $m = 2-$components mixture has the distribution of the population is

$$g_{\boldsymbol{\theta}}(x) = \lambda_1 \mathcal{N}(\mu_1, 1)(x) + \lambda_2 \mathcal{N}(\mu_2, 1)(x),$$

where $\boldsymbol{\theta} = (\lambda_1, \lambda_2, \mu_1, \mu_2)$ is the parameter vector. Based on a sample from this population, suppose that the estimate of $\boldsymbol{\theta}$ be $\widehat{\boldsymbol{\theta}} = (\frac{1}{2}, \frac{1}{2}, 2, 3)$ then the distribution of population can be

$$g_{\boldsymbol{\theta}}(x) = \frac{1}{2}\mathcal{N}(2, 1)(x) + \frac{1}{2}\mathcal{N}(3, 1)(x)$$

or

$$g_{\boldsymbol{\theta}}(x) = \frac{1}{2}\mathcal{N}(3, 1)(x) + \frac{1}{2}\mathcal{N}(2, 1)(x).$$

Then we can not identify the components or we can not specify which estimate value corresponds to each sub-population. It means that (λ_1, μ_1) and (λ_2, μ_2) are exchangeable. In mixture model this is called label switching. Label switching problem is crucial in some computational issues.

2.1.2 Parametric mixture models

Model (2.1) is not identifiable if no restrictions are placed on $\mathcal{F}$, where "identifiable" means that $g_{\boldsymbol{\theta}}$ has a *unique* representation of the form (2.1) and also that we do not consider that "label-switching". The most common restriction in the mixture literature is to assume that the family $\mathcal{F}$ is *parametric*, i.e., that any $f \in \mathcal{F}$ is completely specified by a finite-dimensional parameter or the component densities $f_j(\boldsymbol{x}_i)$ are specified as $f_j(\boldsymbol{x}_i; \xi_j)$, where

ξ_j is the vector of unknown parameters in the postulated form for the jth component density in the mixture. The mixture density $g_{\boldsymbol{\theta}}(x_i)$ can then be written as

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j f_j(\boldsymbol{x}_i; \xi_j). \tag{2.2}$$

The vector

$$\boldsymbol{\theta} = (\lambda_1, ..., \lambda_m, \boldsymbol{\xi}). \tag{2.3}$$

contains all the unknown parameters in the mixture model and $\boldsymbol{\xi}$ is the vector containing all the parameters in $\xi_1, ..., \xi_m$ of the component densities $f_1, ..., f_m$, respectively.

To demonstrate the notation above for defining a parametric mixture, we consider a mixture of univariate normal and Gaussian components with the means μ_j and the variances σ_j^2. For this model, the mixture density of the measurement X_i can be represented as

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j f_j(\boldsymbol{x}_i; \mu_j, \sigma_j), \tag{2.4}$$

where

$$f_j(\boldsymbol{x}_i; \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2}.$$

In this case, the vector $\boldsymbol{\theta}$ of unknown parameters is given by

$$\boldsymbol{\theta} = (\lambda_1, ..., \lambda_m, (\mu_1, \sigma_1), ..., (\mu_m, \sigma_m)).$$

where $\mu_1, ..., \mu_m$ are the distinct elements of the component means, and $\sigma_1^2, ..., \sigma_m^2$ are the distinct elements of the component variances.

The most used and studied parametric mixture model is the Gaussian mixture, where f_j is the density of a (univariate or multidimensional) Gaussian distribution with mean (vector) μ_j and variance (matrix) Σ_j. Such models are called the Gaussian mixtures and are the most used and studied parametric mixture models, such as in Lee and McLachlan [2013], Dempster et al. [1977], etc.

The Old Faithful is a simple example of a dataset to which mixture models may be applied. In this dataset, measurements give time in minutes between eruptions of the Old Faithful geyser in Yellowstone National Park, USA. This sample was depicted in Figure 2.1 as a mixture of two univariate Gaussian distributions. These data are available in the datasets package in R (R Core Team [2016]);

Section 2.1.3 presents various ways of relaxing this parametric assumption while preserving an identifiability property. In the recent literature, finite mixtures of non-normal distributions have been considered as alternatives to the traditional Gaussian mixture, see, e.g., Lee and McLachlan [2013] which provides a comprehensive overview. These non-normal mixtures are mostly proposed to model heavy-tailed or skewed normal distributions, but are not appropriate for, e.g., non-elliptical clusters.

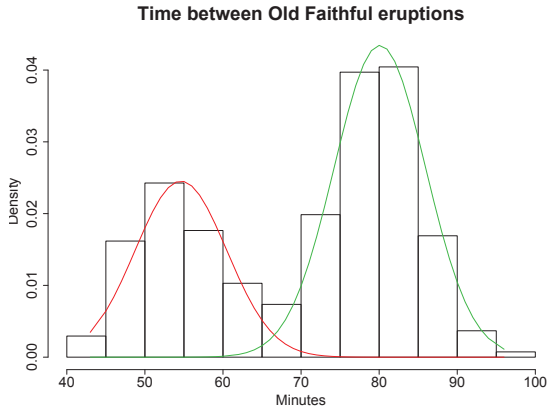


Figure 2.1 – The Old Faithful dataset is suggestive of a two-component mixture of Gaussian functions.

2.1.3 Recent extensions to Semi- and non-parametric mixtures

In this work, the term "nonparametric" will always mean that no assumptions are made about the form of the f_j's, even though the weights $\boldsymbol{\lambda}$ are scalar parameters. Note that other authors as, e.g., Lindsay [1995], speak of "nonparametric mixture modeling" in a different sense: The family $\mathcal{F}$ is fully specified up to a finite-dimensional parameter, but the mixing distribution, rather than having a finite support of known cardinality m like here, is assumed to be completely unspecified.

As said above, nonparametric mixture models are not identifiable if no restrictions are placed on the family $\mathcal{F}$ to which the f_j's belong. The classical definition of identifiability requires that any two different values $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$ correspond to two different distributions $g_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\theta}'}$. Weaker notions of identifiability can be considered, and in the particular case of mixtures, the fact that there always exists $m!$ permutations of the labels in $\boldsymbol{\theta} = (\lambda_1, \ldots, \lambda_m, f_1, \ldots, f_m)$ that result in the same distribution $g_{\boldsymbol{\theta}}$ is one of those. Sometimes, the essentially nonparametric density functions in $\mathcal{F}$ may be partially specified by scalar parameters, a case often called semi-parametric. For instance, in the univariate ($r = 1$) case, Bordes et al. [2006d] and Hunter et al. [2007] proved that when $f_j(x) = f(x - \mu_j)$ for some density $f(\cdot)$ that is symmetric about zero, the mixture (2.1) admits a unique representation whenever $m \leq 3$, except in very special cases. In the multivariate situation, Benaglia et al. [2009a] and recently Chauveau et al. [2015] propose some semiparametric mixture models as well.

In the multivariate situation, the common restriction placed on $\mathcal{F}$ in a number of recent theoretical and algorithmic developments in the statistical literature is that each joint density $f_j(\cdot)$ is equal to the product of its marginal densities. In other words, the coordinates of the $\boldsymbol{X}_i$ vector are independent, conditional on the subpopulation or component (f_1 through f_m) from which $\boldsymbol{X}_i$ is drawn. Therefore, model (2.1) becomes

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jk}(x_{ik}). \tag{2.5}$$

This conditional independence assumption has been introduced by Hall and Zhou [2003], who established that when $m = 2$, identifiability of parameters generally follows in $r \geq 3$ dimensions. This result has been extended by Hall et al. [2005] and finally, Allman et al. [2009] established the identifiability for model (2.5) if $r \geq 3$, regardless of m.

Several authors addressed the problem of estimating the parameters of these semi- or non-parametric mixture models. In the univariate case, Bordes et al. [2006d] and Hunter et al. [2007] both proposed estimators based on a minimum contrast approach, a method very difficult to extend beyond $m = 2$ components, since the key idea is based on the possibility for $m = 2$ to invert the mixture representation, expressing the cumulative density function (c.d.f.) F of the unknown f in terms of λ and the c.d.f. $G_{\boldsymbol{\theta}}$ of $g_{\boldsymbol{\theta}}$. For the multivariate model (2.5), Hall et al. [2005] gave estimators based on an inversion of the mixture, that applies only in the case when $m = 2$ and $r = 3$, due to analytical difficulties appearing beyond this case.

Recently, Hall and Zhou [2003] looked at r-variate data drawn from a mixture of two distributions, each having independent nonparametric components, and proved that under mild regularity assumptions their model is identifiable for $r \geq 3$. The non-identifiability for $r \leq 2$ requires to restrain the class of pdf $\mathcal{F}$. For example, for $r = 1$, restraining $\mathcal{F}$ to the location-shifted symmetric pdf, we obtain the following semiparametric mixture model:

$$g_{\boldsymbol{\theta}}(x) = \sum_{j=1}^{m} \lambda_j f(x - \mu_j), \quad x \in \mathbb{R}, \quad \boldsymbol{\theta} = (\boldsymbol{\lambda}, \boldsymbol{\mu}, f), \tag{2.6}$$

where the $\lambda_j's$, the $\mu_j's$ and $f \in \mathcal{G} = \{\text{even pdf on } \mathbb{R}\}$ are unknown. Hence the model parameter is

$$\boldsymbol{\theta} = (\boldsymbol{\phi}, f) = ((\lambda_j, \mu_j)_{j=1,\cdots,m}, f) \in \boldsymbol{\Theta} = \boldsymbol{\Phi} \times \mathcal{F},$$

where

$$\boldsymbol{\Phi} = \left\{ (\lambda_j, \mu_j)_{j=1,\cdots,m} \in \{(0,1) \times \mathbb{R}\}^m; \sum_{j=1}^{m} \lambda_j = 1 \text{ and } \mu_i \neq \mu_j \text{ for } 1 \leq i < j \leq m \right\}.$$

2.2 The EM algorithm

Mixture models are deeply connected to the EM (Expectation – Maximization) algorithm. In particular, a very detailed treatment of the EM method for exponential families was published by Rolf Sundberg in his thesis and several papers Sundberg [1972], Sundberg [1974], Sundberg [1976]. Dempster et al. [1977] generalized the method and sketched a convergence analysis for a wider class of problems and established the EM method as an important tool of statistical analysis. It is often efficient approach for locating the posterior mode of a distribution (Tanner and Wong [1987], Wei and Tanner [1990]). This algorithm, as defined in the seminal article Dempster et al. [1977], is more properly understood to be a class of algorithms, a number of which predate even Dempster et al. [1977] in the literature. These algorithms are designed for maximum likelihood estimation (MLE) in missing data problems, of which finite mixtures are canonical examples because the unobserved labels of the individuals (as in unsupervised clustering) give an easy interpretation of missing data. A recent account of the EM algorithm principle, properties

and generalizations can be found in McLachlan and Krishnan [2008], and mixture models are deeply detailed in McLachlan and Peel [2000].

The EM algorithm formalizes an intuitive idea for obtaining parameter estimates when some of the data are missing:

1. replace missing values by estimated values,
2. estimate parameters.
3. Repeat
   - step (1) using estimated parameter values as true values, and
   - step (2) using estimated values as "observed" values, iterating until convergence.

In a missing data setup, the n-fold product of the probability density function (pdf) $g_{\boldsymbol{\theta}}$ of the observations corresponds to the *incomplete* data pdf, associated with the log-likelihood $\mathcal{L}_{\boldsymbol{x}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log g_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$. In mixture models and many other missing data situations, maximizing $\mathcal{L}_{\boldsymbol{x}}(\boldsymbol{\theta})$ leads to a difficult problem. Intuitively, EM algorithms replace this unfeasible maximization by the maximization of a pseudo-likelihood that resembles the likelihood for some complete data $\boldsymbol{y}$ that is defined from the model, so that this pseudo-likelihood is easy to maximize. Assuming $\boldsymbol{y}$ comes from a complete data pdf $\boldsymbol{g}^c_{\boldsymbol{\theta}}$, the EM algorithm iteratively maximizes the operator

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) := \mathbb{E}[\log \boldsymbol{g}^c_{\boldsymbol{\theta}}(\boldsymbol{y})|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}],$$

the expectation being taken relatively to the conditional distribution of $(\boldsymbol{y}|\boldsymbol{x})$, for the value $\boldsymbol{\theta}^{(t)}$ of the parameter at iteration t.

Thus the EM algorithm consists of an **E-step** (Estimation step) followed by an **M-step** (Maximization step) defined as follows:

1. **E-step:** `compute` $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$
2. **M-step:** `set` $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$.

Classification procedures that use labeled samples to train the classifier are said to be supervised. Sometimes we do not have the training data. Classification procedures which use only unlabeled samples are said to be unsupervised. One of the statistical approaches for unsupervised learning is the method of moments. In the method of moments, the unknown parameters in the model are related to the moments of one or more random variables, and thus, these unknown parameters can be estimated given the moments. Unsupervised learning is the learning task of inferring a function to describe hidden structure from unlabeled data. The EM algorithm is one of the most practical methods for learning latent variable models and thus, is the primary tool in finite mixture models and model-based clustering. Vandewalle [2009] estimated the models for which the semi-supervised classification is considered using both labeled data and many unlabeled data.

The DLR paper of Dempster et al. [1977] has made many significant contributions and EM algorithm has become a very popular computational method in Statistics. However, Wu [1983] showed that the proof convergence of EM sequences in DLR is incorrect. Wu [1983] studied more broadly two convergence aspects of the EM algorithm that have been considered and obtained several results in the literature:

- does the EM algorithm find a local maximum or a stationary value of the incomplete data likelihood function?

- does the sequence of parameter estimates generated by EM converge?

He summarized some convergence properties of EM algorithm (see more detail in Wu [1983]). Jordan and Xu [1995] have forged a link between EM and gradient methods via the projection matrix and analyzed the convergence properties of EM algorithm for Gaussian Mixtures in terms of special properties of this matrix.

Many important inference problems in Statistics such as latent variable models and random parameter models, turn out to be solvable by EM when they are formulated as missing value problems. However, there are also well documented limitations of EM algorithm: it could converge to local maximum or saddle points of the loglikelihood function and its rate of convergence can be slow since it depends on the starting values. Many non-stochastic improvements on the EM algorithm have been proposed (Louis [1982], Meilijson [1989], Silverman et al. [1990], Green [1990]) but did not completely result. Stochastic EM comes as an attractive alternative to EM algorithm. The main idea of Stochastic EM is to impute a sample value drawn from the conditional distribution of the missing data given the parameter. This is called the S-step. Stochastic EM is particularly useful in problems where EM is intractable. For example, in the problems where the computation of E-step of EM involves high dimensional integrations. Stochastic EM generally converges reasonably quickly to its stationary regime (Diebolt and Robert [1994]). Celeux et al. [1996] compared the characteristics of three stochastic versions of EM: the SEM algorithm (Broniatowski et al. [1983], Celeux and Diebolt [1985]), the SAEM algorithm (Celeux and Diebolt [1989]) and the MCEM algorithm (Wei and Tanner [1990], Tanner [1991]). They show that, for some particular mixture situations, the SEM algorithm is almost always preferable to the EM. Chauveau [1995] proposed an extension of the SEM algorithm in a particular case of incomplete data, where the loss of information is due both to mixture models and censored observations. Recently, Bordes and Chauveau [2017] proposed Stochastic EM algorithms for parametric and semiparametric mixture models for randomly right censored lifetime data, provided they are identifiable.

On spirit of a self-contained literature, we recall the maximum likelihood estimation in Subsection 2.2.1 and present an example to introduce the EM algorithm in Subsection 2.2.2. An overview of the general EM algorithm for mixture models is described in Subsections 2.2.4, and EM algorithms for parametric/non-parametric/semi-parametric mixture are described in Subsections 2.2.5/2.2.7 and 2.2.6.

2.2.1 Maximum Likelihood Estimation (MLE)

We have a density function $g(x|\boldsymbol{\theta})$ that is indexed by the set of parameters $\boldsymbol{\theta}$ (e.g., g might be a set of Gaussian and $\boldsymbol{\theta}$ could be the mean and variance). We also have a data set of size n, supposedly drawn from this distribution, i.e., $\boldsymbol{x} = \{x_1, ..., x_n\}$. That is, we assume that these data vectors are independent and identically distributed (i.i.d.) with distribution g. Therefore, the resulting density for the samples is

$$g(\boldsymbol{x}|\boldsymbol{\theta}) = \prod_{i=1}^{n} g(x_i|\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}). \tag{2.7}$$

This function $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x})$ is called the likelihood of the parameters given the data, or just the likelihood function. The likelihood is thought of as a function of the parameters $\boldsymbol{\theta}$ where the data $\boldsymbol{x}$ is fixed. In the maximum likelihood problem, our goal is to find the $\boldsymbol{\theta}$ that maximizes $\mathcal{L}$. That is, we wish to find $\widehat{\boldsymbol{\theta}}$ where

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}). \tag{2.8}$$

Often we maximize $\log(\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}))$ instead because it is numerically easier.

Depending on the form of $f(\boldsymbol{x}|\boldsymbol{\theta})$ this problem can be easy or hard. For example, if $f(\boldsymbol{x}|\boldsymbol{\theta})$ is simply a single Gaussian distribution where $\boldsymbol{\theta} = (\mu, \sigma^2)$, then we can set the derivative of $\log(\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{x}))$ to zero, and solve directly for μ and σ^2. This example is presented in Example 1. We also present a comprehensive example for the multivariate normal case in Example 2. Moreover, Example 3 shows that the MLE is not easy for mixtures.

Exemple 1 (Univariate normal model)**.** *Let $X_1, X_2, ..., X_n$ be a univariate random sample from a normal distribution with unknown mean μ and variance σ^2. To find maximum likelihood estimators of mean μ and variance σ^2, the probability density function can be written as a function of $\theta_1 = \mu$ and $\theta_2 = \sigma^2$:*

$$g(x_i; \theta_1, \theta_2) = \frac{1}{\sqrt{\theta_2}\sqrt{2\pi}} exp\left[-\frac{(x_i - \theta_1)^2}{2\theta_2}\right]$$

for $-\infty < \theta_1 < \infty$ and $0 < \theta_2 < \infty$.

Now, it makes the likelihood function

$$\mathcal{L}(\theta_1, \theta_2) = \prod_{i=1}^{n} f(x_i; \theta_1, \theta_2) = \theta_2^{-n/2}(2\pi)^{-n/2}\exp\left[-\frac{1}{2\theta_2}\sum_{i=1}^{n}(x_i - \theta_1)^2\right],$$

and therefore the log of the likelihood function

$$\log \mathcal{L}(\theta_1, \theta_2) = -\frac{n}{2}\log\theta_2 - \frac{n}{2}\log(2\pi) - \frac{1}{2\theta_2}\sum_{i=1}^{n}(x_i - \theta_1)^2.$$

Upon taking the partial derivative of the log likelihood with respect to θ_1, and setting to 0, we see that a few things cancel each other out, leaving us with:

$$\frac{\partial \log \mathcal{L}(\theta_1, \theta_2)}{\partial \theta_1} = \frac{1}{\theta_2} \sum_{i=1}^{n} (x_i - \theta_1) := 0.$$

Multiplying through by θ_2, and distributing the summation, we get:

$$\sum_{i=1}^{n} x_i - n\theta_1 = 0.$$

Solving for θ_1, and putting on its hat, we have shown that the maximum likelihood estimate of θ_1 is:

$$\widehat{\theta}_1 = \widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i := \overline{x}.$$

Now for θ_2, taking the partial derivative of the log likelihood with respect to θ_2, and setting to 0, we get:

$$\frac{\partial \log \mathcal{L}(\theta_1, \theta_2)}{\partial \theta_2} = \frac{-n}{2\theta_2} + \frac{\sum_{i=1}^{n} (x_i - \theta_1)^2}{2\theta_2^2} := 0.$$

Multiplying through by $2\theta_2^2$, we get:

$$-n\theta_2 + \sum_{i=1}^{n} (x_i - \theta_1)^2 = 0.$$

Then, solving for θ_2, and putting on its hat, it is shown that the maximum likelihood estimate of θ_2 is:

$$\widehat{\theta}_2 = \widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2.$$

In summary, the maximum likelihood estimators of μ and variance σ_2 for the normal model are:

$$\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i = \overline{X} \text{ and } \widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (X_i - \overline{X})^2.$$

respectively.

Exemple 2 (Multivariate normal model)**.** *Let $X_1, X_2, ..., X_n$ be a multivariate random sample from a normal distribution with unknown mean vector $\boldsymbol{\mu}$ and variance matrix $\boldsymbol{\Sigma}$. The probability density function can be written as*

$$g(\boldsymbol{x}_i|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{r/2}\sqrt{\det \boldsymbol{\Sigma}}} e^{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})}, \ \boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.9}$$

where as stated in the introduction $\boldsymbol{x}_i \in \mathbb{R}^r$, $\boldsymbol{\mu} \in \mathbb{R}^r$ and $\boldsymbol{\Sigma}$ is a $r \times r$ symmetric positive definite matrix.

To find maximum likelihood estimators of mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$, we need to recall some results from matrix algebra.

The trace of a square matrix $tr(A)$ is equal to the sum of A's diagonal elements. The trace of a scalar equals that scalar. Also, $tr(A+B) = tr(A)+tr(B)$, and $tr(AB) = tr(BA)$ which implies that $\Sigma_i x_i^T A x_i = tr(AB)$ where $B = \Sigma_i x_i x_i^T$. Also note that $\det A$ indicates the determinant of a matrix and that $\det A^{-1} = 1/\det A$.

We need to take derivatives of a function of a matrix $f(A)$ with respect to elements of that matrix. Therefore, we define $\frac{\partial f(A)}{\partial A}$ to be the matrix with i,j^{th} entry $\left[\frac{\partial f(A)}{\partial \alpha_{i,j}}\right]$ where $\alpha_{i,j}$ is the i,j^{th} entry of A. The definition also applies taking derivatives with respect to a vector. First, $\frac{\partial x^T A x}{\partial x} = (A + A^T)x$. Second, it can be shown that when A is a symmetric matrix:

$$\frac{\partial \det A}{\partial \alpha_{i,j}} = \begin{cases} \mathcal{A}_{i,j} & \text{if} \quad i = j \\ 2\mathcal{A}_{i,j} & \text{if} \quad i \neq j \end{cases}$$

where $\mathcal{A}_{i,j}$ is the i,j^{th} cofactor of A. Given the above, we see that:

$$\frac{\partial \log\det A}{\partial A} = \left\{ \begin{array}{ll} \mathcal{A}_{i,j}/\det A & \text{if} \quad i = j \\ 2\mathcal{A}_{i,j}/\det A & \text{if} \quad i \neq j \end{array} \right\} = 2A^{-1} - \text{diag}\ (A^{-1})$$

by the definition of the inverse of a matrix. Finally, it can be shown that:

$$\frac{\partial tr(AB)}{\partial A} = B + B^T - \text{diag}(B).$$

Returning the example, it can be shown that the log of the likelihood function is

$$\log \mathcal{L}(\boldsymbol{\theta}) = -\frac{nr}{2}\log(2\pi) - \frac{n}{2}\log(\det \boldsymbol{\Sigma}) - \frac{1}{2}\sum_{i=1}^{n}((\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})). \tag{2.10}$$

Therefore, to find $\boldsymbol{\theta}$, we solve the system

$$\frac{\partial \log \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\frac{n}{2}\frac{\partial \log(\det \boldsymbol{\Sigma})}{\partial \boldsymbol{\theta}} - \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\theta}}\sum_{i=1}^{n}((\boldsymbol{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})) = 0. \tag{2.11}$$

Taking the derivative with respect to $\boldsymbol{\mu}$ by replacing $\boldsymbol{\theta}$ by $\boldsymbol{\mu}$ and using the symmetric feature of matrix $\boldsymbol{\Sigma}^{-1}$, we get

$$\sum_{i=1}^{n} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) = 0,$$

with which we can easily solve for $\boldsymbol{\mu}$ to obtain:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i.$$

To find $\boldsymbol{\Sigma}$, replacing $\boldsymbol{\theta}$ by $\boldsymbol{\Sigma}^{-1}$ and rewriting Equation (2.11) as

$$\sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\Sigma}^{-1}}\left(-\log(\det \boldsymbol{\Sigma}^{-1}) + tr(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T)\right) = 0. \tag{2.12}$$

It is equivalent to

$$\begin{aligned}\sum_{i=1}^{n}(-2\boldsymbol{\Sigma}+\text{diag}(\boldsymbol{\Sigma})+2N_i-\text{diag}(N_i)) &= -2\sum_{i=1}^{n}(\boldsymbol{\Sigma}-N_i)+\text{diag}\left(\sum_{i=1}^{n}(\boldsymbol{\Sigma}-N_i)\right)\\ &= -2\boldsymbol{S}+\text{diag}\ \boldsymbol{S}=0\end{aligned}$$

where $N_i=(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})^T$ and $\boldsymbol{S}=\sum_{i=1}^{n}(\boldsymbol{\Sigma}-N_i)$. This implies that $\boldsymbol{S}=0$. This gives

$$\sum_{i=1}^{n}(\boldsymbol{\Sigma}-N_i)=0 \quad \text{or} \quad \boldsymbol{\Sigma}=\frac{1}{n}\sum_{i=1}^{n}N_i=\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})^T. \tag{2.13}$$

Exemple 3. *Suppose that the p.d.f. of a random vector $\boldsymbol{X}$ has a 2-component mixture form*

$$g(\boldsymbol{x};\lambda)=\lambda_1 f_1(\boldsymbol{x})+\lambda_2 f_2(\boldsymbol{x}), \tag{2.14}$$

where $\boldsymbol{\lambda}=(\lambda_1,\lambda_2)$ is the vector containing the unknown parameters and $\lambda_1+\lambda_2=1$. This mixture model covers situations where the underlying population is modeled as consisting of 2 distinct groups G_1, G_2 in some unknown proportions λ_1,λ_2, and where the conditional p.d.f of $\boldsymbol{X}$ given membership of the ith group G_i is $f_i(\boldsymbol{x})$.
Let $\boldsymbol{x}=(x_1,...,x_n)^T$ denote the observed random sample obtained from the mixture density (2.14)*. The log likelihood function for $\boldsymbol{\lambda}$ that can be formed from the observed data $\boldsymbol{x}$ is given by*

$$\log\mathcal{L}(\boldsymbol{\lambda}) = \sum_{i=1}^{n}\log\left(\lambda_1 f_1(x_i)+\lambda_2 f_2(x_i)\right). \tag{2.15}$$

On differentiating (2.15) *with respect to λ_1 and equating the result to zero, we obtain*

$$\sum_{i=1}^{n}\left(\frac{f_1(x_i)}{g(x_i;\widehat{\boldsymbol{\lambda}})}-\frac{f_2(x_i)}{g(x_i;\widehat{\boldsymbol{\lambda}})}\right)=0 \tag{2.16}$$

as the likelihood equation, which clearly does not yield an explicit solution for $\widehat{\boldsymbol{\lambda}}=(\widehat{\lambda}_1,\widehat{\lambda}_2)$. However, this problem can be posed as an incomplete-data (unobservable or missing data) one. This problem is overcome by the EM algorithm which will be introduced in the next subsection.

2.2.2 Example to introduce the EM algorithm

DLR Dempster et al. [1977] used a multinomial example to introduce the EM algorithm and that example has been subsequently used many times in the literature to illustrate various modifications and extensions of this algorithm. The idea of EM is that the observed data is viewed as being incomplete, then unobservable or missing data is added to achieve the complete-data. The complete-data log likelihood is then used in a step called E-step to find parameters at each iteration.

We will present the EM algorithm through applying the excellent idea for the mixture model in Example 3. Namely, we now introduce as the unobservable or missing data the vector

$$\boldsymbol{z} = (z_1, ..., z_n), \tag{2.17}$$

where z_i is a 2-dimensional vector of zero-one indicator variables and where $z_{ij} = (z_i)_j$ is one or zero according to whether x_i arose or did not arise from the jth component of the mixture $(j = 1, 2; i = 1, ..., n)$.

If these z_{ij} is observable, then the MLE of λ_j is simply given by

$$\frac{1}{n}\sum_{i=1}^{n} z_{ij} \ (j = 1, 2), \tag{2.18}$$

which is the proportion of the sample having arisen from the jth component of the mixture. On defining the complete-data vector $(\boldsymbol{x}, \boldsymbol{z})$ the complete-data log likelihood for $\boldsymbol{\lambda}$ has the multinomial form

$$\log \mathcal{L}(\boldsymbol{\lambda}) = \sum_{i=1}^{n} (z_{i1} \log(\lambda_1) + z_{i2} \log(\lambda_2)) + C, \tag{2.19}$$

where

$$C = \sum_{i=1}^{n} (z_{i1} \log f_1(x_i) + z_{i2} \log f_2(x_i)) \tag{2.20}$$

does not depend on $\boldsymbol{\lambda}$.

As (2.19) is linear in the unobservable data z_{ij}, the socalled **E-step** on the $(t+1)$th iteration simply requires the calculation of the current conditional expectation of Z_{ij} given the observed data $\boldsymbol{x}$, where Z_{ij} is the random variable corresponding to z_{ij}. Now

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\lambda}^{(t)}}(Z_{ij}|\boldsymbol{x}) &= \mathbb{P}_{\boldsymbol{\lambda}^{(t)}}(Z_{ij} = 1|\boldsymbol{x}) \\ &= z_{ij}^{(t)}, \end{aligned} \tag{2.21}$$

where by Bayes Theorem,

$$z_{ij}^{(t)} = p_{ij}^{(t)} = \frac{\lambda_j^{(t)} f_j(x_i)}{g(x_i; \boldsymbol{\lambda}^{(t)})} \tag{2.22}$$

for $j = 1, 2; i = 1, ..., n$. The quantity $p_{ij}^{(t)}$ is the posterior probability that the ith member of the sample with observed value x_i belongs to the jth component of the mixture.

The so called **M-step** on the $(t+1)$th iteration simply requires replacing each z_{ij} by $z_{ij}^{(t)}$ in (2.18) to give

$$\lambda_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} z_{ij}^{(t)}, \text{ for } j = 1, 2. \tag{2.23}$$

Thus in forming the estimate of λ_j on the $(t+1)$th iteration, there is a contribution from each observation x_i equal to its posterior probability of membership of the jth component of the mixture model. The EM solution therefore has an intuitively appealing interpretation.

The computation of the MLE of λ_j by direct maximization of the incomplete-data log likelihood function (2.15) requires solving the likelihood equation (2.16). The latter can be identified with the iterative solution (2.23) provided by the EM algorithm after some manipulation as follows. On multiplying throughout by $\widehat{\lambda}_j$ in equation (2.16), we have that

$$\sum_{i=1}^{n} \left(\widehat{p}_{ij} - \frac{\widehat{\lambda}_j}{\widehat{\lambda}_2} \widehat{p}_{i2} \right) = 0, \text{ for } j = 1, \tag{2.24}$$

where $\widehat{p}_{ij} = \frac{\widehat{\lambda}_j f_j(x_i)}{g(x_i;\widehat{\boldsymbol{\lambda}})}$. As (2.24) also holds for $j = 2$, we can sum over $j = 1, 2$ in (2.24) to give

$$\widehat{\lambda}_2 = \sum_{i=1}^{n} \widehat{p}_{i2}/n. \tag{2.25}$$

Substitution now of (2.25) into (2.24) yields

$$\widehat{\lambda}_j = \frac{1}{n} \sum_{i=1}^{n} \widehat{p}_{ij}. \tag{2.26}$$

The resulting equation (2.26) for the MLE $\widehat{\lambda}_j$ can be identified with the iterative solution (2.22). The latter solves the likelihood equation by substituting an initial value for λ_j into the right-hand side of (2.26), which yields a new estimate for λ_j, which in turn is substituted into the right-hand side of (2.26) to yield a new estimate, and so on until convergence.

Exemple 4 (A Numerical example)**.** *As a numerical example, we generated a random sample of* $n = 50$ *observations* $x_1, ..., x_n$ *from a mixture of two univariate normal densities with means* $\mu_1 = 0$ *and* $\mu_2 = 2$ *and common variance* $\sigma^2 = 1$ *in proportions* $\lambda_1 = 0.8$ *and* $\lambda_2 = 0.2$*. Starting the EM algorithm from* $\lambda_1^{(0)} = 0.5$*, it converged after 27 iterations to the solution* $\widehat{\lambda}_1 = 0.75743$*. The EM algorithm was stopped when*

$$|\lambda_1^{(t+1)} - \lambda_1^{(t)}| < 10^{-5}. \tag{2.27}$$

It was also started from the moment estimate given by

$$\widetilde{\lambda}_1 = (\overline{x} - \mu_2)/(\mu_1 - \mu_2) = 0.86815 \tag{2.28}$$

and, using the same stopping criterion, it converged after 30 iterations to $\widehat{\lambda}_1$*.*
In Table 2.1, we have listed the value of $\lambda_1^{(t)}$ *and of* $\log \mathcal{L}(\lambda_1^{(t)})$ *for various values of* t*. It can be seen that it is during the first few iterations that the EM algorithm makes most of its progress in reaching the maximum value of the log likelihood function.*

This method of moments can also be used for m small.

Iteration t	$\lambda_1^{(t)}$	$\log \mathcal{L}(\lambda_1^{(t)})$
0	0.50000	−91.87811
1	0.68421	−85.55353
2	0.70304	−85.09035
3	0.71792	−84.81398
4	0.72885	−84.68609
5	0.73665	−84.63291
6	0.74218	−84.60978
⋮	⋮	⋮
27	0.75743	−84.58562

Table 2.1 – Results of EM algorithm for example on Estimation of mixing proportions

2.2.3 The MM algorithm and EM algorithm

Every EM algorithm is a special case of the more general class of MM (Majorization-Minorization) optimization algorithms. An MM algorithm operates by creating a surrogate function that minorizes or majorizes the objective function. When the surrogate function is optimized, the objective function is driven uphill or downhill as needed. The original idea of the MM algorithm can be dated back at least to Ortega and Rheinboldt [1970] in the context of line search methods. De Leeuw and Heiser [1977] presented an MM algorithm for multidimensional scaling contemporary with the classic Dempster et al. [1977] paper on EM algorithms. The same idea kept reappearing under different guises in different areas until Hunter and Lange [2000] put forth "MM" as general framework. They stated the general principle, sketched various methods of majorization and proposed a variety of applications. In their view, MM algorithms are useful extensions of the class of EM algorithms and MM algorithms are easier to understand and sometimes easier to apply than EM algorithms.

MM algorithms exploit an optimization technique that extends the central idea of EM algorithms to situations not necessarily involving missing data nor even maximum likelihood estimation. The MM principle is based on the notion of (tangent) majorization. A real-value function of θ whose form depends on $\theta^{(t)}$, denote $h(\theta|\theta^{(t)})$, is said to *minorize* a real-value function $\mathcal{L}(\theta)$ at the point $\theta^{(t)}$ provided

$$h(\theta|\theta^{(t)}) \leq \mathcal{L}(\theta), \ \forall\theta \ \text{ and } \ h(\theta^{(t)}|\theta^{(t)}) = \mathcal{L}(\theta^{(t)}).$$

In other words, the surface $\mathcal{L}(\theta)$ is lower bounded by the surface $\theta \mapsto h(\theta|\theta^{(t)})$ and is tangent to it at the point $\theta = \theta^{(t)}$. The function $h(\theta|\theta^{(t)})$ is said to *majorize* $\mathcal{L}(\theta)$ at $\theta^{(t)}$ if

$$-h(\theta|\theta^{(t)}) \ \text{ minorizes } \ -\mathcal{L}(\theta) \ \text{ at } \theta^{(t)}.$$

Here $\theta^{(t)}$ represents the current iterate in a search of the surface $\mathcal{L}(\theta)$. In the majorization version of the MM algorithm, we maximize the surrogate minimizing function $h(\theta|\theta^{(t)})$ rather than the actual function $\mathcal{L}(\theta)$. If $\theta^{(t+1)}$ denotes the maximizer of $h(\theta|\theta^{(t)})$, then

one can show that the MM procedure forces $\mathcal{L}(\theta)$ uphill. Fig 2.2 shows the relations

$$\mathcal{L}(\theta^{(t)}) = h(\theta^{(t)}|\theta^{(t)}) \leq h(\theta^{(t+1)}|\theta^{(t)}) \leq \mathcal{L}(\theta^{(t+1)}).$$

With straightforward changes, in the minimization version of the MM algorithm, we minimize the surrogate majorizing function $h(\theta|\theta^{(t)})$. Thus, the acronym MM does double duty, serving as an abbreviation of both pairs *majorize-minimize* and *minorize-maximize*.

Let $\boldsymbol{X}$ be random vector which results from a parameterized family and $\boldsymbol{X}$ is associated to a missing data. The EM algorithm is an iterative procedure for maximizing log likelihood function $\mathcal{L}(\theta) = \log \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta})$. Assume that after the tth iteration the current estimate for $\boldsymbol{\theta}$ is given by $\boldsymbol{\theta}^{(t)}$. We wish to compute an updated estimate $\boldsymbol{\theta}$ such that $\mathcal{L}(\boldsymbol{\theta}) > \mathcal{L}(\boldsymbol{\theta}^{(t)})$ or we want to maximize the difference

$$\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}(\boldsymbol{\theta}^{(t)}) = \log \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}) - \log \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}^{(t)}).$$

Denote the hidden random vector by $\boldsymbol{Z}$ and a given realization by $\boldsymbol{z}$. The total probability $\log \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta})$ may be written in terms of the hidden variables $\boldsymbol{z}$ as

$$\mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta}).$$

Since $\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)})$ is a probability measure, we have that

$$\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}) \geq 0 \text{ and } \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}) = 1.$$

Then

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}(\boldsymbol{\theta}^{(t)}) &= \log \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta}) - \log \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}^{(t)}) \\
&= \log \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta}).\frac{\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)})}{\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)})} - \log \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}^{(t)}) \\
&= \log \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}).\frac{\mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)})} - \log \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}^{(t)}) \\
&\geq \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}).\log \frac{\mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)})} - \log \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}^{(t)}) \\
&= \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}).\log \frac{\mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)})\mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}^{(t)})} \\
&:= \Delta(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}).
\end{aligned}$$

Equivalently we may write

$$\mathcal{L}(\boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\theta}^{(t)}) + \Delta(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) := h(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}).$$

Additionally, observe that,

$$\begin{aligned}
h(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) &= \mathcal{L}(\boldsymbol{\theta}^{(t)}) + \Delta(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \\
&= \mathcal{L}(\boldsymbol{\theta}^{(t)}) + \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}). \log \frac{\mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta}^{(t)})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})}{\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)})\mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}^{(t)})} \\
&= \mathcal{L}(\boldsymbol{\theta}^{(t)}) + \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}). \log \frac{\mathbb{P}(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta}^{(t)})}{\mathbb{P}(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta}^{(t)})} \\
&= \mathcal{L}(\boldsymbol{\theta}^{(t)}) + \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}). \log 1 \\
&= \mathcal{L}(\boldsymbol{\theta}^{(t)}).
\end{aligned}$$

The function $h(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ is upper-bounded by the likelihood function $\mathcal{L}(\boldsymbol{\theta})$. The functions are equal at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$. The EM algorithm chooses $\boldsymbol{\theta}^{(t+1)}$ as the value of $\boldsymbol{\theta}$ for which $h(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ is a maximum. Since $\mathcal{L}(\boldsymbol{\theta}) \geq h(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ increasing $h(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ ensures that the value of the likelihood function $\mathcal{L}(\boldsymbol{\theta})$ is increased at each step. Our objective is to choose a values of $\boldsymbol{\theta}$ so that $\mathcal{L}(\boldsymbol{\theta})$ is maximized. Formally we have,

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\{h(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t+1)})\} \\
&= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \left\{ \mathcal{L}(\boldsymbol{\theta}^{(t)}) + \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}). \log \frac{\mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)})\mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta}^{(t)})} \right\} \\
&\quad \text{Now drop terms which are constant w.r.t. } \boldsymbol{\theta} \\
&= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \left\{ \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}). \log \mathbb{P}(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}|\boldsymbol{\theta}) + \text{constant}(\boldsymbol{\theta}^{(t)}) \right\} \\
&= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \left\{ \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}). \log \frac{\mathbb{P}(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{z}, \boldsymbol{\theta})} \frac{\mathbb{P}(\boldsymbol{z}, \boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{\theta})} + \text{constant}(\boldsymbol{\theta}^{(t)}) \right\} \\
&= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \left\{ \sum_{\boldsymbol{z}} \mathbb{P}(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}). \log \mathbb{P}(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta}) + \text{constant}(\boldsymbol{\theta}^{(t)}) \right\} \\
&= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \left\{ \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}}[\log \mathbb{P}(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta})] + \text{constant}(\boldsymbol{\theta}^{(t)}) \right\} \\
&:= \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \left\{ Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) + \text{constant}(\boldsymbol{\theta}^{(t)}) \right\}.
\end{aligned}$$

Given an arbitrary starting value $\boldsymbol{\theta}^{(0)}$, remind here the EM algorithm generates a sequence $(\boldsymbol{\theta}^{(t)})_{t\geq 1}$ by iterating the following steps:

1. **E-step:** `compute` $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}}[\log \mathbb{P}(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta})]$
2. **M-step:** `set` $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$.

The EM algorithm is actually a special case of the MM algorithm. If the log-likelihood of the observed data is $\mathcal{L}(\boldsymbol{\theta})$, and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ is the function created in the E-step. Let

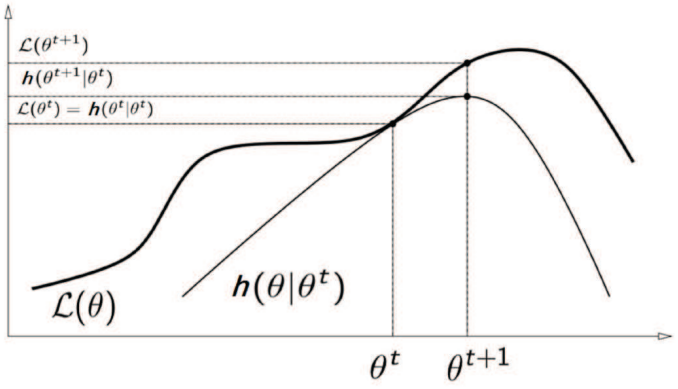


Figure 2.2 – MM principle

$h(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) + \text{constant}(\boldsymbol{\theta}^{(t)})$ such that $h(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ is minorize $\mathcal{L}(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^{(t)}$. Then the minorization

$$\mathcal{L}(\theta) \geq Q(\theta|\theta^{(t)}) + \mathcal{L}(\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)})$$

is the key to EM algorithm. Moreover, this insures an *ascent property*

$$\mathcal{L}(\theta^{(t+1)}) \geq \mathcal{L}(\theta^{(t)}).$$

2.2.4 EM algorithm for mixture model

In finite mixture models, the *complete data* associated with the actually observed sample $\boldsymbol{x}$ is $\boldsymbol{y} = (\boldsymbol{x}, \boldsymbol{Z})$, where to each individual (multivariate) observation $\boldsymbol{x}_i$ is associated an indicator variable Z_i denoting its component of origin. It is common to define $Z_i = (Z_{i1}, \ldots, Z_{im})$ with the indicator variables

$$Z_{ij} = \mathbb{I}\{\text{observation } i \text{ comes from component } j\}, \quad \sum_{j=1}^{m} Z_{ij} = 1.$$

From (2.1), this means that $\mathbb{P}_{\boldsymbol{\theta}}(Z_{ij} = 1) = \lambda_j$, and $(\boldsymbol{X}_i|Z_{ij} = 1) \sim f_j$, $j = 1, ..., m$. In this case, the expectation is w.r.t. the conditional distribution of the Z_{ij}'s,

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &:= \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{m} Z_{ij} \log \lambda_j f_j(\boldsymbol{x}_i)|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}\right] \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}\left[Z_{ij}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}\right] \log(\lambda_j f_j(\boldsymbol{x}_i)). \end{aligned} \tag{2.29}$$

Next we compute

$$\begin{aligned}\mathbb{E}\left[Z_{ij}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}\right] &= \mathbb{P}_{\boldsymbol{\theta}^{(t)}}(Z_{ij}=1|\boldsymbol{x}_i)\\ &= \frac{\mathbb{P}_{\boldsymbol{\theta}^{(t)}}(\boldsymbol{x}_i|Z_{ij}=1)\mathbb{P}_{\boldsymbol{\theta}^{(t)}}(Z_{ij}=1)}{\mathbb{P}_{\boldsymbol{\theta}^{(t)}}(\boldsymbol{x}_i)}\\ &= \frac{\lambda_j^{(t)} f_j^{(t)}(\boldsymbol{x}_i)}{\sum_{j'=1}^m \lambda_{j'}^{(t)} f_{j'}^{(t)}(\boldsymbol{x}_i)} := p_{ij}^{(t)}. \end{aligned} \tag{2.30}$$

This is Bayes formula and $p_{ij}^{(t)} := \mathbb{P}_{\boldsymbol{\theta}^{(t)}}(Z_{ij}=1|\boldsymbol{x}_i)$ is the *posterior probability* that the individual i comes from component j.

The M-step is a constrained maximization, which means that there are constraints on valid solutions not encoded in the function itself. Namely, maximizing $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ such that $\sum_{j=1}^m \lambda_j = 1$. Such problems can be solved using the method of Lagrange multipliers. To maximize a function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ on the open set in $\mathbb{R}$ subject to the constraint $\sum_{j=1}^m \lambda_j - 1 = 0$ it suffices to maximize the unconstrained function

$$F(\boldsymbol{\theta},\alpha) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) - \alpha\left(\sum_{j=1}^m \lambda_j - 1\right) \tag{2.31}$$

in the unusual unconstrained manner, by solving the system of equations

$$\begin{cases} \frac{\partial F(\boldsymbol{\theta},\alpha)}{\partial \lambda_j} = 0, \quad j = 1,\cdots,m \\ \frac{\partial F(\boldsymbol{\theta},\alpha)}{\partial \alpha} = 0 \end{cases}$$

namely,

$$\sum_{i=1}^n p_{ij}^{(t)} \frac{1}{\lambda_j} - \alpha = 0, \text{ and } \sum_{j=1}^m \lambda_j - 1 = 0.$$

It is equivalent to

$$\lambda_j = \frac{1}{\alpha}\sum_{i=1}^n p_{ij}^{(t)}, \text{ and } \frac{1}{\alpha}\sum_{j=1}^m\sum_{i=1}^n p_{ij}^{(t)} = 1$$

which leads to the solution $\alpha = \frac{1}{n}$ and $\lambda_j = \frac{1}{n}\sum_{i=1}^n p_{ij}^{(t)}$.

Therefore, the M-step for finite mixture models always looks partly the same: No matter what form the f_j's take, the updates to the mixing proportions are given by

$$\lambda_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^n p_{ij}^{(t)}, \quad \text{for } j = 1,\ldots,m, \tag{2.32}$$

The updates for the f_j's depend on the particular form of the component densities. In parametric mixtures (i.e. when the family $\mathcal{F}$ is completely specified by a finite-dimensional parameter), the updates of these parameters are often straightforward, and can be looked like weighted MLE estimates. This is the case for, e.g., Gaussian mixtures.

2.2.5 The EM algorithm for the parametric mixture model

The mixture-density parameter estimation problem is probably one of the most widely used applications of the EM algorithm in the computational pattern recognition community. In this case, we assume the following probabilistic model:

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{m} \lambda_j f_j(\boldsymbol{x}|\xi_j),$$

where the parameters are $\boldsymbol{\theta} = (\lambda_1, ..., \lambda_m, \xi_1, ..., \xi_m)$ such that $\sum_{j=1}^{m} \lambda_i = 1$ and each f_j is a density function parameterized by ξ_j. In other words, we assume we have m component densities mixed together with m unknown parameters ξ_j.

Given an arbitrary starting value $\boldsymbol{\theta}^{(0)}$, the EM algorithm is given by iterating the following steps:

1. **E-step:** `compute` $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$.
   In this case, from (2.29) and (2.30),

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}^{(t)} \log \lambda_j f_j(\boldsymbol{x}_i|\xi_j), \tag{2.33}$$

   where the *posterior probability* is given by

$$p_{ij}^{(t)} = \frac{\lambda_j^{(t)} f_j(\boldsymbol{x}_i|\xi_j^{(t)})}{\sum_{j'=1}^{m} \lambda_{j'}^{(t)} f_{j'}(\boldsymbol{x}_i|\xi_{j'}^{(t)})}. \tag{2.34}$$

2. **M-step:** `set` $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$.
   In the finite mixture model, the updated estimates $\lambda_j^{(t+1)}$ of mixing proportions λ_j are calculated independently of the updated estimate $\boldsymbol{\xi}^{(t+1)}$ of the parameter vector $\boldsymbol{\xi} = (\xi_1, \cdots, \xi_m)$, namely as (2.32)

$$\lambda_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{(t)}, \quad \text{for } j = 1, \ldots, m.$$

   Obviously, $\boldsymbol{\xi}^{(t+1)}$ is obtained as an appropriate root of

$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})}{\partial \boldsymbol{\xi}} = \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}^{(t)} \frac{\partial \log f_j(\boldsymbol{x}_i|\xi_j)}{\partial \boldsymbol{\xi}} = 0. \tag{2.35}$$

One nice feature of the EM algorithm is that the solution of (2.35) often exists in closed form, as is to be demonstrated for the normal mixture model hereafter.

Gaussian mixture model (Adapted in Plasse [2013])

We now turn our attention to the case when each f_j may be represented as

$$f_j(\boldsymbol{x}_i|\xi_j) = \frac{1}{(2\pi)^{r/2}\sqrt{\det \Sigma_j}} e^{-\frac{1}{2}(\boldsymbol{x}_i-\mu_j)^T\Sigma_j^{-1}(\boldsymbol{x}_i-\mu_j)},\ \xi_j = (\mu_j, \Sigma_j), \tag{2.36}$$

where as stated in the introduction $\boldsymbol{x}_i \in \mathbb{R}^r$, $\mu_j \in \mathbb{R}^r$ and Σ_j is a $r \times r$ symmetric positive definite matrix. We assume that we have a current approximate maximizer of our function Q which we denote by $\boldsymbol{\theta}^{(t)} = (\lambda_1^{(t)}, ..., \lambda_m^{(t)}, (\mu_1^{(t)}, \Sigma_1^{(t)}), ..., (\mu_m^{(t)}, \Sigma_m^{(t)}))$. Our goal is to now implement the maximization that occurs during the M-Step of the EM algorithm to obtain updated maximizers denoted by $\boldsymbol{\theta}^{(t+1)} = (\lambda_1^{(t+1)}, ..., \lambda_m^{(t+1)}, (\mu_1^{(t+1)}, \Sigma_1^{(t+1)}), ..., (\mu_m^{(t+1)}, \Sigma_m^{(t+1)}))$. The updated maximizers for our mixture proportions are derived first. The results were presented as in (2.32). Next, the derivations of $\mu_j^{(t+1)}$ and $\Sigma_j^{(t+1)}$ are implemented by (2.35).

Taking the log of Equation (2.36), ignoring any constant terms (since they disappear after taking derivatives), we get:

$$\log f_j(\boldsymbol{x}_i|\xi_j) = -\frac{1}{2}\log(\det \Sigma_j) - \frac{1}{2}(\boldsymbol{x}_i - \mu_j)^T\Sigma_j^{-1}(\boldsymbol{x}_i - \mu_j).$$

Substituting into the right side of Equation (2.35), we get:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{(t)} \frac{\partial}{\partial \boldsymbol{\xi}} \left(\log(\det \Sigma_j) + (\boldsymbol{x}_i - \mu_j)^T\Sigma_j^{-1}(\boldsymbol{x}_i - \mu_j)\right) = 0. \tag{2.37}$$

To find μ_j, replacing $\boldsymbol{\xi}$ by μ_j (i.e., taking the derivative with respect to μ_j), we get:

$$\sum_{i=1}^{n} \Sigma_j^{-1}(\boldsymbol{x}_i - \mu_j)p_{ij}^{(t)} = 0$$

with which we can easily solve for μ_j to obtain:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^n \boldsymbol{x}_i p_{ij}^{(t)}}{\sum_{i=1}^n p_{ij}^{(t)}} = \frac{\sum_{i=1}^n \boldsymbol{x}_i p_{ij}^{(t)}}{n\lambda_j^{(t+1)}}.$$

To find Σ_j, replacing $\boldsymbol{\xi}$ by Σ_j^{-1} and rewritting Equation (2.37) as:

$$\sum_{i=1}^{n}\sum_{k=1}^{m} p_{ik}^{(t)} \frac{\partial}{\partial \boldsymbol{\Sigma}_j^{-1}} \left(-\log(\det \Sigma_k^{-1}) + tr(\Sigma_k^{-1}(\boldsymbol{x}_i - \mu_k)(\boldsymbol{x}_i - \mu_k)^T)\right) = 0. \tag{2.38}$$

It is equivalent to

$$\begin{aligned} & \sum_{i=1}^{n} p_{ij}^{(t)} \left(-2\Sigma_j + \operatorname{diag}(\Sigma_j) + 2N_{ij} - \operatorname{diag}(N_{ij})\right. \\ = \ & -2\sum_{i=1}^{n} p_{ij}^{(t)} (\Sigma_j - N_{ij}) + \operatorname{diag}\left(\sum_{i=1}^{n} p_{ij}^{(t)}(\Sigma_j - N_{ij})\right) \\ = \ & -2\boldsymbol{S} + \operatorname{diag} \boldsymbol{S} = 0 \end{aligned}$$

where $N_{ij} = (\boldsymbol{x}_i - \mu_j)(\boldsymbol{x}_i - \mu_j)^T$ and where $\boldsymbol{S} = \sum_{i=1}^n p_{ij}^{(t)} (\Sigma_j - N_{ij})$. This implies that $\boldsymbol{S} = 0$. This gives

$$\sum_{i=1}^{n} p_{ij}^{(t)} (\Sigma_j - N_{ij}) = 0$$

or

$$\Sigma_j = \frac{\sum_{i=1}^n p_{ij}^{(t)} N_{ij}}{\sum_{i=1}^n p_{ij}^{(t)}} = \frac{\sum_{i=1}^n p_{ij}^{(t)} (\boldsymbol{x}_i - \mu_j)(\boldsymbol{x}_i - \mu_j)^T}{\sum_{i=1}^n p_{ij}^{(t)}}. \tag{2.39}$$

Summarizing, the Updated Parameter Estimates

$$\lambda_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{(t)},$$

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^n \boldsymbol{x}_i p_{ij}^{(t)}}{\sum_{i=1}^n p_{ij}^{(t)}},$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^n p_{ij}^{(t)} (\boldsymbol{x}_i - \mu_j^{(t+1)})(\boldsymbol{x}_i - \mu_j^{(t+1)})^T}{\sum_{i=1}^n p_{ij}^{(t)}}.$$

These updated parameter estimates are analog with the univariate case. A model-based approach is a way to deal with clustering problems. It consists in using certain models for clusters and attempting to optimize the fit between the data and the model. Typically the data are clustered using some assumed mixture modeling structure. Then the group memberships are learned in an unsupervised fashion. In practice, each cluster can be mathematically represented by a parametric distribution, like a Gaussian (univ- or multivariate case). The entire data set is therefore modeled by a mixture of these distributions. The advantages of model-based clustering with multivariate mixtures are the flexibility in choosing the component distributions, the well-studied statistical inference techniques available. Moreover, the mixture model covers the data well and one can obtain a density estimation for each cluster.

Celeux and Govaert [1995] especially analyzed the influence of the volumes of clusters in Gaussian parsimonious clustering models. Biernacki et al. [2006] examined model-based cluster with the Mixture Modeling MIXMOD software and included different information criteria for choosing a parsimonious model. Hennig [2010] proposed methods to decide whether and which Gaussian mixture components should be merged in order to interpret their union as cluster.

2.2.6 A semiparametric EM algorithm

In the univariate case, Bordes et al. [2007] first proposed a univariate semiparametric (and stochastic) "EM-like" algorithm for a location-shift semiparametric mixture model (2.6)

$$g_{\boldsymbol{\theta}}(x) = \sum_{j=1}^{m} \lambda_j f(x - \mu_j), \quad x \in \mathbb{R}, \quad \boldsymbol{\theta} = (\boldsymbol{\lambda}, \boldsymbol{\mu}, f).$$

where the pdf f itself is an unknown, even density, considered as a parameter which has to be estimated from the data x.

The novelty that is hidden behind the term EM-like is that the M step is not a genuine maximization step. It is a hybrid algorithm that introduces a nonparametric, Weighted Kernel Density Estimation (WKDE) step. This algorithm hence provides a kernel density estimate for f. It is also a stochastic algorithm since, at each iteration, each observation in the dataset is randomly assigned to one of the mixture components, the assignment being based on the posterior probabilities of component membership. This algorithm is simple to program and is applicable practically for any number m of components, even beyond the cases for which identifiability has been proved.

The parameter of the semiparametric model is $\boldsymbol{\theta} = ((\lambda_j, \mu_j)_{j=1,\cdots,m}, f) = (\boldsymbol{\phi}, f) \in \boldsymbol{\Theta} = \Phi \times \mathcal{F}$, where $\mathcal{F}$ is the set of continuous even pdf's over $\mathbb{R}$. In this framework, we still have that the pdf of the observed and complete data are

$$\begin{aligned} g_{\boldsymbol{\theta}}(x) = g(x|\boldsymbol{\theta}) &= \sum_{j=1}^{m} \lambda_j f(x - \mu_j), \\ h(y|\boldsymbol{\theta}) &= h((x,z)|\boldsymbol{\theta}) = \lambda_z f(x - \mu_z) \end{aligned}$$

and, formally, the log-likelihood associated to x for the parameter $\boldsymbol{\theta}$ is

$$\mathcal{L}_x(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log g(x_i|\boldsymbol{\theta}).$$

To design an EM-like algorithm which "mimic" the parametric version, we have to define, for a current value $\boldsymbol{\theta}^{(t)} = (\boldsymbol{\phi}^{(t)}, f^{(t)})$ of the parameter at iteration t, the operator

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}[\log h(y|\boldsymbol{\theta})|x, \boldsymbol{\theta}^{(t)}].$$

As in the parametric case, the expectation is taken with respect to the distribution of the y given x, for the value $\boldsymbol{\theta}^{(t)}$ of the parameter:

$$\boldsymbol{k}(y|x, \boldsymbol{\theta}^{(t)}) = \prod_{i=1}^{n} k(y_i|x_i, \boldsymbol{\theta}^{(t)}) = \prod_{i=1}^{n} k(z_i|x_i, \boldsymbol{\theta}^{(t)}),$$

where

$$k(j|x, \boldsymbol{\theta}^{(t)}) = \mathbb{P}(Z = j|x, \boldsymbol{\theta}^{(t)}) = \frac{\lambda_j^{(t)} f^{(t)}(x - \mu_j^{(t)})}{\sum_{j'=1}^{m} \lambda_{j'}^{(t)} f^{(t)}(x - \mu_{j'}^{(t)})}, \quad j = 1, \cdots, m.$$

Hence $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ is given by

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{n} \sum_{j=1}^{m} k(j|x, \boldsymbol{\theta}^{(t)})[\log(\lambda_j) + \log f(x_i - \mu_j)].$$

Bordes et al. [2006a] describe the flavor of the method in what can be considered as an "ideal situation". Assume that the complete data $y = (x, z)$ is available, and that $\boldsymbol{\theta}$ is known. Then a consistent estimate of f would be given by the following steps:

1. compute $\tilde{x} = (\tilde{x_1}, \cdots, \tilde{x_n})$, where $\tilde{x_i} = x_i - \mu_{z_i}, i = 1, \cdots, n$
2. compute a kernel density estimate using some kernel K and bandwidth h_n,

$$\hat{f}_{\tilde{x}}(u) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{u - \tilde{x}_i}{h_n}\right).$$

Assume now that the z are missing, but that the true parameter $\boldsymbol{\theta}$ is known. The difficulty then is to recover a sample from f be given a sample from $g_{\boldsymbol{\theta}}$. The allocation can only be deduced from the posterior probabilities $k(j|x, \boldsymbol{\theta})$. An "expectation strategy" following the EM principle:

$$\tilde{x}_i = x_i - \sum_{j=1}^{m} k(j|x, \boldsymbol{\theta})\mu_j, \quad i = 1, \cdots, n.$$

We may also use the maximum of the posterior probabilities, as it is usually done in classification algorithms based on EM:

$$\tilde{x}_i = x_i - \mu_{j_i^*}, \quad j_i^* = \underset{j \in \{1, \cdots, m\}}{\operatorname{argmax}}\, k(j|x, \boldsymbol{\theta}), \quad i = 1, \cdots, n.$$

Unfortunately, even with $\boldsymbol{\theta}$ known, none of these strategies return a sample f distributed, as it can be checked on simple explicit situations. To recover a sample from f , we need to simulate the ith allocation according to the posterior probabilities $(k(j|x, \boldsymbol{\theta}), \quad j = 1, \cdots, m)$ i.e. from a multinomial distribution of order 1:

- S-1: for $i = 1, \cdots, n$ simulate $Z(x_i, \boldsymbol{\theta}) \sim \mathcal{M}(1; (k(j|x, \boldsymbol{\theta}), \quad j = 1, \cdots, m)$
- S-2: set $\tilde{x}_i = x_i - \mu_{Z(x_i, \boldsymbol{\theta})}$,

where $Z(x, \boldsymbol{\theta}) \in \{1, \cdots, m\}$ and $\mu_{Z(x, \boldsymbol{\theta}) = \mu_j}$ when $Z(x, \boldsymbol{\theta}) = j$. Then they proved that this procedure returns a sample f distributed.

It is then possible to compute a kernel density estimate of f. Finally, the step $\boldsymbol{\theta}^{(t)} \to \boldsymbol{\theta}^{(t+1)}$ of the semiparametric EM algorithm (SP-EM) is defined by:

1. **E-step:** compute

$$k(j|x_i, \boldsymbol{\theta}^{(t)}) = \mathbb{P}(Z = j|x_i, \boldsymbol{\theta}^{(t)}) = \frac{\lambda_j^{(t)} f^{(t)}(x_i - \mu_j^{(t)})}{\sum_{j'=1}^{m} \lambda_{j'}^{(t)} f^{(t)}(x_i - \mu_{j'}^{(t)})}, \; i = 1, \ldots, n, \; j = 1, \ldots, m.$$

2. **S-step:**
   – S-1: for $i = 1, \ldots, n$, draw $Z^{(t+1)}(x_i, \boldsymbol{\theta}^{(t)}) \sim \mathcal{M}(1; (k(j|x_i, \boldsymbol{\theta}^{(t)}), \quad j = 1, \ldots, m)$
   – S-2: set $\tilde{x}_i^{(t+1)} = x_i - \mu^{(t)}_{Z^{(t+1)}(x_i, \boldsymbol{\theta}^{(t)})}$,

3. **Nonparametric step:**
   – kernel density estimate

$$\hat{f}_{\tilde{x}^{(t+1)}}(u) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{u - \tilde{x}_i^{(t+1)}}{h_n}\right);$$

– symetrization

$$f^{(t+1)}(u) = \frac{\hat{f}_{\tilde{x}^{(t+1)}}(u) + \hat{f}_{\tilde{x}^{(t+1)}}(-u)}{2}.$$

4. **M-step:** (parametric EM strategy to update the Euclidean parameter)

$$\lambda_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} k(j|x_i, \boldsymbol{\theta}^{(t)}),$$

$$\mu_j^{t+1} = \frac{\sum_{i=1}^{n} k(j|x_i, \boldsymbol{\theta}^{(t)}) x_i}{\sum_{i=1}^{n} k(j|x_i, \boldsymbol{\theta}^{(t)})}, \quad j = 1, \cdots, m.$$

2.2.7 A nonparametric EM algorithm (`npEM` algorithm) in multivariate case

As reviewed briefly in Subsection 2.1.3, the common restriction placed on $\mathcal{F}$ in is that each joint density $f_j(\cdot)$ is equal to the product of its marginal densities. Then, we have mixture model (2.5)

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jk}(x_{ik}).$$

Hettmansperger and Thomas [2000] have developed an estimation method, *the cutpoint approach*, that discretizes the continuous measurements by replacing each r-dimensional observation, for the conditionally i.i.d. model. Hettmansperger and Thomas [2000] treat the case in which joint density $f_j(.)$ is equal to the product of its marginal densities $f_j = \prod_{k=1}^{r} f_{jk}$ and consider the special case in which the density $f_{jk}(.)$ does not depend on k or $f_{j1}(.) = \cdots = f_{jr}(.)$– that is, in which the X_i are not only conditionally independent but identically distributed as well

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{j}(x_{ik}).$$

In some situations, the later assumption may be too restrictive. Thus, to encompass both the special case and the more general case, Benaglia et al. [2009a] introduced a more flexible and important model: They allowed that the coordinates of X_i are conditionally independent and that there exist blocks of coordinates that are also identically distributed. This model admitted for continuous component densities f_{ik}'s. Let b_k denote the block to which the kth coordinate belongs, where $1 \leq b_k \leq B$ and B is the total number of such blocks, then (2.5) is replaced by

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jb_k}(x_{ik}). \tag{2.40}$$

Benaglia et al. [2009a] proposed an algorithm for nonparametric estimation for finite mixtures of multivariate random vectors that strongly resembles a true EM algorithm – an EM-like algorithm: Suppose we are given initial values $\boldsymbol{\theta}^0 = (\boldsymbol{\lambda}^0, \boldsymbol{f}^0)$. Then for $t = 1, 2, \cdots$, we follow these three steps:

1. **E-step:** Calculate the "posterior" probabilities (conditional on the data and $\boldsymbol{\theta}^{(t)}$) of component inclusion,

$$p_{ij}^{(t)} = \frac{\lambda_j \prod_{k=1}^{r} f_{jb_k}^{(t)}(x_{ik})}{\sum_{j'=1}^{m} \lambda_{j'} \prod_{k=1}^{r} f_{j'b_k}^{(t)}(x_{ik})}, \quad \forall i = 1, \cdots, n \text{ and } j = 1, \cdots, m.$$

2. **M-step:**

$$\lambda_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{(t)}, \quad \text{for } j = 1, \ldots, m.$$

3. **Nonparametric density estimation step:** For any real u, define for each component $j \in \{1, \cdots, m\}$ and each block $\ell \in \{1, \cdots, B\}$

$$f_{j\ell}^{(t+1)}(u) = \frac{1}{nhC_\ell \lambda_j^{(t+1)}} \sum_{k=1}^{r} \sum_{i=1}^{n} p_{ij}^{(t)} \mathbb{I}_{\{b_k=\ell\}} K\left(\frac{u - x_{ik}}{h}\right),$$

where $K(.)$ is a kernel density function, h is a bandwidth chosen by the user, and

$$C_\ell = \sum_{k=1}^{r} \mathbb{I}_{\{b_k=\ell\}}$$

is the number of coordinates in the ℓth block.

This algorithm is flexible and can be extended to any number of mixture components and any number of coordinates of the multivariate observations. It is called `npEM` algorithm (non-parametric EM) by Benaglia et al. [2009a] and eliminates the stochasticity of the univariate algorithm from Bordes et al. [2007], but also relies on a weight kernel density estimation (WKDE) step for the updates of the f_{jk}'s. However, this algorithm lacks of theoretical justification because it has not been proved the monotonicity property of loglikelihood function. The `npEM` algorithm is available online from the Comprehensive R Archive Network (CRAN) in R package: `mixtools` (Young et al. [2009]).

2.3 Kernel density estimation (KDE)

Given a sufficiently large number of mixture components, a Gaussian mixture model can be used to approximate any density. If we associate a single Gaussian with every data point, we get what is called a kernel density estimate. This is a nonparametric density estimator. This method first finds a single kernel density estimate of the entire data, and then detect clusters by identifying modes or regions of high density in the estimated density. KDE is a widely used method of nonparametric density estimation. For instance, in the third step of `npEM` algorithm, the component estimate $f_j^{(t)}$ at tth iteration is obtained by a weighted nonparametric (kernel) density estimate. In this section, we look at kernel density estimation, where we approximate the true distribution by sticking a small weighted copy of a kernel pdf at each observed data point.

2.3.1 Discrete estimator and kernel estimator

Let X be a random variable with continuous distribution $F(x)$ and density $f(x) = \frac{d}{dx}F(x)$. The goal is to estimate $f(x)$ from a random sample $\{X_1, ..., X_n\}$.

The distribution function $F(x)$ is naturally estimated by the EDF

$$\widehat{F}(x) = n^{-1} \sum_{i=1}^{m} \mathbb{I}(X_i \leq x).$$

It might seem natural to estimate the density $f(x)$ as the derivative of $\widehat{F}(x)$, $\frac{d}{dx}\widehat{F}(x)$, but this estimator would be a set of mass points, not a density, and as such is not a useful estimate of $f(x)$.

Instead, consider a discrete derivative. For some small $h > 0$, let

$$\widehat{f}(x) = \frac{\widehat{F}(x+h) - \widehat{F}(x-h)}{2h}. \tag{2.41}$$

We can write this as

$$\begin{aligned} \frac{1}{2nh} \sum_{i=1}^{n} \mathbb{I}(x - h < X_i \leq x + h) &= \frac{1}{2nh} \sum_{i=1}^{n} \mathbb{I}\left(\frac{|X_i - x|}{h} \leq 1\right) \\ &= \frac{1}{nh} \sum_{i=1}^{n} K_{uni}\left(\frac{X_i - x}{h}\right), \end{aligned}$$

where

$$K_{uni}(u) = \begin{cases} \frac{1}{2}, & |u| \leq 1 \\ 0, & |u| > 1 \end{cases}$$

is the uniform density function on $[-1, 1]$.

The estimator $\widehat{f}(x)$ counts the percentage of observations which are closed to the point x. If many observations are near x, then $\widehat{f}(x)$ is large. Conversely, if only a few X_i are near x, then $\widehat{f}(x)$ is small.

This discrete estimator is not wholly satisfactory from the point of view of using density estimates for presentation. It follows from the definition that $\widehat{f}(x)$ is not a continuous function, but has jumps at the points $X_i \pm h$ and has zero derivative everywhere else. This gives the estimates a somewhat ragged character which is not only aesthetically undesirable, but, could provide the untrained observer with a misleading impression. Partly to overcome this difficulty, it is of interest to consider the generalization of the discrete estimator. Replace the above uniform density function by a *kernel function* K which satisfies the condition

$$\int_{\mathbb{R}} K(u)du = 1. \tag{2.42}$$

A *non-negative* kernel satisfies $K(u) \geq 0$ for all u. In this case, $K(u)$ is a probability density function. A *symmetric* kernel function satisfies $K(u) = K(-u)$ for all u. Most

nonparametric estimation uses symmetric probability density function, and we focus on this case. By analogy with the definition of the discrete estimator, the *kernel estimator* with kernel K is defined by

$$\widehat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right), \tag{2.43}$$

where the **bandwidth** h, also called the *smoothing parameter* or *window width* by some authors, controls the degree of smoothing.

The most commonly used kernels are special cases of the polynomial family

$$K_s(u) = \frac{(2s+1)!!}{2^{s+1}s!}(1-u^2)^s \mathbb{I}(|u| \leq 1), \tag{2.44}$$

where the double factorial means $(2s+1)!! = (2s+1) \times (2s-1) \times ... \times 5 \times 3 \times 1$. The Gaussian kernel is obtained by taking the limit as $s \to \infty$ after rescaling. The kernels with higher s are smoother, yielding estimates $\widehat{f}(x)$ which are smoother and possessing more derivatives. Estimates using the Gaussian kernel have derivatives of all orders.

For the purpose of nonparametric estimation the scale of the kernel is not uniquely defined. That is, for any kernel $K(u)$ we could have defined the alternative kernel $K^*(u) = b^{-1}K(u/b)$ for some constant $b > 0$. These two kernels are equivalent in the sense of producing the same density estimator, so long as the bandwidth is rescaled. That is, if $\widehat{f}(x)$ is calculated with kernel K and bandwidth h, it is numerically identically to a calculation with kernel K^* and bandwidth $h^* = h/b$.

2.3.2 Measures of discrepancy: mean-squared error and mean integrated square error

When considering estimation at a single point, a common and convenient measure of estimation precision is the mean-squared error (abbreviated MSE), defined by

$$\begin{aligned} MSE(\widehat{f}(x)) &= E\left(\widehat{f}(x) - f(x)\right)^2 \\ &= bias(\widehat{f}(x))^2 + var\left(\widehat{f}(x)\right), \end{aligned}$$

where $bias\widehat{f}(x) = E[\widehat{f}(x)] - f(x)$ and $var(\widehat{f}(x)) = E[\widehat{f}(x)^2] - E[\widehat{f}(x)]^2$. From the point of view of approximation, these bias and variance are estimated as two following steps:

First step: Estimation of bias

It is useful to observe that expectations of kernel transformations can be written as integrals which take the form of a convolution of the kernel and the density function:

$$E\left[\frac{1}{h}K\left(\frac{X_i - x}{h}\right)\right] = \int_{\mathbb{R}} \frac{1}{h}K\left(\frac{z-x}{h}\right) f(z)dz. \tag{2.45}$$

Kernel	**Equation**	$\kappa_2(K)$
Uniform	$K_0(u) = \frac{1}{2}\mathbb{I}(\lvert u\rvert \leq 1)$	$\frac{1}{3}$
Epanechnikov	$K_1(u) = \frac{3}{4}(1-u^2)\mathbb{I}(\lvert u\rvert \leq 1)$	$\frac{1}{5}$
Biweight	$K_2(u) = \frac{15}{16}(1-u^2)^2\mathbb{I}(\lvert u\rvert \leq 1)$	$\frac{1}{7}$
Triweight	$K_3(u) = \frac{35}{32}(1-u^2)^3\mathbb{I}(\lvert u\rvert \leq 1)$	$\frac{1}{9}$
Gaussians	$K_\phi(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{u^2}{2}}$	1

Table 2.2 – Common Second-Order Kernels

Using the change-of variables $u = (z-x)/h$, this equals

$$\int_{\mathbb{R}} K(u)f(x+hu)du. \tag{2.46}$$

By the linearity of the estimator we see

$$E[\widehat{f}(x)] = \frac{1}{n}\sum_{i=1}^{n} E\left[\frac{1}{h}K\left(\frac{X_i - x}{h}\right)\right] = \int_{\mathbb{R}} K(u)f(x+hu)du. \tag{2.47}$$

The last expression shows that the expected value is an average of $f(z)$ locally about x. This integral (typically) is not analytically solvable, so we approximate it using a Taylor expansion of $f(x+hu)$ in the argument hu, which is valid as $h \to 0$. Conveniently, we give some new definitions.

The *moments* of a kernel are $\kappa_j(K) = \int_{\mathbb{R}} u^j K(u)du$. The *order of a kernel*, ν, is defined as the order of the first non-zero moment. For example, if $\kappa_1(K) = 0$ and $\kappa_2 > 0$ then K is a second-order kernel and $\nu = 2$. If $\kappa_1(K) = \kappa_2(K) = \kappa_3(K) = 0$ but $\kappa_4(K) > 0$ then K is a fourth-order kernel and $\nu = 4$. The order of a symmetric kernel is always even. Symmetric non-negative kernels are second-order kernels. Common second-order kernels are listed in Table 2.2.

For a νth-order kernel we take the expansion out to the νth term

$$f(x+hu) = f(x) + f^{(1)}(x)hu + \frac{1}{2}f^{(2)}(x)h^2u^2 + \frac{1}{3!}f^{(3)}(x)h^3u^3 + ... + \frac{1}{\nu!}f^{(\nu)}(x)h^\nu u^\nu + o(h^\nu). \tag{2.48}$$

The remainder is of smaller order than h^ν as $h \to \infty$, which is written as $o(h^\nu)$. (This expansion assumes $f^{(\nu+1)}(x)$ exists). Integrating term by term and using $\int_{\mathbb{R}} K(u)du = 1$ and the definition $\int_{\mathbb{R}} u^j K(u)du = \kappa_j(K)$,

$$\begin{aligned}\int_{\mathbb{R}} K(u)f(x+hu)du &= f(x) + f^{(1)}(x)h\kappa_1(K) + \frac{1}{2}f^{(2)}(x)h^2\kappa_2(K) + \frac{1}{3!}f^{(3)}(x)h^3\kappa_3(K) \\ &\quad + ... + \frac{1}{\nu!}f^{(\nu)}(x)h^\nu\kappa_\nu(K) + o(h^\nu) \\ &= f(x) + \frac{1}{\nu!}f^{(\nu)}(x)h^\nu\kappa_\nu(K) + o(h^\nu),\end{aligned}$$

where the second equality uses the assumption that K is a νth-order kernel (so $\kappa_j(K) = 0$ for $j < \nu$).
That means that

$$\begin{aligned} E[\widehat{f}(x)] &= \frac{1}{n}\sum_{i=1}^{n} E\left[\frac{1}{h}K\left(\frac{X_i - x}{h}\right)\right] \\ &= f(x) + \frac{1}{\nu!}f^{(\nu)}(x)h^{\nu}\kappa_{\nu}(K) + o(h^{\nu}). \end{aligned}$$

The bias of $\widehat{f}(x)$ is then

$$bias(\widehat{f}(x)) = E[\widehat{f}(x)] - f(x) = \frac{1}{\nu!}f^{(\nu)}(x)h^{\nu}\kappa_{\nu}(K) + o(h^{\nu}). \tag{2.49}$$

For second-order kernels, this simplifies to

$$bias(\widehat{f}(x)) = E[\widehat{f}(x)] - f(x) = \frac{1}{2}f^{(2)}(x)h^{2}\kappa_{2}(K) + o(h^{4}). \tag{2.50}$$

The bias is increasing in the square of the bandwidth. Smaller bandwidths imply reduced bias. The bias is also proportional to the second derivative of the density $f^{(2)}(x)$. Intuitively, the estimator $\widehat{f}(x)$ smooths data local to $X_i = x$, so is estimating a smoothed version of $f(x)$. The bias results from this smoothing, and is larger the curvature in $f(x)$.

When higher-order kernels are used (and the density has enough derivatives), the bias is proportional to h^{ν}, which is of lower order than h^2. Thus the bias of estimates using higher-order kernels is of lower order than estimates from second-order kernels, and this is why they are called bias-reducing kernels. This is the advantage of higher-order kernels.

Second step: Estimation of the variance

Since the kernel estimator is a linear estimator, and $K\left(\frac{X_i - x}{h}\right)$ is i.i.d.,

$$\begin{aligned} var\left(\widehat{f}(x)\right) &= \frac{1}{nh^2}var\left(K\left(\frac{X_i - x}{h}\right)\right) \\ &= \frac{1}{nh^2}E\left[K\left(\frac{X_i - x}{h}\right)^2\right] - \frac{1}{n}\left(\frac{1}{h}E\left[K\left(\frac{X_i - x}{h}\right)\right]\right)^2. \end{aligned}$$

From the analysis of bias it is known that $\frac{1}{h}E\left[K\left(\frac{X_i - x}{h}\right)\right] = f(x) + o(1)$ so the second term is $O\left(\frac{1}{n}\right)$.
For the first term, write the expectation as an integral, make a change-of-variables and a first-order Taylor expansion

$$\begin{aligned} \frac{1}{h}E\left[K\left(\frac{X_i - x}{h}\right)^2\right] &= \frac{1}{h}\int_{\mathbb{R}} K\left(\frac{z - x}{h}\right)^2 f(x)dz \\ &= \int_{\mathbb{R}} K(u)^2 f(x + hu)du \\ &= \int_{\mathbb{R}} K(u)^2 (f(x) + O(h))du \\ &= f(x)R(K) + O(h), \end{aligned}$$

where $R(K) = \int_{\mathbb{R}} K(u)^2 du$ is the roughness of the kernel. Together, we see

$$var\left(\widehat{f}(x)\right) = \frac{f(x)R(K)}{nh} + O\left(\frac{1}{n}\right). \tag{2.51}$$

The remainder $O\left(\frac{1}{n}\right)$ is of smaller order than the $O\left(\frac{1}{nh}\right)$ leading term, since $h^{-1} \to \infty$.

Approximation of the MSE

From above estimators, the approximation of MSE is given by

$$\begin{aligned} MSE(\widehat{f}(x)) &\approx \left(\frac{1}{\nu!} f^{(\nu)}(x) h^{\nu} \kappa_{\nu}(K)\right)^2 + \frac{f(x)R(K)}{nh} \\ &= \frac{\kappa_{\nu}^2(K)}{(\nu!)^2} f^{(\nu)}(x)^2 h^{2\nu} + \frac{f(x)R(K)}{nh} \\ &:= AMSE(\widehat{f}(x)). \end{aligned}$$

Since this approximation is based on asymptotic expansions this is called the asymptotic mean-squared-error (AMSE). Note that it is a function of the sample size n, the bandwidth h, the kernel function (through κ_{ν} and $R(K)$), and varies with x as $f^{(\nu)}(x)$ and $f(x)$ vary.

Notice as well that the first term (the squared bias) is increasing in h and the second term (the variance) is decreasing in nh. For $MSE(\widehat{f}(x))$ to decline as $n \to \infty$ both of these terms must get small. Thus as $n \to \infty$ we must have $h \to 0$ and $nh \to \infty$. That is, the bandwidth must decrease, but not at a rate faster than sample size. This is sufficient to establish the pointwise consistency of the estimator. That is, for all $x, \widehat{f}(x) \to_p f(x)$ as $n \to \infty$.

MISE and its approximation

The most widely used way of placing a measure on the global accuracy of $\widehat{f}$ as an estimator of f is the *mean integrated square error* defined by

$$MISE = \mathbb{E}\left[\int_{\mathbb{R}} \left(\widehat{f}(x) - f(x)\right)^2 dx\right]. \tag{2.52}$$

The asymptotic mean integrated squared error (AMISE) can be defined by

$$\begin{aligned} AMISE &= \int_{\mathbb{R}} AMSE(\widehat{f}(x)) dx \\ &= \frac{\kappa_{\nu}^2(K)}{(\nu!)^2} R\left(f^{(\nu)}\right) h^{2\nu} + \frac{R(K)}{nh}, \end{aligned}$$

where $R\left(f^{(\nu)}\right) = \int_{\mathbb{R}} \left(f^{(\nu)}\right)^2 dx$ is the *roughness* of $f^{(\nu)}$.

2.3.3 Choosing bandwidth

The AMISE formula expresses the MSE as a function of h: The value of h which minimizes this expression is called the *asymptotically optimal bandwidth*. The solution is found by taking the derivative of the AMISE with respect to h and setting it equal to zero:

$$\begin{aligned}\frac{d}{dh}AMISE &= \frac{d}{dh}\left(\frac{\kappa_\nu^2(K)}{(\nu!)^2}R\left(f^{(\nu)}\right)h^{2\nu}\right)+\frac{R(K)}{nh}\\ &= 2\nu h^{2\nu-1}\frac{\kappa_\nu^2(K)}{(\nu!)^2}R\left(f^{(\nu)}-\frac{R(K)}{nh^2}\right):=0\end{aligned}$$

with solution

$$\begin{aligned}h_0 &= C_\nu(K,f)n^{-1/(2\nu+1)}\\ C_\nu(K,f) &= R\left(f^{(\nu)}\right)^{-1/(2\nu+1)}A_\nu(K)\\ A_\nu(K) &= \left(\frac{(\nu!)^2R(K)}{2\nu\kappa_\nu^2(K)}\right)^{1/(2\nu+1)}.\end{aligned} \tag{2.53}$$

The optimal bandwidth is propotional to $n^{-1/(2\nu+1)}$. We say that the optimal bandwidth is of order $O\left(n^{-1/(2\nu+1)}\right)$. For second-order kernels the optimal rate is $O\left(n^{-1/5}\right)$. For higher-order kernels the rate is slower, suggesting that bandwidths are generally larger than for second-order kernels. The intuition is that since higher-order kernels have smaller bias, they can afford a larger bandwidth.

The constant of proportionality $C_\nu(K,f)$ depends on the kernel through the function $A_\nu(K)$ (which can be calculated from Table 2.2), and the density through $R(f^{(\nu)})$ (which is unknown).

If the bandwidth is set to h_0, then with some simplification the AMISE equals

$$AMISE_0(K) = (1+2\nu)\left(\frac{R\left(f^{(\nu)}\right)\kappa_\nu^2(K)R(K)^{2\nu}}{(\nu!)^2(2\nu)^{2\nu}}\right)^{1/(2\nu+1)}n^{-2\nu/(2\nu+1)}. \tag{2.54}$$

For second-order kernels, this equals

$$AMISE_0(K) = \frac{5}{4}\left(\kappa_2^2(K)R(K)^4R(f^{(2)})\right)^{1/5}n^{-4/5}. \tag{2.55}$$

As ν gets large, the convergence rate approaches the parametric rate n^{-1}. Thus, at least asymptotically, the slow convergence of nonparametric estimation can be mitigated through the use of higher-order kernels.

Reference to a standard distribution

A very easy and natural approach is to use a standard family of distributions (second-order kernel) to assign a value to the term $R(f^{(2)})$ in the expression (2.53) for the ideal

bandwidth. For example, the normal distribution with variance σ^2 has , setting ϕ to be the standard normal density,

$$\begin{aligned}\int_{\mathbb{R}} f^{(2)}(x)^2 dx &= \sigma^{-5} \int_{\mathbb{R}} \phi^{(2)}(x)^2 dx \\ &= \frac{3}{8}\sqrt{\pi}\sigma^{-5} \approx 0.212\sigma^{-5}. \end{aligned} \tag{2.56}$$

If a Gaussian kernel is being used, then the window width obtained from (2.53) would be, substituting the value (2.56),

$$\begin{aligned} h_0 &= (4\pi)^{-1/10}\frac{3}{8}\sqrt{\pi}\sigma n^{-1/5} \\ &= \left(\frac{4}{3}\right)^{1/5} \sigma n^{-1/5} \approx 1.06\sigma n^{-1/5}. \end{aligned} \tag{2.57}$$

A quick way of choosing the smoothing parameter, therefore, would be to estimate σ from the data and then to substitute the estimate into (2.57). Either the usual sample standard deviation or a more robust estimator of σ could be used.

Better results can be obtained using a robust measure of spread. Formula (2.57) written in terms of the interquartile range IQR of the underlying normal distribution becomes

$$h_0 = 0.79 \times \mathrm{IQR} \times n^{-1/5}. \tag{2.58}$$

Unfortunately, using (2.58) for the bimodal distributions makes matters worse, because it oversmooths even further. The best of both worlds can be obtained using the adaptive estimate of spread

$$\mathcal{A} = \min\left\{SD, \frac{\mathrm{IQR}}{1.34}\right\} \tag{2.59}$$

instead of σ in the formula (2.57) with SD is standard deviation. This will cope well with the unimodal densities and will not do too badly if the density is moderately bimodal. Another modification, which will improve matters further, is to reduce the factor 1.06 in (2.57); for instance, the choice, for a Gaussian kernel,

$$h = 0.9\mathcal{A}n^{-1/5} \tag{2.60}$$

will yield a mean integrated square error within 10% of the optimum for all the t-distributions considered, for the log-normal with skewness up to about 1.8, and for the normal mixture with separation up to 3 standard deviations. This rule is commonly used in practice and it is often referred to as Silverman's reference bandwidth or Silverman's rule of thumb (Silverman [1986], page 48). From (2.59) and (2.60) the default bandwidth is

$$h = 0.9\min\left\{SD, \frac{\mathrm{IQR}}{1.34}\right\}.n^{-1/5}.$$

In `R`, Silverman's bandwidth is invoked by bw = "bw.nrd0".

2.3.4 Multivariate Density Estimation

Multivariate Kernel Density Estimation has been used since a long time in multivariate data analysis (see, e.g.,Scott [1992]). Considering a single sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ iid from a pdf f over $\mathbb{R}^r$, the general form of a multivariate KDE is

$$\hat{f}_H(\boldsymbol{u}) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{K}_H(\boldsymbol{u} - \boldsymbol{x}_i), \tag{2.61}$$

where for $\boldsymbol{u} = (u_1, u_2, \ldots, u_r)^t \in \mathbb{R}^r$

$$\boldsymbol{K}_H(\boldsymbol{u}) = |H|^{-1/2} \boldsymbol{K}(H^{-1/2}.\boldsymbol{u}),$$

and where $\boldsymbol{K}$ is a multivariate kernel function, H is a symmetric positive definite $r \times r$ "bandwidth matrix", and $H^{-1/2}.\boldsymbol{u}$ is the usual matrix product.

With a full bandwidth matrix, the corresponding kernel smoothing is equivalent to pre-rotating the data by an optimal amount and then using a diagonal bandwidth matrix. The bandwidth matrix can be restricted to a class of positive definite diagonal matrices, and then the corresponding kernel function is often a product kernel (e.g. Gaussian). In this case, $H = \text{diag}(h_1^2, h_2^2, \ldots, h_r^2)$ where h_k denotes the kth coordinate bandwidth. Then $|H|^{1/2} = h_1 \cdots h_r$ so that (denoting informally by $\boldsymbol{K}$ for the multivariate kernels and K for univariate kernels)

$$\boldsymbol{K}_H(\boldsymbol{u}) = \frac{1}{h_1 \cdots h_r} \boldsymbol{K}\left(\frac{u_1}{h_1}, \ldots, \frac{u_r}{h_r}\right) = \prod_{k=1}^{r} \frac{1}{h_k} K\left(\frac{u_k}{h_k}\right).$$

In the simplest case $H = \text{diag}(h^2, \ldots, h^2)$, we have

$$\boldsymbol{K}_H(\boldsymbol{u}) = \frac{1}{h^r} K\left(\frac{1}{h}\,\boldsymbol{u}\right).$$

As in the univariate case, $\widehat{f}(\boldsymbol{u})$ has the property that it integrates to one, and is non-negative if $\boldsymbol{K}(\boldsymbol{u}) \geq 0$.

2.4 Maximum Smoothed Likelihood for Multivariate Mixtures

Under the assumption of conditional independence, remind here that the mixture density evaluated at the point $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{ir})^\top$ can be presented as in model (2.5):

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jk}(x_{ik}).$$

To estimate parameters $\boldsymbol{\theta}$ in a finite mixture of completely unspecified multivariate components in at least three dimensions of model (2.5), Benaglia et al. [2009a] proposed an algorithm that strongly resembles a true EM algorithm. Indeed, this EM-like algorithm lacks theoretical justification (as we mentioned in Section 2.2.7). Levine et al. [2011] corrected this problem and introduced an alternative algorithm which possesses a desirable descent property just as any EM algorithm does.

2.4.1 Smoothing the log-density

Assume that Ω is a compact subset of $\mathbb{R}^r$ and define the linear vector function space

$$\mathcal{F} = \{\boldsymbol{f} = (f_1, \cdots, f_m)^\top : 0 < f_j \in L_1(\Omega), \log f_j \in L_1(\Omega), j = 1, \cdots, m\}.$$

The assumption of compact support may appear somewhat limiting from a theoretical point of view, but it is not problematic from a practical point of view because of the bounded property of a dataset.

Define a smoothing operator $\mathcal{S}$ and a nonlinear smoothing operator $\mathcal{N}$ for any function $f \in L_1(\Omega)$ by

$$\mathcal{S}f(x) = \int_\Omega K_h(\boldsymbol{x} - \boldsymbol{u}) f(\boldsymbol{u}) d\boldsymbol{u},$$

$$\mathcal{N}f(\boldsymbol{x}) = \exp\{(\mathcal{S}f)(\boldsymbol{x})\} = \exp \int_\Omega K_h(\boldsymbol{x} - \boldsymbol{u}) \log f(\boldsymbol{u}) d\boldsymbol{u}.$$

where $K(.)$ denote some kernel density function on the real line. The product kernel function $K(\boldsymbol{u}) = \prod_{k=1}^r K(u_k)$ and its rescaled version $k_h(\boldsymbol{u}) = h^{-r} \prod_{k=1}^r K(h^{-1}u_k)$. This operator $\mathcal{N}$ is strictly concave, and it is also multiplicative in the sense that $\mathcal{N}f_j = \prod_k \mathcal{N}f_{jk}$ (see Eggermont and Lariccia [1999]).

Then, Levine et al. [2011] introduce the finite mixture operator

$$\mathcal{M}_{\boldsymbol{\lambda}} \boldsymbol{f}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{j=1}^m \lambda_j f_j(\boldsymbol{x}),$$

$$\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{j=1}^m \lambda_j \mathcal{N} f_j(\boldsymbol{x}).$$

Let $g(\boldsymbol{x})$ now represent a known target density function, define the following functional of $\boldsymbol{\theta}$ (and, implicitly, g):

$$\mathcal{L}(\boldsymbol{\theta}) = \int_\Omega g(\boldsymbol{x}) \log \frac{g(\boldsymbol{x})}{[\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}](\boldsymbol{x})}.$$

The goal is to find a minimizer of $\mathcal{L}(\boldsymbol{\theta})$ subject to the assumptions that each f_{jk} is a univariate density function and $\boldsymbol{\lambda}$ satisfies $\sum_{j=1}^m \lambda_j = 1,\ \lambda_j \geq 0$. An immediate consequence is that $\mathcal{L}(\boldsymbol{\theta})$ can be viewed as a penalized Kullback-Leibler distance between $g(\boldsymbol{x})$ and $(\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f})(\boldsymbol{x})$:

$$\mathcal{L}(\boldsymbol{\theta}) = D(g|(\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f})) + \int g(\boldsymbol{x}) d\boldsymbol{x} - \sum_{j=1}^m \lambda_j \int \mathcal{N} f_j(\boldsymbol{x}) d\boldsymbol{x},$$

where $-\lambda_j \int \mathcal{N} f_j(\boldsymbol{x}) d\boldsymbol{x}$ is a penalization term (see Eggermont and Lariccia [1999]).

Levine et al. [2011] defined an iterative algorithm for the `npEM` algorithm of Benaglia et al. [2009a]. It possesses a descent property with respect to the functional $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{f})$; that is, we wish to ensure that the value of $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{f})$ cannot increase from one

iteration to the next. Let $(\boldsymbol{\lambda}^0, \boldsymbol{f}^0)$ denote the current parameter values in an iterative algorithm. Define a functional $b^0(\boldsymbol{\lambda}, \boldsymbol{f})$ that, when shifted by a constant, majorizes $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{f})$– i.e.,

$$b^0(\boldsymbol{\lambda}, \boldsymbol{f}) + C^0 \geq \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{f}), \text{ with equality when } (\boldsymbol{\lambda}, \boldsymbol{f}) = (\boldsymbol{\lambda}^0, \boldsymbol{f}^0).$$

For $j = 1, \cdots, m$, let

$$w_j^0(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{\lambda_j^0 \mathcal{N} f_j^0(\boldsymbol{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0} \mathcal{N} \boldsymbol{f}^0(\boldsymbol{x})}, \quad \sum_{j=1}^m w_j^0 = 1$$

and

$$\begin{aligned} b^0(\boldsymbol{\lambda}, \boldsymbol{f}) &\stackrel{\text{def}}{=} -\int g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x}) \log[\lambda_j \mathcal{N} f_j(\boldsymbol{x})] d\boldsymbol{x} \\ &= -\sum_{j=1}^m \sum_{k=1}^r \int \int K_h(x_k - u) g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) \log f_{jk}(u) du d\boldsymbol{x} \\ &\quad -\sum_{j=1}^m \log \lambda_j \int g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x} \\ &= -\sum_{j=1}^m \sum_{k=1}^r b_{jk}^0(f_{jk}) + b_j^0(\lambda_j). \end{aligned}$$

Note that $b^0(\boldsymbol{\lambda}, \boldsymbol{f})$ separates the parameters from each other, in the sense that it is the sum of separate functions of the individual f_{jk} and λ_j.

Subject to the constraint $\sum_j \lambda_j = 1$, it is not hard to minimize $b^0(\boldsymbol{\lambda}, \boldsymbol{f})$ with respect to the $\boldsymbol{\lambda}$ parameter: For each j, the minimizer is

$$\widehat{\lambda}_j = \frac{\int g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x}}{\sum_{j=1}^m \int g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x}} = \int g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x}. \tag{2.62}$$

For $j = 1, \cdots, m$ and $k = 1, \cdots, r$

$$\widehat{f}_{jk} = \alpha_{jk} \int K_h(x_k - u) g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x}, \tag{2.63}$$

where α_{jk} is a constant chosen so that $\int \widehat{f}_{jk}(u) dt = 1$ is the unique (up to changes on a set of Lebesgue measure zero) density function minimizing $b_{jk}^0(.)$.

From the convexity of the negative logarithm function, Levine et al. [2011] proved that

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{f}) - \mathcal{L}(\boldsymbol{\lambda}^0, \boldsymbol{f}^0) \leq b^0(\boldsymbol{\lambda}, \boldsymbol{f}) - b^0(\boldsymbol{\lambda}^0, \boldsymbol{f}^0).$$

Since each individual piece of the $b^0(.)$ function is minimized by the corresponding piece of $(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{f}}) = (\widehat{\lambda}_j, \widehat{f}_{jk})_{j=1,\cdots,m,\ k=1,\cdots,r}$ then

$$\mathcal{L}(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{f}}) - \mathcal{L}(\boldsymbol{\lambda}^0, \boldsymbol{f}^0) \leq b^0(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{f}}) - b^0(\boldsymbol{\lambda}^0, \boldsymbol{f}^0) \leq 0,$$

which proves the descent property

$$\mathcal{L}(\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{f}}) \leq \mathcal{L}(\boldsymbol{\lambda}^0, \boldsymbol{f}^0).$$

2.4.2 Inference for the parameters of nonparametric mixture model

Given a simple random sample $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ distributed according to the $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ density defined in Equation 2.5. Letting $\tilde{G}_n(.)$ denote the empirical distribution function of the sample and ignoring the term $\int g_{\boldsymbol{\theta}}(\boldsymbol{x}) \log g_{\boldsymbol{\theta}}(\boldsymbol{x}) d\boldsymbol{x}$ that does not involve any parameters, a discrete version of $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{f})$ is

$$\mathcal{L}_n(\boldsymbol{\lambda}, \boldsymbol{f}) \stackrel{\text{def}}{=} \int \log \frac{1}{[\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}](\boldsymbol{x})} d\tilde{G}_n(\boldsymbol{x}) = -\sum_{i=1}^{n} \log[\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}](\boldsymbol{x}_i).$$

Levine et al. [2011] show that the following algorithm results in an EM algorithm in which the value of $\mathcal{L}_n(.)$ decreases at each iteration: Given initial values $(\boldsymbol{\lambda}^0, \boldsymbol{f}^0)$, iterate the following three steps for $t = 1, 2, \cdots$

1. **E-step:** Define, for each i and j,

$$w_{ij}^{(t)} = \frac{\lambda_j^{(t)} \mathcal{N} f_j^{(t)}(\boldsymbol{x}_i)}{\mathcal{M}_{\boldsymbol{\lambda}^{(t)}} \mathcal{N} \boldsymbol{f}^{(t)}(\boldsymbol{x}_i)} = \frac{\lambda_j^{(t)} \mathcal{N} f_j^{(t)}(\boldsymbol{x}_i)}{\sum_{j'=1}^{m} \lambda_{j'} \mathcal{N} f_{j'}^{(t)}(\boldsymbol{x}_i)}.$$

2. **M-step, part 1:** Set

$$\lambda_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} w_{ij}^{(t)}, \quad \text{for } j = 1, \cdots, m.$$

3. **M-step, part 2:** For each j and k, let

$$f_{jk}^{(t+1)}(u) = \frac{\sum_{i=1}^{n} w_{ij}^{(t)} K_h(u - x_{ik})}{\sum_{i=1}^{n} w_{ij}^{(t)}} = \frac{1}{nh\lambda_j^{(t+1)}} \sum_{i=1}^{n} w_{ij}^{(t)} K\left(\frac{u - x_{ik}}{h}\right).$$

With regard to the convergence properties of the algorithm, if we hold $\boldsymbol{\lambda}$ fixed and repeatedly iterate equation (2.63), then the sequence of $\boldsymbol{f}$ functions converges to a global minimizer of $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{f})$ for that value of $\boldsymbol{\lambda}$ (see Appendix of Levine et al. [2011]).

If we allow that the coordinates of $\boldsymbol{X}_i$ are conditionally independent and that there exists blocks of coordinates that are also identically distributed, B is the total number of such blocks and let b_k denote the block index of the kth coordinate, where $1 \leq b_k \leq B$, then equation (2.5) is replaced by (2.40):

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jb_k}(x_{ik}).$$

The non-linear smoothing operator $\mathcal{N}$ applied to f_j is simply $\mathcal{N} f_j = \prod_{k=1}^{r} \mathcal{N} f_{jb_k}$, and definitions of $\mathcal{M}_{\boldsymbol{\lambda}} \boldsymbol{f}$ and $\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}$ are unchanged. The algorithm can easily be adapted for handling the block structure with the second part of the M-step becomes

3'. **M-step, part 2:** For each component j and block $\ell \in \{1, \cdots, B\}$, let

$$\begin{aligned} f_{j\ell}^{(t+1)} &= \frac{\sum_{k=1}^{r}\sum_{i=1}^{n} w_{ij}^{(t)} \mathbb{I}_{\{b_k=\ell\}} K_h(u - x_{ik})}{\sum_{k=1}^{r}\sum_{i=1}^{n} w_{ij}^{(t)} \mathbb{I}_{\{b_k=\ell\}}} \\ &= \frac{1}{nh\lambda_j^{(t+1)} C_\ell} \sum_{k=1}^{r}\sum_{i=1}^{n} w_{ij}^{(t)} \mathbb{I}_{\{b_k=\ell\}} K\left(\frac{u - x_{ik}}{h}\right), \end{aligned}$$

where $C_\ell = \sum_{k=1}^{r} \mathbb{I}_{\{b_k=\ell\}}$ is the number of coordinates in the ℓth block.

This algorithm is implemented by the function `npMSL` in the version of the publicly available `R` (R Core Team [2016]) package called `mixtools`.

There is still the question of asymptotic convergence rates. Empirical studies in Benaglia et al. [2009a] are suggestive of rates of convergence of the original `npEM` algorithm, though no theoretical result on this subject is yet known. Levine et al. [2011] demonstrated that their new algorithm may be used to optimize a particular objective function. This result is not a definitive proof of consistency. We have not yet convergence in the statistical sense, ie when $n \to \infty$, of the estimator towards the true value of $\boldsymbol{\theta}$ since the smoothed version is optimizing a smoothed loglikelihood, not the true one. It will perhaps be possible to establish such results in the future.

Chapter 3

Nonparametric mixture models with conditionally independent multivariate component densities

3.1 Introduction

Model (2.40) which is proposed by Benaglia et al. [2009a] has required the conditionally independent coordinates and there exit blocks of coordinates that are also identically distributed. However this assumption is not always satisfied in all cases. We introduce here a dataset in which model (2.40) and `npEM` algorithm of Benaglia et al. [2009a] does not work.

Breast cancer is the most common invasive cancer in females worldwide. Machine learning applications are vast; one such particular application to be investigated is in regards to classifying whether a breast tumor is malignant or benign. In fact, the medical literature is already becoming rich in such methods, with the potential goal of submitting patients to fewer extensive testing. The Breast Cancer Wisconsin Diagnostic (WDBC) dataset, which was obtained from the University of Wisconsin Hospitals, Madison, saves the Diagnostic together with 30 features related to Breast Cancer in Wisconsin. The features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Malignant breast tumors were detected from a set of benign (-) and malignant (+) samples. The comprehensive dataset utilized is available from the Breast Cancer Wisconsin (Diagnostic) Dataset on the UC Irvine Machine Learning Repository through the UW CS ftp server. The dataset is fairly rich in examples, considering $n = 569$ patients. It consists of a matrix with 32 columns, where the first such column is the patient ID and so ignored in this study and the second column is the label M for malignant and B for benign. Ten real-valued features computed for each cell nucleus: Radius, Texture, Perimeter, Area, Smoothness, Compactness, Concavity, Concave points, Symmetry and Fractal dimension. These features are computed from a digitized image of a breast mass. The mean, standard error, and "worst" (mean of the three largest values) of these features are computed for each image, resulting in a total of 30 features in the remaining 30 columns. The class distribution is given by 357 benign samples (62.74%) and 212 malignant samples (37.26%).

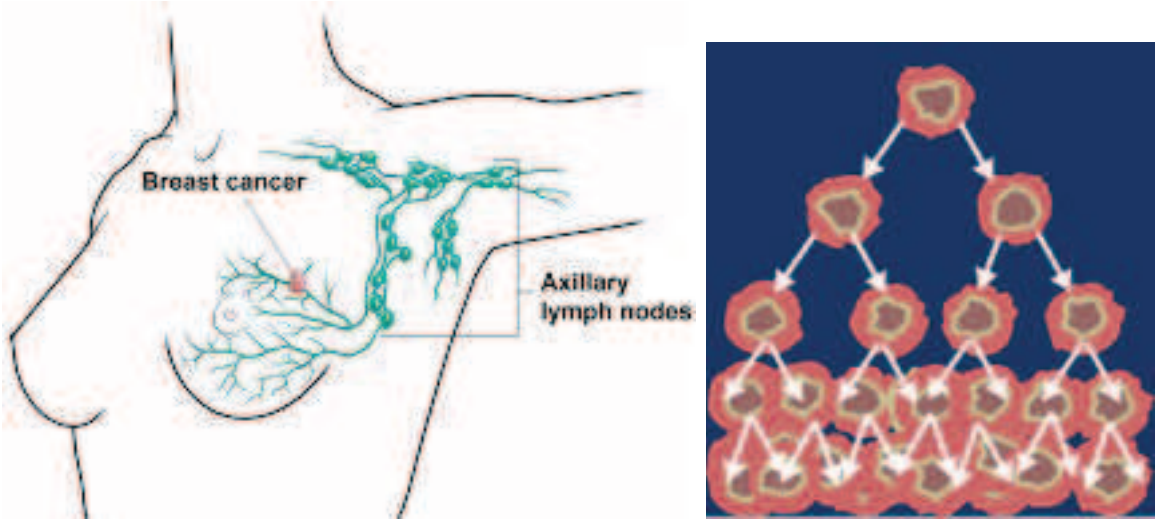


Figure 3.1 – Breast Cancer.

We first observed the ten "mean" features of WDBC data. This $r = 10$ dimensional dataset can be consider as a mixture of $m = 2$ components (Malignant/Benign), $r = 10$ coordinates. With the conditional independence assumption on the subpopulation or component, an identifiable model in the multivariate case can be defined as in (2.5) and a nonparametric EM algorithm (npEM) can be applied for this model (see Benaglia et al. [2009a]). With this kind of model, the dependence only comes from the mixture. A simple graphical exploration of the data shows that there are some obvious correlations across coordinates, not due to a mixture. Fig. 3.2 displays such dependencies among the ten mean features, for instance group of 3 features: radius, perimeter and area or compactness, concavity and number of concave points.

Our motivation in view of such datasets, is to relax the conditional independence of coordinates. Then, we consider such multivariate conditionally independent blocks instead of just coordinates. This chapter describes a new nonparametric mixture model that extends model (2.5) in the sense that it allows for conditionally independent *multivariate and nonparametric* component densities. Importantly, this extension allows for dependence structures within multivariate subsets of coordinates, apart from the dependence induced by the mixture that is the unique dependence allowed in model (2.5). Note that the idea of using conditionally independent multivariate subsets of variables itself is not new in the world of usual parametric mixtures; see, e.g., Hunt and Jorgensen [2003]. But the idea there is usually motivated by specific modelling needs, or for reducing the number of parameters in the covariance matrices of the component distributions. Our objective here is motivated by the need to extend the currently available nonparametric mixture models from the recent literature.

We present this model in Section 3.2, and verify in Section 3.3 that its parameters are identifiable using results from Allman et al. [2009] that go beyond the conditionally independent univariate case. We then focus on statistical estimation of the parameters in Section 3.4. We propose a new "EM-like" algorithm called `mvnpEM` since it relies – and is a multivariate (mv) per block extension of – the `npEM` algorithm introduced by Benaglia et al. [2009a]. Like the EM-like algorithms presented in this introduction, our

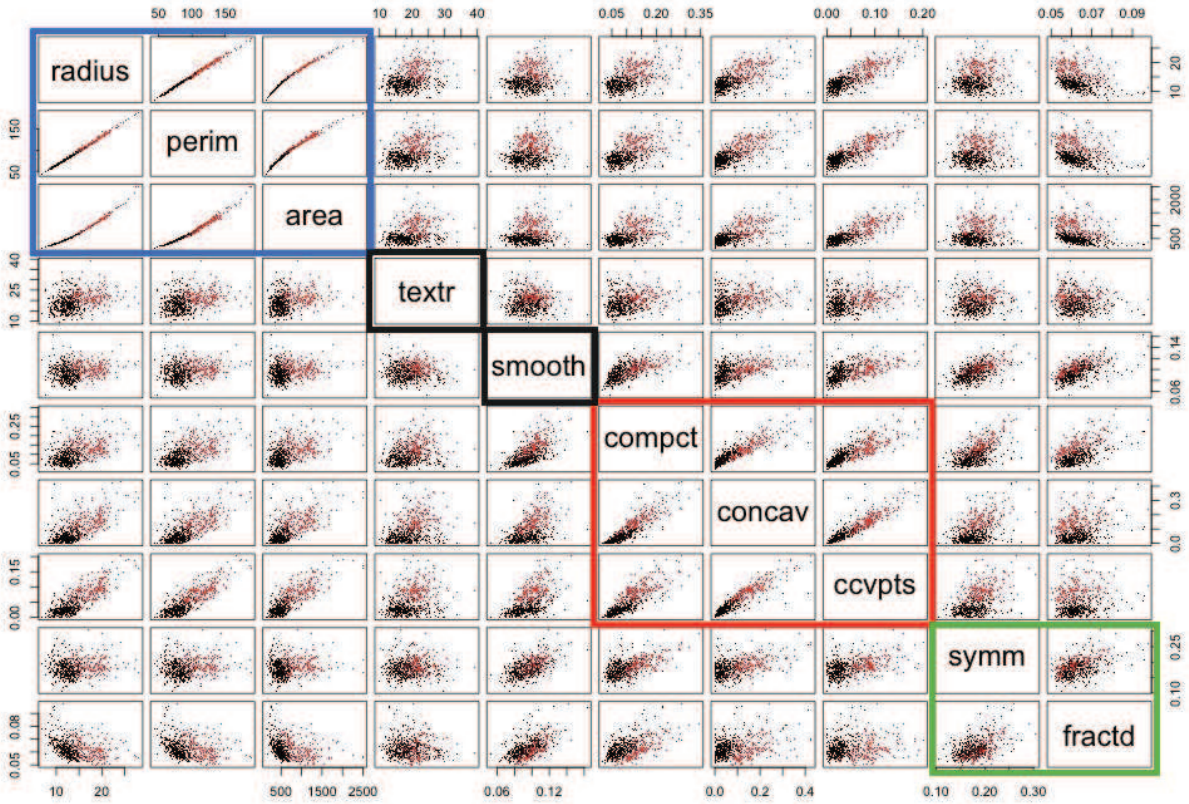


Figure 3.2 – The pairs plot of the first 10 features, colored by diagnostic. The 5 colored rectangles show a data-driven possible dependencies.

algorithm requires a weighted kernel density estimation step, which turns out here to be a multivariate WKDE. We thus describe possible bandwidth selection strategies for this WKDE in Section 3.4.2. Section 3.5 is devoted to implementation considerations and a study of the algorithm through large scale Monte-Carlo simulations. Section 3.6 describes an analysis, using our model, of the WDBC dataset. The perspective there is unsupervised model-based clustering, illustrating the potential usefulness of our new mixture model approach relaxing the conditional independence assumption.

3.2 Nonparametric mixture with multivariate blocks

We assume now that each joint density f_j is equal to the product of B multivariate densities that will correspond to conditionally independent multivariate *blocks* in the mixture model. Let the set of indices $\{1, ..., r\}$ be partitioned into B disjoint subsets s_ℓ, i.e. $\{1, ..., r\} = \bigcup_{l=1}^{B} s_l$, where $2 \leq B < r$ is the total number of such blocks, and d_ℓ is the number of coordinates in ℓth block, i.e. the lth block dimension. Actually, we will impose $B \geq 3$ in practice since from Allman et al. [2009] and the identifiability discussion in section 3.3 there is little hope to have an identifiable model for less than 3 independent blocks.

Here, the indices i, j, k and ℓ denote a generic individual, component, coordinate, and block, $1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq r$ and $1 \leq \ell \leq B$ (m, r, B and n stand for the number of mixture components, repeated measurements, blocks, and the sample size). Each f_j is equal to the product of the $f_{j\ell}$'s, where $f_{j\ell}$ is the multivariate density function for jth component and ℓth block. Then model (2.1) becomes

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{l=1}^{B} f_{j\ell}(x_{is_\ell}), \tag{3.1}$$

where $x_{is_\ell} = \{x_{ik}, k \in s_\ell\}$ is the multivariate variable which have its coordinates in the ℓth block. Hence this model assumes independence of blocks of multivariate densities, conditional on the subpopulation from which each observation is drawn. This is a main difference in comparison with model (2.5) introduced by Hall and Zhou [2003] assuming conditional independence: here the dependence structure does not come only from the mixture structure, since an additional within-block dependence is allowed. This model thus brings more flexibility with respect to the conditional independence assumption, that is in some applications a shortcoming of model (2.5) (see, e.g., discussion on actual data in Section 3.1 and in Section 3.6).

When all blocks are of size 1 (univariate blocks), then $B = r$ and the model is exactly model (2.5). Thus, to have at least one multivariate block of size ≥ 2, we assume $B < r$ in the sequel. Note that "block" has a different meaning in Benaglia et al. [2009a] and successive works on smoothed versions like Chauveau et al. [2015]; in these works block means a group of coordinates sharing a same *univariate* density for component j, allowing for more parsimonious models motivated by some actual applications from psychometrics.

As reviewed briefly, Hall et al. [2005] explored the identifiability question related to model (2.5) with univariate conditionally independent marginals. They also suggest that

a similar result *could* be achievable for conditionally independent blocks of multivariate densities, that is precisely our model (3.1). Then Allman et al. [2009] proved a collection of identifiability results, based on a representation of some latent variable model in terms of 3-way contingency tables. Next section provides the proof more details, and a survey-like shorter description for application to model (2.5) can be found in Chauveau et al. [2015].

3.3 Identifiability of the mixture with multivariate blocks

Hall et al. [2005] stated that an identifiability result similar to the one they claimed for model (2.5) with univariate conditionally independent marginals, could be achievable for conditionally independent blocks of multivariate densities. Then Allman et al. [2009] proved more generally a collection of identifiability results, based on a representation of some latent variable model in terms of 3-way contingency tables.

Their work describes a 3-way contingency table that cross-classifies a sample of n individuals with respect to three polytomous variables, the kth of which has a state space $\{1, \cdots, \kappa_k\}$. This classification can also be described in terms of the latent structure model. Assume that there is a latent (unobservable) variable Z with values in $\{1, \cdots, m\}$. Let us suppose that each of the individuals is known to belong to one of m latent classes and, conditionally on knowing the exact class j, $\quad j = 1, \cdots, m$, the 3 observed variables are mutually independent. Then latent class structure explains relationships among the categorical variables that we observe through the contingency table.

For a more detailed explanation, some algebraic notation is needed. For $k = 1, 2, 3$, let A_k be a matrix of size $m \times \kappa_k$, with $\boldsymbol{a}_j^k = (a_j^k(1), \cdots, a_j^k(\kappa_k))$ being the jth row of A_k. Later, we will see that $a_j^k(l)$ is the probability that the kth variable is in the lth state, conditional on the observation coming from the jth mixture component. Let $A_1 \times A_2 \times A_3$ be the $\kappa_1 \times \kappa_2 \times \kappa_3$ tensor defined by

$$[A_1, A_2, A_3] = \sum_{j=1}^{m} \boldsymbol{a}_j^1 \otimes \boldsymbol{a}_j^2 \otimes \boldsymbol{a}_j^3. \tag{3.2}$$

Using simpler language, the tensor $[A_1, A_2, A_3]$ is a 3-dimensional array whose element with coordinates (u, v, w) is a sum of products of elements of matrices A_k, $\quad k = 1, 2, 3$, with column numbers u, v and w, respectively, added up over all of m rows:

$$[A_1, A_2, A_3]_{u,v,w} = \sum_{j=1}^{m} a_j^1(u) a_j^2(v) a_j^3(w).$$

Such a tensor describes exactly the probability distribution in a finite latent class model with three observed variables. To see why this is the case, imagine that there is some latent variable Z that takes positive integer values from 1 to some $m > 1$ and each of the n individuals belongs to one of m latent class. If the 3 observed variables are mutually independent when the specific latent class j, $\quad 1 \leq j \leq m$, is known, we have a mixture of m components with each component being a product of finite measures and probabilities $\lambda_j := \mathbb{P}(Z = j)$, $\quad j = 1, \cdots, m$ being the mixing probabilities. Now, let the jth row of

the matrix A_k be the vector of probabilities of the kth variable conditioned on belong to jth class $\boldsymbol{p}_j^k = \mathbb{P}(X_k = .|Z = j)$. Choose one of the three matrices (say, A_1) and define $\tilde{A}_1 = diag(\boldsymbol{\lambda})A_1$, where $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_m)^\top$ is a vector describing the distribution of the latent class variable Z. Then, the (u, v, w) element of the tensor $[\tilde{A}_1, A_2, A_3]$ is the unconditional probability $\mathbb{P}(X_1 = u, X_2 = v, X_3 = w)$ and, therefore, the joint probability distribution in such a model is exactly described by the tensor 3.2.

Define the Kruskal rank of a matrix A, $rank_K A$, as the largest number I of rows such that every set of I rows of A is independent. The following result was established by Kruskal in the mid-1970s.

Theorem 1. *Let $I_k = rank_K A_k$. If*

$$I_1 + I_2 + I_3 \geq 2m + 2 \tag{3.3}$$

then $[A_1, A_2, A_3]$ uniquely determines the A_k, up to simultaneous permutation and rescaling of rows.

Kruskal's result is very general and is a cornerstone of several subsequent results establishing identifiability criteria for various latent structure models with multiple observed variables. The one that follows most directly is the identifiability result of finite mixtures of finite measure products. We refer to the model described above as the $m-$class, $r = 3-$feature model with state space $\{1, ..., \kappa_1\} \times \{1, ..., \kappa_2\} \times \{1, ..., \kappa_3\}$, and denote it by $\mathcal{M}(m; \kappa_1, \kappa_2, \kappa_3)$. The equivalence between the distributions of $3-$variate latent class models and $3-$tensors, combined with the fact that rows of stochastic matrices sum to 1, Theorem 1 gives the following reformulation.

Corollary 1. *Consider the model $\mathcal{M}(m; \kappa_1, \kappa_2, \kappa_3)$. Suppose all entries of $\boldsymbol{\lambda}$ are positive. For each $k = 1, 2, 3$, let A_k denote the matrix whose rows are $\boldsymbol{p}_j^k$, $j = 1, ..., m$, and let I_k denote its Kruskal rank. Then if*

$$I_1 + I_2 + I_3 \geq 2m + 2 \tag{3.4}$$

the parameters of the model are uniquely identifiable, up to label swapping.

Allman et al. [2009] expressed from Kruskal's theorem on $3-$variate models, to a similar one for $r-$variate models by combining into 3 agglomerate variables, so that Kruskal's result can be applied.

Theorem 2. *Consider the model $\mathcal{M}(m; \kappa_1, \kappa_2, \kappa_3)$ where $r \geq 3$. Suppose there exists a tripartition of the set $S = \{1, ..., r\}$ into three disjoint nonempty subsets S_1, S_2, S_3, such that if $\tau_l = \prod_{k \in S_l} \kappa_k$ then*

$$\min(m, \tau_1) + \min(m, \tau_2) + \min(m, \tau_3) \geq 2m + 2. \tag{3.5}$$

Then model parameters are generically identifiable, up to label swapping. Moreover, the statement remains valid when the mixing proportions $\{\lambda_j\}_{1 \leq j \leq m}$ are held and positive.

The above inequality is coming from the property: for any fixed choice of a positive integer $I_k \leq \min(m, \kappa_k)$, those $m \times \kappa_k$ matrices A_k, whose Kruskal rank is strictly less than I_k, form a proper algebraic variety (see the details in Allman et al. [2009]).

Let us recall that we are specifically interested in finite mixtures of nonparametric measure products. We consider a nonparametric model of finite mixtures of m probability distributions. Each distribution is specified as a measure μ_j on $\mathbb{R}^r, 1 \leq j \leq m$. Assume that the dimensionality r (the number of classification variables) is at least 3. The kth marginal of μ_j is denoted μ_j^k. As before, let Z be the variable defining the latent structure of the model with values in $\{1, ..., m\}$ and $\mathbb{P}(Z = j) = \lambda_j$ for any $j = 1, ..., m$. Then, the mixture model becomes

$$\mathcal{P} = \sum_{j=1}^{m} \lambda_j \mu_j = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} \mu_j^k. \tag{3.6}$$

This model implies that the r variates are, yet again, independent conditional on a latent structure. The next theorem proves identifiability of the model's parameters under a mild and explicit regularity condition on $\mathcal{P}$, as soon as there are at least 3 variates and m is known.

Theorem 3. *(**Theorem 8 in Allman et al. [2009]**) Let $\mathcal{P}$ be a mixture of nonparametric measure products as defined in* (3.6) *and, for every variate $k \in \{1, ..., r\}$, assume the marginal measures $\{\mu_j^k\}_{1 \leq j \leq m}$ are linearly independent in the sense that the corresponding (univariate) distribution functions satisfy no nontrivial linear relationship. Then, if the number of variables $r \geq 3$, the parameters $\{\lambda_j, \mu_j^k\}_{1 \leq j \leq m, 1 \leq k \leq r}$ are uniquely identifiable from $\mathcal{P}$, up to label swapping.*

The proof of Theorem 3 is making a judicious use of cut points to discretize the distribution, and then using Kruskal's work. The idea is to construct a binning of 3 random variables at a time only, beginning first with the random variables X_1, X_2 and X_3, using $\kappa_k - 1 \in \mathbb{N}$ cut points for X_k, $k = 1, 2, 3$, consider a partition of $\mathbb{R}$ into κ_k consecutive intervals $\{I_k^l\}_{1 \leq l \leq \kappa}$ and the random variable $Y_k = \{\mathbb{I}_{X_k \in I_k^1}, ..., \mathbb{I}_{X_k \in I_k^{\kappa_k}}\}$, where $\mathbb{I}_{\{A\}}$ denotes the indicator function of set A. Allman et al. [2009] proved it is able to choose the cut points $u_1 < u_2 < ... < u_{\kappa_k - 1}$ so for general enough and to have well-chosen partitions $\{I_k^l\}_{1 \leq l \leq \kappa_k}$ of $\mathbb{R}$ for recovering the measure μ_j^k, $k = 1, 2, 3$, $1 \leq j \leq m$. That is able to construct partitions that involve any chosen cut points in such a fashion that Kruskal's result in the form of Corollary 1 will apply. Note that an earlier result linking identifiability with linear independence of the densities to be mixed appears in literature in a parametric context. The linear independence of the probability distributions $\{\mu_j^k\}_{1 \leq j \leq m}$ is equivalent to linear independence of the c.d.f.s $\{F_j\}_{1 \leq j \leq m}$. The proof needs the following lemma.

Lemma 1. *Let $\{F_j\}_{1 \leq j \leq m}$ be linearly independent functions on $\mathbb{R}$. Then there exists some $\kappa \in \mathbb{N}$ and real numbers $u_1 < u_2 < ... < u_{\kappa - 1}$ such that the vectors*

$$\{F_j(u_1), F_j(u_2), ..., F_j(u_{\kappa - 1}), 1\}_{1 \leq j \leq m} \tag{3.7}$$

are linearly independent.

And relying only on the binned observed variables $\{Y_1, Y_2, Y_3\}$, the identifiability of the proportions λ_j and the probability measures μ_j^k can be infer. Repeating the same procedure with the random variables X_1, X_2, X_4 and getting the set $\{\mu_1, \mu_2, \mu_3, \mu_4\}$ can be recover, up to a relabeling of the groups. Adding a new random variable at a time finally gives the result.

This result also generalizes to nonparametric mixture models where at least three blocks of variates are independent conditioned on the latent structure. It is exactly our models which was described above. Remind here that if we let the set of indices $\{1, ..., r\}$ be partitioned into B disjoint subsets s_ℓ, i.e. $\{1, ..., r\} = \cup_{\ell=1}^{B} s_\ell$, where $3 \leq B \leq r$ is the total number of such blocks, and d_ℓ is the ℓth block dimension and the μ_j^ℓ be absolutely continuous probability measures on $\mathbb{R}^{d_\ell}$. With $r = \sum_{l=1}^{B} d_\ell$, consider the mixture distribution on $\mathbb{R}^r$ given by

$$\mathcal{P} = \sum_{j=1}^{m} \lambda_j \prod_{\ell=1}^{B} \mu_j^\ell. \tag{3.8}$$

Theorem 4. *(**Theorem 9 in Allman et al. [2009]**) Let $\mathcal{P}$ be a mixture of the form (3.8), such that for every $\ell \in \{1, ..., B\}$, the measures $\{\mu_j^\ell\}_{1 \leq j \leq m}$ on $\mathbb{R}^{d_\ell}$ are linear independent. Then, if $B \geq 3$, the parameters $\{\lambda_j, \mu_j^\ell\}_{1 \leq j \leq m, 1 \leq \ell \leq B}$ are strictly identifiable from $\mathcal{P}$, up to label swapping.*

Allman et al. [2009] proceeded much as in the proof of Theorem 3, but construct a binning into product intervals. For instance, if X is two dimensional, constructing $Y = \{\mathbb{I}_{\{X \in I^1 \times J^1\}}, ..., \mathbb{I}_{\{X \in I^\kappa \times J^\kappa\}}\}$, where $\{J^\ell\}_{1 \leq l \leq \kappa}$ is a second partition of $\mathbb{R}$ into $\kappa \in \mathbb{N}$ consecutive intervals. Lemma 1 generalizes to the following.

Lemma 2. *Let $\{F_j\}_{1 \leq j \leq m}$ be linearly independent functions on $\mathbb{R}^b$. There exists some κ, and b collections of real numbers $u_1^\ell < u_2^\ell < ... < u_{\kappa-1}^\ell$, for $1 \leq \ell \leq b$, such that the m row vectors composed of the values*

$$\{F_j(u_{j_1}^1, ..., u_{j_b}^b) | j_1, ..., j_b \in \{1, ..., \kappa\}\}, \text{ for } 1 \leq j \leq m \tag{3.9}$$

are linearly independent.

The equivalence between linear independence of the probability distributions and corresponding multidimensional c.d.f.'s remains valid, and conclude the proof in the same way as the last theorem. The details can be found in Allman et al. [2009].

3.4 Estimating the parameters

The algorithm we propose is an extension of the original `npEM` algorithm that was designed for estimation in the multivariate mixture model (2.5). The EM principle is first applied in the E-step, i.e. computation of the posterior probabilities given the current value $\boldsymbol{\theta}^{(t)}$ of the parameter. The EM machinery is also applied straightforwardly for the M-step of the scalar parameters that are only the weights $\boldsymbol{\lambda}$. Then a nonparametric WKDE is applied to update the component densities per blocks. The main difference is that in this model, we need multivariate density estimates. This is also why this algorithm becomes "EM-like", since kernel density estimation is not a genuine maximization step.

3.4.1 A multivariate npEM algorithm (mvnpEM)

Given initial values $\boldsymbol{\theta}^{(0)} = (\boldsymbol{\lambda}^{(0)}, \boldsymbol{f}^{(0)})$, the `mvnpEM` algorithm consists in iterating the following steps:

1. **E-step:** Calculate the posterior probabilities (conditional on the data and $\boldsymbol{\theta}^{(t)}$), for each $i = 1, \ldots, n$ and $j = 1, \ldots, m$:

$$p_{ij}^{(t)} := \mathbb{P}_{\boldsymbol{\theta}^{(t)}}(Z_{ij} = 1 | \boldsymbol{x}_i) = \frac{\lambda_j^{(t)} f_j^{(t)}(\boldsymbol{x}_i)}{\sum_{j'=1}^{m} \lambda_{j'}^{(t)} f_{j'}^{(t)}(\boldsymbol{x}_i)}, \tag{3.10}$$

   where $f_j^{(t)}(\boldsymbol{x}_i) = \prod_{\ell=1}^{B} f_{j\ell}^{(t)}(x_{is_\ell})$.

2. **M-step for λ:**

$$\lambda_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{(t)}, \quad j = 1, \ldots, m. \tag{3.11}$$

3. **Nonparametric kernel density estimation step:** For any $\boldsymbol{u}$ in $\mathbb{R}^{d_\ell}$, define for each component $j \in \{1, \ldots, m\}$ and block $\ell \in \{1, \ldots, B\}$

$$f_{j\ell}^{(t+1)}(\boldsymbol{u}) = \frac{1}{n\lambda_j^{(t+1)}} \sum_{i=1}^{n} p_{ij}^{(t)} \boldsymbol{K}_{H_{j\ell}}(\boldsymbol{u} - x_{is_\ell}), \tag{3.12}$$

   where $K_{H_{j\ell}}$ is a multivariate kernel density function, typically Gaussian, and $H_{j\ell}$ is a symmetric positive definite $d_\ell \times d_\ell$ matrix known as the bandwidth matrix. This matrix may depend on the ℓth block and jth component, and even on the tth iteration, as it will be precised in the next Section.

3.4.2 Bandwidth selection in multivariate KDE

The central decision in the nonparametric density estimation step of both the `npEM` and `mvnpEM` algorithm is the selection of an appropriate value for the (scalar or matrix) bandwidth or smoothing parameter. Following Benaglia et al. [2009a], we first simply use a single fixed bandwidth for all components per coordinate within each block, selected by default according to a rule of thumb from Silverman [1986] (see also Section 2.3). We then investigate a (often) more appropriate strategy defining iterative and per component and coordinate bandwidths by adapting Silverman's rule of thumb as in Benaglia et al. [2011].

Forgetting for now about blocks and components, considering the general multivariate KDE (2.61) in the case of bandwidth matrix $H = \mathrm{diag}(h_1^2, h_2^2, \ldots, h_r^2)$ where h_k denotes the kth coordinate bandwidth. The multivariate kernel is the product of univariate kernels (see more detail in Section 2.3.4):

$$\boldsymbol{K}_H(\boldsymbol{u}) = \frac{1}{h_1 \cdots h_r} \boldsymbol{K}\left(\frac{u_1}{h_1}, \ldots, \frac{u_r}{h_r}\right) = \prod_{k=1}^{r} \frac{1}{h_k} K\left(\frac{u_k}{h_k}\right),$$

where $\boldsymbol{u} = (u_1, u_2, \ldots, u_r)^t \in \mathbb{R}^r$ and denote $\boldsymbol{K}$ for the multivariate kernel, K for univariate kernels.

In our mixture model with multivariate blocks, we propose to consider two cases for the $d_\ell \times d_\ell$ diagonal bandwidth matrix of the ℓth block.

Case (i) Same bandwidth per block for all components The bandwidth matrix for block ℓ is diagonal with scalar bandwidths for each coordinates in the block: $H_\ell = \text{diagonal}\,(\boldsymbol{h}_{s_\ell}^2)$, where $\boldsymbol{h}_{s_\ell} = (h_k)_{k \in s_\ell}$. The multivariate kernel for block ℓ becomes

$$\boldsymbol{K}_{H_\ell}(\boldsymbol{u}) = \frac{1}{\prod_{k \in s_\ell} h_k} \boldsymbol{K}(H_\ell^{-1/2}.\boldsymbol{u}), \quad \boldsymbol{u} \in \mathbb{R}^{d_\ell},$$

where h_k is fixed and selected by default according to a rule of thumb from Silverman [1986], page 48:

$$h_k = 0.9 \, \min \left\{ SD_k, \frac{IQR_k}{1.34} \right\} (n)^{-1/5}, \tag{3.13}$$

and SD_k and IQR_k are respectively the standard deviation and interquartile range of the n univariate observations from the kth coordinate.

Case (ii) Adaptive bandwidth per block and component In this case the bandwidth matrix for block ℓ is diagonal with a scalar bandwidth for each coordinate in the block, but it depends also on component j and current algorithm iteration t:

$$H_{j\ell}^{(t)} = \text{diagonal}((\boldsymbol{h}_{js_\ell}^{(t)})^2), \quad \text{where } \boldsymbol{h}_{js_\ell}^{(t)} = (h_{jk}^{(t)})_{k \in s_\ell}.$$

The multivariate Kernel for block ℓ, component j and iteration t is

$$\boldsymbol{K}_{H_{j\ell}^{(t)}}(\boldsymbol{u}) = \frac{1}{\prod_{k \in s_\ell} h_{jk}^{(t)}} \boldsymbol{K}\left((H_{j\ell}^{(t)})^{-1/2}.\boldsymbol{u} \right), \quad \boldsymbol{u} \in \mathbb{R}^{d_\ell}.$$

The values of the per-block and component bandwidths are computed following the adaptive bandwidth strategy from Benaglia et al. [2011], except that in the present definition of our model there are no i.i.d. coordinates for which the n data can be pooled; as said previously, *blocks* in our model have a different meaning than in Benaglia et al. [2009a]. Each scalar bandwidth is hence determined from the corresponding n scalar observations of coordinate k, using a Silverman's like rule weighted by the posterior probabilities at each iterations of the `mvnpEM` algorithm:

$$h_{jk}^{(t+1)} = 0.9 \min \left\{ \sigma_{jk}^{(t+1)}, \frac{IQR_{jk}^{(t+1)}}{1.34} \right\} (n\lambda_j^{(t+1)})^{-1/5}, \tag{3.14}$$

where $n\lambda_j^{(t+1)}$ estimates the sample size in the jth component, and

$$\begin{aligned} \mu_{jk}^{(t+1)} &= \frac{\sum_{i=1}^n p_{ij}^{(t)} x_{ik}}{\sum_{i=1}^n p_{ij}^{(t)}} = \frac{\sum_{i=1}^n p_{ij}^{(t)} x_{ik}}{n\lambda_j^{(t+1)}} \\ \sigma_{jk}^{(t+1)} &= \left[\frac{1}{n\lambda_j^{(t+1)}} \sum_{i=1}^n p_{ij}^{(t)} (x_{ik} - \mu_{jk}^{(t+1)})^2 \right]^{1/2} \end{aligned}$$

are the weighted empirical means and variances.

To define the iterative interquartile range $IQR_{jk}^{(t+1)}$ appearing in (3.14), we introduce a weighted quantile estimate as in Benaglia et al. [2011]. Let $a_1, \ldots, a_\nu$ be real numbers and $w_1, \ldots, w_\nu$ be associated (nonnegative) weights, with $W = w_1 + \cdots + w_\nu$. Denote $\tau(\cdot)$ the permutation sorting the a_i's in non-decreasing order, $a_{\tau(1)} \leq \cdots \leq a_{\tau(\nu)}$. For $\alpha \in (0, 1)$, define *the weighted α quantile estimate* to be $\alpha_{\tau(i_\alpha)}$, where

$$i_\alpha = \min\{s : \sum_{i=1}^{s} w_{\tau(i)} \geq \alpha W\},$$

is the smallest integer that gives at least a proportion α of the total sum of weights W. We compute $IQR_{jk}^{(t+1)}$ as the difference between the estimated 0.75 and 0.25 quantiles of the $\nu = n$ observations from the kth coordinate, using weights $w_i = p_{ij}^{(t+1)}$ for the jth component. Note that function `wIQR` for computing these quantiles is provided in the mixtools package Benaglia et al. [2009b].

3.5 Implementation and simulated examples

We propose in this section some examples illustrating the performances of our algorithm, on three synthetic multivariate models, after some details about implementation and experiment settings. The `mvnpEM` algorithm defined in Section 3.4.1 has been implemented in the most recent public version of the mixtools package Benaglia et al. [2009b] for the R statistical software R Core Team [2016]. In particular, the step requiring nonparametric multivariate WKDE's has been coded in `C` to speed up approximately 9 times the CPU time.

3.5.1 Initialization of the mvnpEM algorithm.

As for EM algorithms, the choice of the starting parameter value $\boldsymbol{\theta}^{(0)}$ is important. In parametric settings, a simple manner consists in starting the algorithm from a parameter value "reasonably close" to the true value, that may be given by *a priori* knowledge obtained from some expert on the model and data. When this sort of information is not available, the usual practice consists in starting the algorithm from several values randomly drawn from a uniform distribution on the parameter space (or a subset of it), and retaining the EM estimate achieving the maximum of the observed likelihood among all the trials. If this exhaustive exploration of the parameter support is done with enough precision (enough random draws), then at least some of these randomly chosen $\boldsymbol{\theta}^{(0)}$'s fall close enough to the global maximum so that the final estimate corresponds to the global maximum.

In our nonparametric setup, we can see that the first E-step of `mvnpEM` requires initial values for the $f_j^{(0)}$'s (and $\lambda_j^{(0)}$'s) that themselves only require an initial $n \times m$ matrix of posteriors $\boldsymbol{P}^{(0)} := (p_{ij}^{(0)}, i = 1, \ldots, n, j = 1, \ldots, m)$. To obtain this matrix, the most appealing method consists in using a prior clustering of the data using any unsupervised

algorithm such as k-means, that assigns each observation to one initial component as, e.g., in Benaglia et al. [2009a]. At this point, an equivalent of the parametric initialization method based on some prior knowledge on the model and data consists in providing k-means with meaningful cluster centers instead of letting it randomly choose m centers. These "weakly informative" centers, even vaguely related to the true component means, usually help k-means finding an initial clustering good enough for an EM algorithm to start with. If such even crude prior information is not available, one can just provide k-means with the number of clusters m, so that m data points are randomly chosen as the initial centers. To be fair in our experiments, this completely blind, automatic and data-driven initialization is actually what we did in all our simulated and real data situations hereafter. Note also that this k-means based initialization is also often used in standard EM algorithms for, e.g., multivariate Gaussian mixtures, where cluster means and (co)-variances are used as initialization means and variances for the component Gaussian distributions.

In even more complex situations the above initialization strategies may fail. A first point is how to detect that k-means failed, i.e., end up with a poor clustering? The common answers from the unsupervised clustering community are (i) analyse the clusters obtained with the help of prior (external) knowledge about the clustering objective; (ii) check for the number of iterations k-means required; (iii) compare k-means solutions when started with several random centroids; (iv) compare k-means clustering against other clustering methods including a Gaussian EM (see, e.g., Sawant [2015] for precisions about (i–iii)). In addition, a k-means failure or poor initialization may also be detected in the EM framework by the algorithm itself, which often "degenerates", producing an estimate with $m-1$ components after few iterations i.e., one of the λ_j estimates goes to zero (see Section 3.5.4 for an example when this can occur). Hence, when the above initialization strategies fail we can proceed by analogy with the parametric space exploration: draw $\boldsymbol{P}^{(0)}$ posterior matrices randomly (uniformly) several times, and run several `mvnpEM` algorithms initialized with these $\boldsymbol{P}^{(0)}$'s. Then retain the $\hat{\boldsymbol{\theta}}$ corresponding to the largest "observed loglikelihood" $\sum_{i=1}^{n} \log g_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_i)$ which is not in the nonparametric case a true likelihood but merely an empirical criterion. The uniform simulation of $\boldsymbol{P}^{(0)}$ can be done in several ways, e.g. simply by choosing, for each row the j for which $p_{ij}^{(0)} = 1$ uniformly in $\{1, \ldots, m\}$. One can also use uniform Dirichlet if non 0/1 weights are desired. There is also always the possibility to run a first parametric Gaussian EM to get a first matrix of posteriors to start `mvnpEM`. We tried the initialization strategy using uniform Dirichlet for Models A and B defined below, and obtained the same results as with the k-means initialization.

Handling the label-switching problem Not surprisingly, the data-driven initialization without specifying initial centers for the k-means procedure generates more label-switching than when proper centers are provided. As explained in Section 2.1.1, label-switching refers to the fact that arbitrary re-orderings of the component indices $(1, \ldots, m)$ correspond to the same mixture model. In a single real data study, label switching is not important since a component index does not change its interpretation. But these re-orderings are possible when numerous instances of the same mixture problem are solved. Hence label-switching becomes problematic in Monte-Carlo simulation studies and bootstrap estimation involving mixture models. For detailed explanation, see discussion in

McLachlan and Peel [2000] (section 4.9), and for an illustrative stochastic EM example see Celeux et al. [1996]. In their study, Hall et al. [2005] dealt with label-switching in the same context by enforcing the constraint $\hat{\lambda}_1 < \hat{\lambda}_2$. We choose here to detect and "switch-back" the estimates (the final matrix of posteriors from which the other estimates are computed) to be in accordance with the initial representation. Since in all our experiments we ordered the weights $\lambda_1 < \cdots < \lambda_m$, we decided that a switching occured after a replication if this order was not preserved for the estimates.

In our Monte-Carlo experiments, we computed the errors in terms of the square root of the Mean Integrated Squared Error (MISE) for the densities as in Hall et al. [2005] and Benaglia et al. [2009a]:

$$MISE_{j\ell} = \frac{1}{S}\sum_{s=1}^{S}\int(\hat{f}_{j\ell}^{(s)}(\boldsymbol{u}) - f_{j\ell}(\boldsymbol{u}))^2 d\boldsymbol{u},$$

where the integral over $\mathbb{R}^{d_\ell}$ is computed numerically and $\hat{f}_{j\ell}^{(s)}$ is the density estimate at replication s, computed from (3) but using the final values of the $p_{ij}^{(t)}$'s, i.e. the posterior probabilities after convergence of the algorithm that we denote $\hat{p}_{ij}$'s.

In our experiments, when we applied `mvnpEM` for Model A (see the detail in Table 3.2) which has two univariate blocks and one bivariate block, we found a problem with numerical integral of a probability density function with strong correlation on a hypercube. Fig 3.3 gives an example for showing that the $MISE_{j\ell}$ for block 3 only (the only multivariate block) depends on the correlation ρ.

A difference with both Hall et al. [2005] and Benaglia et al. [2009a] results is that in their work the Integrated Squared Errors $ISE_{j\ell} = \int(\hat{f}_{j\ell} - f_{j\ell})^2$ were evaluated using numerical integrations of univariate densities (since the $f_{j\ell}$'s were univariate only). Here, it appears that estimating $f_{j\ell}$ for multivariate densities with very strong dependence structure using a kernel density estimate (KDE) with diagonal bandwidth matrix is more difficult, and this difficulty may result in overestimated MISE values, not necessarily implying a poor fitting of the mixture by the algorithm. To illustrate that in a simple case, we ran $S = 300$ replications of $n = 300$ observations of a single bivariate sample (i.e. no mixture, no posteriors, usage of standard unweighted KDE) from a centered bivariate Gaussian density f with unit variances and varying correlation ρ. We then computed $MISE_f = \frac{1}{S}\sum_{s=1}^{S}\int(\hat{f}^{(s)} - f)^2$ using a bandwidth matrix following Silverman [1986] as in (3.13). Results are in Table 3.1:

ρ	0.25	0.5	0.8	0.95	0.99
$MISE_f$	0.00339	0.00349	0.00601	0.03547	0.25591

Table 3.1 – The effect of correlation ρ on MISE of the estimation of a centered bivariate Gaussian density f with unit variances.

This shows that estimation of the MISE deteriorates as correlation increases. Using a non-diagonal bandwidth matrix is thus an interesting perspective for future work, to

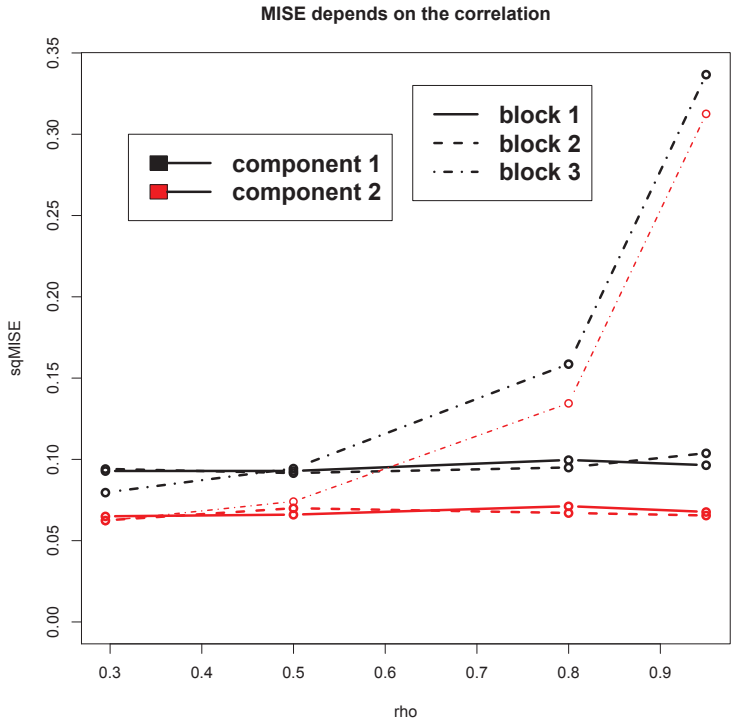


Figure 3.3 – Square roots of MISE for the densities $f_{j\ell}$ of 3 blocks $\ell = 1, 2, 3$, for Model A, $n = 500$ and $S = 300$ replications, adaptive bandwidth. The two colors correspond to the components.

better recover multivariate and strongly correlated component and block densities. In our present setup and experiment, in order to get results not too biased by this KDE problem i.e. to obtain comparable $MISE_{j\ell}$'s between univariate and multivariate blocks, we selected variance matrices Σ_j's with correlations not larger than 75%.

We also computed the mean squared errors (MSE's) for the proportions that are the only scalar parameters in these models. For a weight λ_j,

$$MSE_{\lambda_j} = \frac{1}{S}\sum_{s=1}^{S}(\hat{\lambda}_j^{(s)} - \lambda_j)^2,$$

where, at replication s, $\hat{\lambda}_j^{(s)}$ is computed using (2) with the final values of the posterior probabilities, $\hat{p}_{ij}$'s. Note that we computed and displayed as well MSE's for other scalar empirical moments like means and variances, but these are not genuine parameters of the model, i.e. they are provided only as additional criteria. At each replication, these scalar measures are weighted versions of the empirical estimates; for instance, the mean for component j and coordinate k is given by

$$\hat{\mu}_{jk} = \frac{\sum_{i=1}^{n}\hat{p}_{ij}\,x_{ik}}{\sum_{i=1}^{n}\hat{p}_{ij}} = \frac{\sum_{i=1}^{n}\hat{p}_{ij}\,x_{ik}}{n\hat{\lambda}_j}.$$

Finally, we compared several models in terms of their clustering efficiency. Model-based clustering using mixture models is done using the *Maximum A Posteriori* (MAP) strategy deduced from the parameter estimate $\hat{\boldsymbol{\theta}}$ given by any EM-like algorithm. The MAP consists in setting

$$\hat{Z}_{ij_0} = 1, \quad \text{where } j_0 = \underset{j=1,\ldots,m}{\arg\max}\left\{\hat{p}_{ij}\right\}, \quad \text{and } \hat{Z}_{ij} = 0 \text{ for } j \neq j_0,$$

where the $\hat{p}_{ij}$'s are as above the posterior probabilities after convergence of the algorithm.

3.5.2 Model A: simple Gaussian data

We first introduce this simple model with two univariate blocks and one bivariate block, chosen intentionally as close as possible to model (2.5) (with conditionally independent univariate marginals) used first by Hall et al. [2005] to illustrate the performance of their estimation technique based on inverting the mixture. Their example was considering $r = 3$ conditionally independent univariate Gaussian, all $\mathcal{N}(0,1)$ for component 1, and $\mathcal{N}(3,1)$, $\mathcal{N}(4,1)$ and $\mathcal{N}(5,1)$ for component 2. This model has being used later in Benaglia et al. [2009a] for comparison with the `npEM` algorithm. We consider a $r = 4$ variables, $m = 2$ components Gaussian mixture which have one multivariate block, i.e. $B = 3$ blocks of coordinates with $s_1 = \{1\}, s_2 = \{2\}, s_3 = \{3,4\}$. Densities $f_{j\ell}$ are univariate normals for $l = 1, 2$, and bivariate Gaussian for block $l = 3$, where the means and the common covariance matrix of the bivariate block are given in Table 3.2: Hence to allow comparison with the original `npEM` and both Hall et al. [2005] and Benaglia et al. [2009a] results for the univariate coordinates, we kept individual densities as in their examples for the first and the second block. We also kept their experiment settings: $S = 300$ replications of $n = 500$ observations each, where λ_1 is varying from 0.1 to 0.4.

Model A	Block 1	Block 2	Block 3
Coordinate(s)	1	2	$\{3,4\}$
Component 1	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 1/2\\1/2 & 1\end{bmatrix}\right)$
Component 2	$\mathcal{N}(3,1)$	$\mathcal{N}(4,1)$	$\mathcal{N}_2\left(\begin{bmatrix}3\\3\end{bmatrix}, \begin{bmatrix}1 & 1/2\\1/2 & 1\end{bmatrix}\right)$

Table 3.2 – Parameters for Model A.

Results for model A ran with the adaptive bandwidth strategy are given in Fig. 3.4. We obtained similar results with the simplest bandwidth setting. For this model with similar ranges across components and blocks, the bandwidth strategy does not make a noticeable difference. These results were obtained, as said in Section 3.5.1, using k-means initialization with randomly chosen initial centers and checking for label switching.

The stable behavior of the MSE's for λ_1 and for the other scalar measures (means, covariances) estimates show that the algorithm behaves well. In particular, density and scalar estimate errors associated with component 1 decrease when λ_1 increases, as expected since the proportion of data actually coming from this component increases with λ_1. Simultaneously, the estimate errors associated with component 2 increase. Moreover, the results for the $\sqrt{MISE}_{f_{j\ell}}$'s are close to the results we can see on the plots on page 517, figure 2 of Benaglia et al. [2009a] and outperform the plots on page 675, figure 2 of Hall et al. [2005] for univariate blocks.

3.5.3 Model B: Three-component Gaussian heavy tailed and skewed data

We also experimented our method on a second model, with $m = 3$ components and three bivariate blocks using the full potential of our approach. We wanted here to show that our algorithm can compete to some extent with a fit based on a Gaussian mixture model where mixture components are indeed Gaussian, and do better when they are non Gaussian, all this using a single model for brevity. Model B thus has one bivariate Gaussian block, one bivariate block with heavy-tailed (Student) distributions, and one bivariate block with heavy-tailed and severely skewed distributions. Precisely, it has $r = 6$ variables, $m = 3$ components, and $B = 3$ blocks. Block 1 involves bivariate Gaussian densities $\mathcal{N}_2(\mu_{j1}, \Sigma)$'s with some correlation structure; block 2 involves bivariate non-central Student densities with the same correlation structure and four degrees of freedom. The component densities of block 3 are themselves mixtures of bivariate Gaussian contaminated by bivariate Student's, thus generating skewed densities. Note that it is common to use parametric mixtures as a simulation tool to build synthetic complex models like skewed or contaminated distributions, and this is what we did here for the third block. Nevertheless, since the other block densities are not themselves mixtures, model B is a genuine three-component mixture. Of course, one could re-write and interpret it as a 6-component mixture, but with several non-natural equality constraints between blocks 1

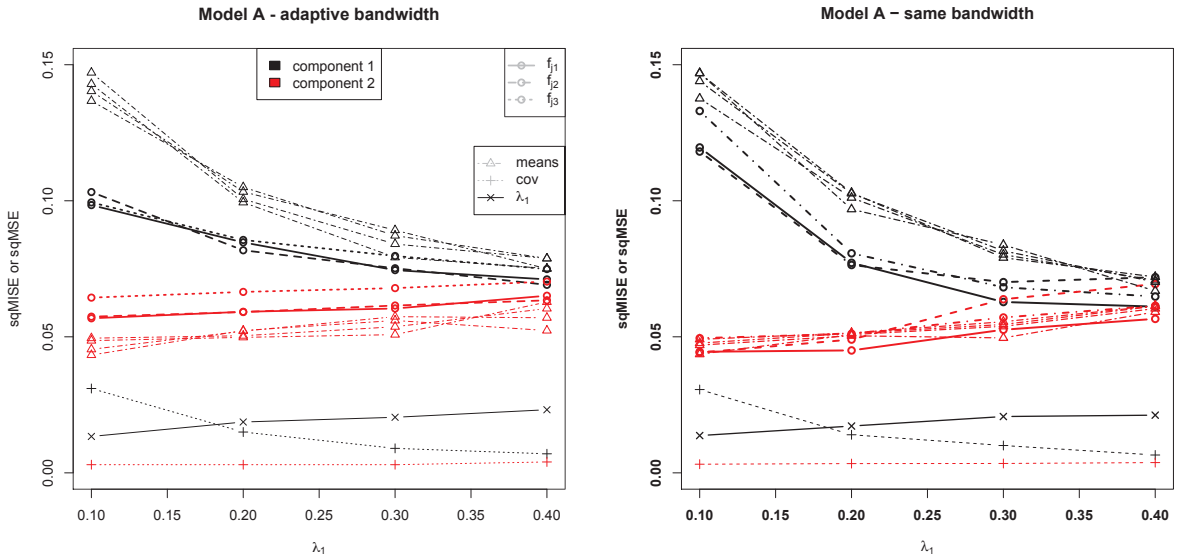


Figure 3.4 – Square roots of MISE for the densities and square roots of MSE for the scalar parameter λ_1, and means and covariances (that are not parameters in the model), for several values of λ_1, for Model A, $n = 500$ and $S = 300$ replications, adaptive bandwidth and same bandwidth. The gray line types in the legend are identifying densities and scalar criterions, that are plotted colored by component.

and 2 densities, which is not the model fitted here.

The component proportions for this model are set to $(15\%, 35\%, 50\%)$, and it involves two covariance matrices,

$$\Sigma = \begin{pmatrix} 1 & 3/4 \\ 3/4 & 1 \end{pmatrix} \quad \text{and} \quad \Sigma' = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 4 \end{pmatrix},$$

where Σ' is used only in block 3, component 2 and 3. The other parameters are given in Table 3.3, where two settings for the means are displayed: the first setting (model B) corresponds to a complex model because most of the within-block components are severely overlapping, it is the model for which we provide detailed results; the second setting defines more separated components for some coordinates, resulting in an easier model denoted B2, which purpose is to allow interesting comparisons between a parametric Gaussian EM and our non-parametric algorithm.

Before presenting a full Monte-Carlo experiment as for model A, we display in Figure 3.5 the true marginal densities of model B, together with results from a single run of the `mvnpEM` algorithm and a standard Gaussian EM algorithm using the `mvnormalmixEM` function from the mixtools package Benaglia et al. [2009b].

Figure 3.5 shows that the `mvnpEM` fit is rather good in all component and block marginal densities (including the recovering of the contaminated densities in block 3), whereas the Gaussian EM cannot recover the shape of the f_{jl} marginals (except for block 1, component 1). Surprisingly, the Gaussian EM fails even for the Gaussian block 1, this being probably due to these severely overlapping and non Gaussian components and blocks. The MAP clustering from the `mvnpEM` solution gave a clustering error of 4.2%,

Model B {B2}	Block 1	Block 2	Block 3
Coords	$\{1,2\}$	$\{3,4\}$	$\{5,6\}$
$j=1$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}\begin{Bmatrix}-3\\0\end{Bmatrix}\right)$	$t_2\left(\begin{bmatrix}0\\0\end{bmatrix}\begin{Bmatrix}-3\\0\end{Bmatrix}\right)$	$\alpha t_2\left(\begin{bmatrix}0\\0\end{bmatrix}\begin{Bmatrix}-3\\0\end{Bmatrix}\right)+\beta\mathcal{N}_2\left(\begin{bmatrix}4\\6\end{bmatrix}\right)$
$j=2$	$\mathcal{N}_2\left(\begin{bmatrix}1\\5\end{bmatrix}\begin{Bmatrix}1\\4\end{Bmatrix}\right)$	$t_2\left(\begin{bmatrix}2\\5\end{bmatrix}\begin{Bmatrix}1\\5\end{Bmatrix}\right)$	$\alpha t_2\left(\begin{bmatrix}2\\8\end{bmatrix}\begin{Bmatrix}1\\8\end{Bmatrix}\right)+\beta\mathcal{N}_2\left(\begin{bmatrix}5\\14\end{bmatrix},\Sigma'\right)$
$j=3$	$\mathcal{N}_2\left(\begin{bmatrix}3\\7\end{bmatrix}\begin{Bmatrix}5\\8\end{Bmatrix}\right)$	$t_2\left(\begin{bmatrix}3\\7\end{bmatrix}\begin{Bmatrix}5\\7\end{Bmatrix}\right)$	$\alpha t_2\left(\begin{bmatrix}3\\10\end{bmatrix}\begin{Bmatrix}5\\10\end{Bmatrix}\right)+\beta\mathcal{N}_2\left(\begin{bmatrix}7\\15\end{bmatrix},\Sigma'\right)$

Table 3.3 – Parameters for Model B, together with the alternative mean vectors for the easier model B2 displayed in braces when appropriate. The covariance matrices used in Gaussian and Student distributions are Σ except when Σ' is specified. All the multivariate Student distributions involve 4 degrees of freedom. The weights for the mixture within block 3 are $\alpha = 0.87$ and $\beta = 1 - \alpha = 0.13$.

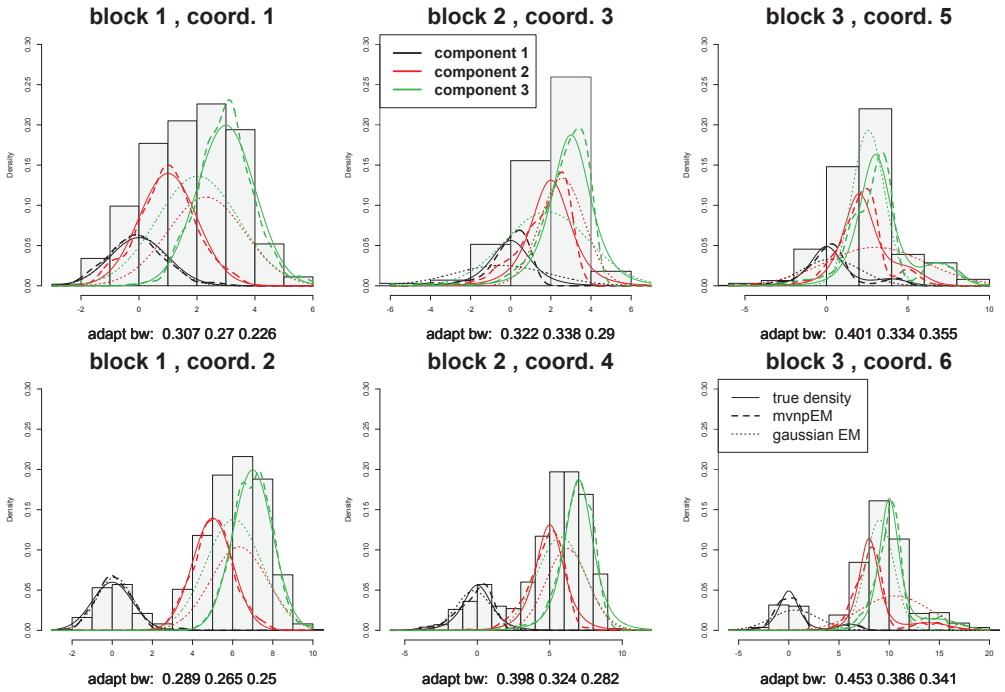


Figure 3.5 – Marginal density estimates for a sample of size $n = 1000$ from Model B, where column l corresponds to the two marginals of the lth bivariate block, $l = 1, 2, 3$. Each plot shows the true marginals (solid lines), the `mvnpEM` with adaptive bandwidth estimates (dashed lines), Gaussian EM estimates (dotted lines). The final values of the adaptive bandwidths are also given under each plot.

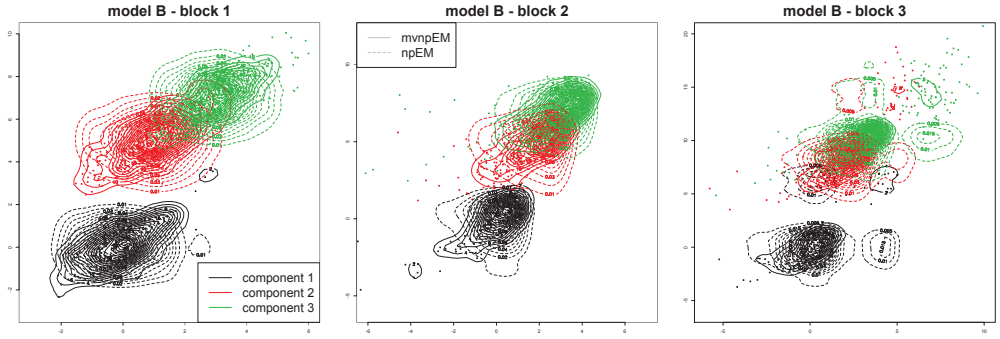


Figure 3.6 – Level plots of the bivariate mixture densities per block, estimated by the `mvnpEM` (solid lines) and Benaglia et al. [2009a] (dashed lines), for Model B. Scatterplots are colored by their true cluster membership.

whereas the Gaussian EM returns 45%, with confusion mostly between components 2 and 3, as the marginals suggest. A run of the `npEM` algorithm from Benaglia et al. [2009a] (which fits $r = 6$ univariate blocks) returns a clustering error of 5.4%. Since plots of the marginal densities do not show the estimation of the dependence structure, we then propose to illustrate the difference betwen these two alternative nonparametric solutions by plotting the estimated bivariate densities: the $\hat{f}_{jl}$'s estimated from the `mvnpEM` versus the product of the univariate kernel density estimates `npEM` given by Benaglia et al. [2009a]. Figure 3.6 shows that the `mvnpEM` captures the dependence structure (here the 75% within-block correlation and block 3 contamination). This illustrates the essential difference with the univariate conditional independence assumption of Benaglia et al. [2009a], for which the joint density can only be obtained by the product of two univariate marginals, resulting in wrong joint densities (see, e.g., block 1 component 1).

We then ran $S = 300$ replications of samples of sizes $n = 400, 600, 800, 1000$. The fact that the Gaussian EM did not recover even the Gaussian block 1 for model B motivates a second experiment with our model B2, for which we could more easily compare our method against a Gaussian solution. Moreover, since our purpose was also to build a model illustrating the performance of the adaptive bandwidth strategy (Section 3.4.2), which is appropriate typically for models with different ranges of observations per components and coordinates, we ran this experiment for both bandwidth strategies. We computed the $MISE$'s of the densities and the MSE's of the scalar parameters $\boldsymbol{\lambda}$ for all these cases. Figure 3.7 (left and middle) first shows that the MISE's decrease when the sample size n increases, which can be understood as numerical evidence of "consistency". These results also show a slight advantage for the adaptive bandwidth strategy, as expected. Finally we can see by comparing in Figure 3.7 the nonparametric and parametric solutions block per block, that for the well-separated mixture (B2), our method is slightly outperformed by the Gaussian EM for the Gaussian block 1, but gives better results than it for the heavy-tailed block 2, and this is even better for the heavy-tailed and skewed block 3 (in blocks 2 and 3, the parametric estimates even show no convergence at all as n increases). Not surprisingly, for the more overlapping model B, our algorithms entirely gave the best estimation.

We finally compared the performance of the two nonparametric algorithms in term

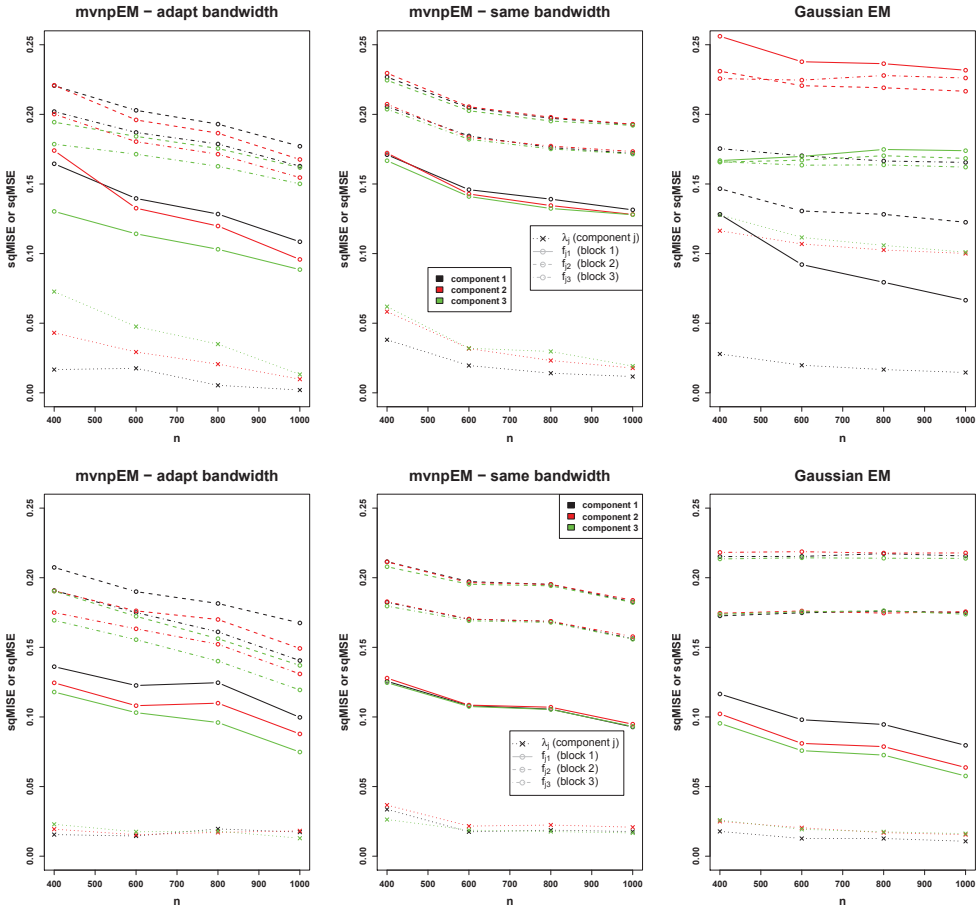


Figure 3.7 – Square roots of MISE's for the densities as a function of the sample size n, $S = 300$ replications, for the two algorithm settings for Model B (top) and for the less overlapping Model B2 (bottom): `mvnpEM` adaptive bandwidth (left), `mvnpEM` same bandwidth (middle) and Gaussian EM (right). MISE's for densities are plotted in circles and solid lines (block 1), dashed lines (block 2) and dot-dashed (block 3). MSE's for the proportion estimates are given in dotted lines.

Method	(%) correctly-classified
k-means	83.11
`npEM` & MAP	73.27
`mvnpEM` & MAP	92.87

Table 3.4 – The % of correct clustering averaged over $S = 300$ replications, for model B using `mvnpEM` and the MAP strategy, compared with the k-means algorithm and the method given by Benaglia et al. [2009a].

Method	(%) correctly-classified
k-means	85.07
`npEM` & MAP	45.15
`mvnpEM` & MAP	99.43

Table 3.5 – The % of correct clustering averaged over $S = 300$ replications of size $n = 2000$ from model C.

of their MAP clustering performance. Table 3.4 clearly shows that the `mvnpEM` for our model with within-block dependence structure outperforms the algorithm proposed by Benaglia et al. [2009a], and is also better than the classical k-means algorithm (which is not model-based).

3.5.4 Model C: non-linear dependence within clusters

In this section we briefly show how our method behaves for a model involving non-linear dependencies within clusters (components). We choose as a building density the non-linear "banana-shaped" distribution that has been proposed by Haario et al. [2001] and used by several authors since then, mostly in Monte-Carlo Markov Chain literature. It is constructed by "twisting" a d-multivariate Gaussian distribution $\mathcal{N}_d(\mathbf{0}, C)$ with diagonal covariance matrix $C = \text{diag}(100, a, \ldots, a)$ and density denoted $\mathcal{N}_d$ as well. The banana-shaped density is $f_b(\boldsymbol{x}) = \mathcal{N}_d \circ \phi_b(\boldsymbol{x})$, where $\phi_b(x_1, \ldots, x_d) = (x_1, x_2 + bx_1^2 - 100b, x_3, \ldots, x_d)$. The so-called "bananicity" constant b controls the non linearity of the distribution. Haario uses values $a = 1$, $b = 0.03$ for a moderately twisted density, and $b = 0.1$ for a strongly twisted density. We designed here a 2-component mixture with $B = 3$ bivariate blocks ($r = 6$) by shifting simulated data from f_b in a way to obtain overlapping and non-linear clusters of different "bananicity" constants in each blocks. Figure 3.8 shows the contour plots of the two-dimensional density estimates for a typical run from a sample of size $n = 2000$. It is clear that the `mvnpEM` model and algorithm captures the non-linearity of these clusters in all blocks, whereas the univariate block strategy fails. The typical MAP clustering for this model is given in Table 3.5 which compares k-means and the two nonparametric algorithms.

Note that for this model with non convex clusters, the initialization based on the k-means algorithm is expected to behave poorly. Indeed, we did observe few cases where the

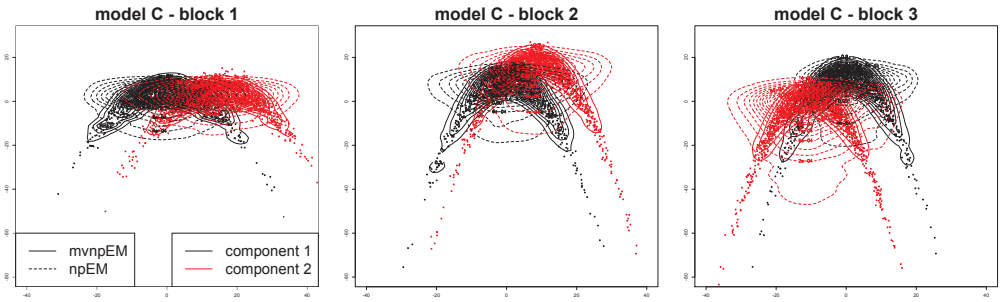


Figure 3.8 – Level plots of the bivariate mixture densities per block, estimated by the `mvnpEM` (solid lines) and Benaglia et al. [2009a] (dashed lines), for Model C, $n = 2000$. Scatterplots are colored by their true cluster membership.

`mvnpEM` algorithm degenerates (by emptying one component). These cases where solved as indicated in Section 3.5.1 by applying random initialization.

3.6 A real data example

We consider here a real dataset, Wisconsin Diagnostic Breast Cancer (WDBC), from an experiment involving $n = 569$ instances (see description in Section 3.1).

This actual dataset has already been used as an illustration for comparing supervised and unsupervised clustering methods. The principle of such a study from the unsupervised clustering perspective consists in clustering the population based on the quantitative variables, and after that comparing these estimates given the observed classes.

Our motivation in using this dataset is not to find a scientific definitive answer or the best clustering algorithm. We have chosen this dataset because: (i) it illustrates the potential and feasibility of our estimation algorithm for models involving blocks and data of moderate to large dimensions; (ii) there are obvious dependence structures across some coordinates that prevent the usage of the previous nonparametric `npEM` approach from Benaglia et al. [2009a] since the conditional independence of coordinates is obviously violated (see Fig. 3.9); (iii) it has been used recently in Hennig [2010], who proposed a competitive, alternative model-based parametric but not simply Gaussian clustering: their method amounts to build clusters by merging components obtained from a Gaussian mixture model fit. Hence their cluster distributions are not Gaussian, they can e.g., be multimodal.

In their merging Gaussian method, Hennig [2010] just used the ten "mean" features of the WDBC dataset. Hence we first tried our approach on this $r = 10$ dimensional dataset. As we discussed in Section 3.1 and showed the dependence among this ten features in Fig 3.2, we had to define multivariate conditionally independent blocks prior to apply our `mvnpEM` algorithm. Fig. 3.9 displays the most obvious such dependencies among the ten mean features. It is for instance clear that radius, perimeter and area must be grouped in one block. Similarly, compactness, concavity and number of concave points can be grouped in another block.

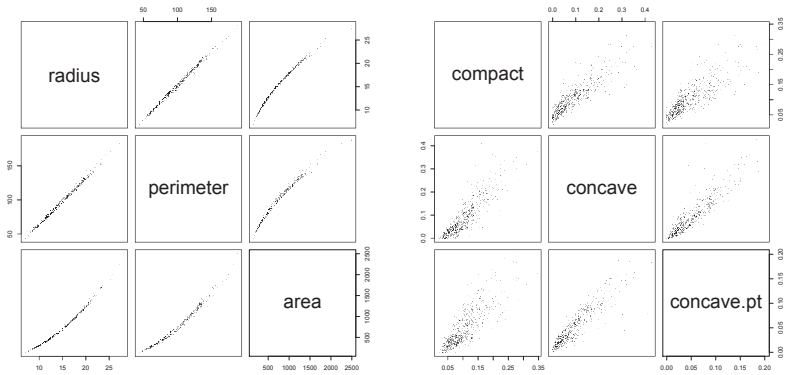


Figure 3.9 – Pair plots for selected "mean" features from the WDBC database; $s_1 = \{1, 3, 4\}$ for block 1 (left), and $s_2 = \{6, 7, 8\}$ for block 2 (right).

Method	B (over 357)	M (over 212)	(%) correctly-classified
k-means	355	122	83.831
merging Gaussian	344	178	91.740
`mvnpEM` & MAP	350	183	93.673

Table 3.6 – The % of correct classification of the WDBC data using `mvnpEM` and the MAP strategy, compared with the k-means clustering strategy and the merging Gaussian method of Hennig [2010].

This way of designing blocks can be summarized by the following general guidelines:

1. group in a block coordinates obviously dependent "by definition" whatever their component membership, using prior or expert information from the model;
2. look for obvious dependence structures between groups of coordinates not due to clustering (when clusters are visible on some scatterplots); this can be done using pair plots as in Fig. 3.9, and also using the correlation matrix.
3. Try several reasonable block structures for the remaining coordinates for which none of the above rules could lead to a clear block design. Compare MAP clustering results between the possible block structures, in view of what is expected from the clustering (analysis of the clusters with the help of prior or external knowledge about the clustering objective).

Proceeding like this, we are able to design some plausible models. One of the best ones for clustering precision uses $B = 5$ blocks: the two trivariate blocks from Fig. 3.9, a block of size 2 (symmetry and fractal dimension), and two remaining blocks of size 1. The results are given in Table 3.6, together with k-means that we tried as well (and that is used in our initialization of the `mvnpEM`, see Section 3.5.1). The MAP classifier is compared with the classes given by the Diagnosis (62.74% B and 37.26% M).

In experimenting some alternative block designs we somehow proceed like Hennig [2010] who tries several (heuristical) merging criterions and reports the best result ob-

tained. However, in our case, these alternative models (e.g., merging smoothness with the block in Fig. 3.9, right) always showed results between 92.5 and 93.67% i.e. better than Hennig [2010]. We tried more complex models by adding the other groups of available measures, the 10 standard errors (se), or the 10 "worst" measures keeping the same structure in $B = 5$ blocks (also supported by the exploration of the scatterplots). We found that adding the se's where not bringing better results, whereas a $r = 20$ dimensional model made of the means and worst features with $B = 5$ blocks as before but of double dimensions (e.g., $s_1 = 6$ and block 1 made of radius, perimeter and area features) gave a slightly better 94% of correct classification. Finally, we tried the full $r = 30$ model with $B = 5$ blocks of sizes up to $s_1 = s_2 = 9$ with no better results. However, this showed us that running the algorithm on these large dimensional models and $n = 569$ individuals only took a few minutes (1.45ms) on a common laptop computer.

Chapter 4

Maximum Smoothed Likelihood for Nonparametric mixture with multivariate blocks

4.1 Introduction

The convergence of our `mvnpEM` algorithm presented in Chapter 3, is not proved. The reason is that, as for the original `npEM` algorithm for univariate blocks from Benaglia et al. [2009a], the `mvnpEM` is not proved to maximize any objective function, and its weighted KDE step is not a genuine M-step. Like its predecessors in recent literature, it however provides "numerical evidence of consistency" in the sense that MSE an MISE measures decrease when we let n increase, for all the models we tried. The purpose of this chapter is precisely to show some type of convergence, extending the ideas from Levine et al. [2011] and Chauveau et al. [2015] introducing a non-linearly smoothed log-likelihood objective function and developing an iterative algorithm with a monotony property as for a genuine EM.

We introduce in Section 4.2 a smoothed model for the finite mixture model of completely unspecified multivariate components under the assumption of conditionally independent blocks of coordinates which was presented in Section 3.2 of Chapter 3. Then we define an iterative algorithm for the `mvnpEM` algorithm in Section 4.3. We prove that this algorithm, based on a majorization-minimization idea, possesses a desirable descent property just as any EM algorithm does. Section 4.4 devotes to estimating the parameters of this smoothed model. We also present some convergence properties in the end of this Section. Some simulation studies show that the new algorithm gives very similar results to `mvnpEM` algorithm that does not satisfy the descent property, are described in Section 4.5.

4.2 The smoothed model

We first recall here the nonparametric mixture model with multivariate blocks and multivariate kernel density function.

Suppose the vectors $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n$ are a simple random sample of $m > 1$ components, the weight of jth component is λ_j. Assume that each jth component density f_j is equal to the product of B conditionally independent multivariate *blocks* of densities, i.e. $f_j(\boldsymbol{x}_i) = \prod_{\ell=1}^{B} f_{j\ell}(x_{is_\ell})$, where $\boldsymbol{x}_i \in \mathbb{R}^r$ and $x_{is_\ell} = \{x_{ik}, k \in s_\ell\} \in \mathbb{R}^{d_\ell}$ with s_ℓ's are disjoint subsets satisfy $\bigcup_{\ell=1}^{B} s_\ell = \{1, \cdots, r\}$ and $d_\ell = \text{ card}(s_\ell)$ is the number of coordinates in ℓth block. Then the density of each $\boldsymbol{X}_i$ is written

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{\ell=1}^{B} f_{j\ell}(x_{is_\ell}),$$

where $\boldsymbol{\theta} = (\boldsymbol{\lambda}, \boldsymbol{f}) = (\lambda_1, \cdots, \lambda_m, f_1, \cdots, f_m)$ are the parameters of the model.

In the E-step of `mvnpEM` algorithm, Chapter 3 and Chauveau and Hoang [2016] used a multivariate kernel density function to estimate nonparametric densities. This multivariate kernel density function is considered in the constrained context with diagonal bandwidth matrices (see more detail in Section 2.3.4). It also works for any block-diagonal bandwidth matrices.

For $\boldsymbol{u} = (u_1, u_2, ..., u_r)^T \in \mathbb{R}^r$, remind here in the simple case of the bandwidth matrix $H = \text{diag}(h_1^2, h_2^2, ..., h_r^2)$, where h_k denotes the kth coordinate bandwidth, the multivariate kernel is multiplicative in the sense that:

$$\boldsymbol{K}_H(\boldsymbol{u}) = \frac{1}{h_1 ... h_r} \boldsymbol{K}\left(\frac{u_1}{h_1}, ..., \frac{u_r}{h_r}\right) = \prod_{k=1}^{r} \frac{1}{h_k} K\left(\frac{u_k}{h_k}\right) = \prod_{\ell=1}^{B} \boldsymbol{K}_{H_\ell}(\underline{u}_{s_\ell}),$$

where $\underline{u}_{s_\ell} = \{u_k, k \in s_\ell\} \in \mathbb{R}^{d_\ell}$. It itself can be a d_ℓ-product. $H_\ell = \text{diag}(h_k^2)_{k \in s_\ell}$ is $d_\ell \times d_\ell$ diagonal symmetric positive bandwidth matrix of ℓth block.

Smoothed model We now assume that Ω is a compact subset of $\mathbb{R}^r$ and define the linear vector function space

$$\mathcal{F} = \{\boldsymbol{f} = (f_1, ..., f_m)^\top : 0 < f_j \in L_1(\Omega),\ \log f_j \in L_1(\Omega),\ j = 1, ..., m\}.$$

In the theoretical side, this assumption appears as a limitation in most cases. But this limit does neither appear in the practical computation implementation nor play an important role in our algorithm.

Then we define the smooth operators for any function $f_j \in L_1(\Omega)$ and any d_ℓ- multivariate function $f_{j\ell} \in L_1(\Omega^\ell)$, $\quad \Omega^\ell \subsetneq \mathbb{R}^{d_\ell}$ as

$$\mathcal{S}_H f_j(\boldsymbol{x}) = \int_\Omega \boldsymbol{K}_H(\boldsymbol{x} - \boldsymbol{u}) f_j(\boldsymbol{u}) d\boldsymbol{u},$$

$$\mathcal{S}_{H_\ell} f_{j\ell}(\underline{x}_{s_\ell}) = \int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{s_\ell} - \underline{u}_{s_\ell}) f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}.$$

Their corresponding nonlinear operators are

$$\mathcal{N}_H f_j(\boldsymbol{x}) = \exp\{(\mathcal{S}_H \log f_j)(\boldsymbol{x})\} = \exp \int_\Omega \boldsymbol{K}_H(\boldsymbol{x}-\boldsymbol{u}) \log f_j(\boldsymbol{u}) d\boldsymbol{u},$$

$$\mathcal{N}_{H_\ell} f_{j\ell}(\underline{x}_{s_\ell}) = \exp\{(\mathcal{S}_H \log f_{j\ell})(\underline{x}_{s_\ell})\} = \exp \int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{s_\ell}-\underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}.$$

These smooth and nonlinear operators depend on the multivariate kernel density function which itself depends on the bandwidth matrix H or H_ℓ per block. However to be brief, from here we use the simple notations $\mathcal{S}f_j$, $\mathcal{S}f_{j\ell}$, $\mathcal{N}f_j$ and $\mathcal{N}f_{j\ell}$ instead of $\mathcal{S}_H f_j$, $\mathcal{S}_{H_\ell} f_{j\ell}$, $\mathcal{N}_H f_j$ and $\mathcal{N}_{H_\ell} f_{j\ell}$, respectively.

From $f_j(\boldsymbol{x}) = \prod_{\ell=1}^B f_{j\ell}(\underline{x}_{s_\ell})$ and the Fubini's theorem we have that the operator $\mathcal{N}$ is multiplicative in sense $\mathcal{N}f_j(\boldsymbol{x}) = \prod_{\ell=1}^B \mathcal{N}f_{j\ell}(\underline{x}_{s_\ell})$. Indeed,

$$\begin{aligned}
& \mathcal{N}f_j(\boldsymbol{x}) \\
= \;& \exp \int_\Omega \boldsymbol{K}_H(\boldsymbol{x}-\boldsymbol{u}) \log f_j(\boldsymbol{u}) d\boldsymbol{u} \\
= \;& \exp \int_\Omega \boldsymbol{K}_H(\boldsymbol{x}-\boldsymbol{u}) \log \left[\prod_{\ell=1}^B f_{j\ell}(\underline{u}_{s_\ell}) \right] d\boldsymbol{u} \\
= \;& \exp \int_\Omega \boldsymbol{K}_H(\boldsymbol{x}-\boldsymbol{u}) \left[\sum_{\ell=1}^B \log f_{j\ell}(\underline{u}_{s_\ell}) \right] d\boldsymbol{u} \\
= \;& \exp \sum_{\ell=1}^B \left[\int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{s_\ell}-\underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} \prod_{\ell' \in \{1,\cdots,B\}, \ell' \neq \ell} \int_{\Omega^{\ell'}} \boldsymbol{K}_{H_{\ell'}}(\underline{x}_{s_{\ell'}}-\underline{u}_{s_{\ell'}}) d\underline{u}_{s_{\ell'}} \right] \\
= \;& \prod_{\ell=1}^B \left[\exp \int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{s_\ell}-\underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} \right] = \prod_{\ell=1}^B \mathcal{N}f_{j\ell}(\underline{x}_{s_\ell}) \quad \square
\end{aligned}$$

Similarly, $\mathcal{S}f_j(\boldsymbol{x})$ itself has a ℓ-product form and owns the multiplicative property.

To simplify notation, we introduce the finite mixture operator

$$\mathcal{M}_{\boldsymbol{\lambda}} \boldsymbol{f}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{j=1}^m \lambda_j f_j(\boldsymbol{x}) = \sum_{j=1}^m \lambda_j \prod_{\ell=1}^B f_{j\ell}(x_{is_\ell}),$$

so that $\mathcal{M}_{\boldsymbol{\lambda}} \boldsymbol{f}(\boldsymbol{x}) = g_{\boldsymbol{\theta}}(\boldsymbol{x})$, and

$$\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{j=1}^m \lambda_j \mathcal{N}f_j(\boldsymbol{x}) = \sum_{j=1}^m \lambda_j \prod_{\ell=1}^B \mathcal{N}f_{j\ell}(\underline{x}_{s_\ell}). \tag{4.1}$$

Hence $\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}(.)$ is considered as a *smoothed finite mixture model*.

4.3 A MM algorithm

We follow here the technique introduced in Levine et al. [2011]. Let $g(\boldsymbol{x})$ now represent a known target density function. We consider the following function of $\boldsymbol{\theta}$, where $\boldsymbol{\theta} = (\boldsymbol{f}, \boldsymbol{\lambda})$

are the parameters of model (3.1)

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda}) = \int_\Omega g(\boldsymbol{x}) \log \frac{g(\boldsymbol{x})}{[\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}](\boldsymbol{x})} d\boldsymbol{x} \tag{4.2}$$

$$= D(g|\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}) + \int_\Omega g(\boldsymbol{x}) d\boldsymbol{x} - \sum_{j=1}^m \lambda_j \int_\Omega \mathcal{N} f_j(\boldsymbol{x}) d\boldsymbol{x}. \tag{4.3}$$

where $D(a|b)$ is viewed as penalized Kullback-Leibler distance between $a(\boldsymbol{x})$ and $b(\boldsymbol{x})$:

$$D(a|b) = \int_\Omega \left[a(\boldsymbol{x}) \log \frac{a(\boldsymbol{x})}{b(\boldsymbol{x})} + b(\boldsymbol{x}) - a(\boldsymbol{x}) \right] d\boldsymbol{x}$$

and the term $-\sum_{j=1}^m \lambda_j \int_\Omega \mathcal{N} f_j(\boldsymbol{x}) d\boldsymbol{x}$ can be viewed like a penalization term (cf. Eggermont [1999]).

The goal is to define an iterative algorithm that possesses a descent property with respect to the functional $\mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda})$. We start by defining a function $b(\boldsymbol{\theta}|\boldsymbol{\theta}^0) = b^0(\boldsymbol{\theta})$, which depends also on the parameter $\boldsymbol{\theta}^0$ which denotes the current values we use in our iterative algorithm so that when shifted by a constant, it majorizes $\mathcal{L}(\boldsymbol{\theta})$.

For $j = 1, \ldots, m$, let

$$w_j^0(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{\lambda_j^0 \mathcal{N} f_j^0(\boldsymbol{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0} \mathcal{N} \boldsymbol{f}^0(\boldsymbol{x})} \tag{4.4}$$

be the "weight" functions and w_j^0's satisfy $\sum_j w_j^0(\boldsymbol{x}) = 1$. The majorizing function

$$b^0(\boldsymbol{\theta}) = b^0(\boldsymbol{f}, \boldsymbol{\lambda}) \stackrel{\text{def}}{=} -\int_\Omega g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x}) \log[\lambda_j \mathcal{N} f_j(\boldsymbol{x})] d\boldsymbol{x} \tag{4.5}$$

satisfies

$$b^0(\boldsymbol{\theta}) - b^0(\boldsymbol{\theta}^0) \geq \mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}(\boldsymbol{\theta}^0). \tag{4.6}$$

which comes from convexity of $-\log$ function since $\sum_j^m w_j^0(\boldsymbol{x}) = 1$ and Jensen's inequality. Indeed,

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}(\boldsymbol{\theta}^0) = \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{f}) - \mathcal{L}(\boldsymbol{\lambda}^0, \boldsymbol{f}^0) &= -\int g(\boldsymbol{x}) \log \frac{\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}(\boldsymbol{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0} \mathcal{N} \boldsymbol{f}^0(\boldsymbol{x})} d\boldsymbol{x} \\
&= -\int g(\boldsymbol{x}) \log \frac{\sum_{j=1}^m \lambda_j \mathcal{N} f_j(\boldsymbol{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0} \mathcal{N} \boldsymbol{f}_j^0(\boldsymbol{x})} d\boldsymbol{x} \\
&= -\int g(\boldsymbol{x}) \log \sum_{j=1}^m \frac{\lambda_j^0 \mathcal{N} f_j^0(\boldsymbol{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0} \mathcal{N} \boldsymbol{f}^0(\boldsymbol{x})} \frac{\lambda_j \mathcal{N} f_j(\boldsymbol{x})}{\lambda_j^0 \mathcal{N} f_j^0(\boldsymbol{x})} d\boldsymbol{x} \\
&= -\int g(\boldsymbol{x}) \log \sum_{j=1}^m w_j^0(\boldsymbol{x}) \frac{\lambda_j \mathcal{N} f_j(\boldsymbol{x})}{\lambda_j^0 \mathcal{N} f_j^0(\boldsymbol{x})} d\boldsymbol{x} \\
&\leq -\int g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x}) \log \frac{\lambda_j \mathcal{N} f_j(\boldsymbol{x})}{\lambda_j^0 \mathcal{N} f_j^0(\boldsymbol{x})} d\boldsymbol{x} \\
&= b^0(\boldsymbol{\lambda}, \boldsymbol{f}) - b^0(\boldsymbol{\lambda}^0, \boldsymbol{f}^0).
\end{aligned}$$

■

Rewriting (4.5) we obtain

$$
\begin{aligned}
b^0(\boldsymbol{f},\boldsymbol{\lambda}) &\stackrel{\text{def}}{=} -\int_\Omega g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x})\ \log[\lambda_j \mathcal{N} f_j(\boldsymbol{x})] d\boldsymbol{x} \\
&= -\int_\Omega g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x})\ \log[\mathcal{N} f_j(\boldsymbol{x})] d\boldsymbol{x} - \int_\Omega g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x})\ \log \lambda_j d\boldsymbol{x} \\
&= -\int_\Omega g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x})\ \log \left[\prod_{\ell=1}^B \mathcal{N} f_{j\ell}(\underline{x}_{s_\ell})\right] d\boldsymbol{x} - \int_\Omega g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x})\ \log \lambda_j d\boldsymbol{x} \\
&= -\int_\Omega g(\boldsymbol{x}) \sum_{j=1}^m w_j^0(\boldsymbol{x})\ \sum_{\ell=1}^B \log \left[\exp \int_{\Omega_\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{s_\ell} - \underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}\right] d\boldsymbol{x} \\
&\quad - \sum_{j=1}^m \log \lambda_j \int_\Omega g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x} \\
&= -\sum_{j=1}^m \sum_{\ell=1}^B \int_\Omega \left[\int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{s_\ell} - \underline{u}_{s_\ell}) g(\boldsymbol{x}) w_j^0(\boldsymbol{x})\ \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}\right] d\boldsymbol{x} \\
&\quad - \sum_{j=1}^m \log \lambda_j \int_\Omega g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x} \\
&\stackrel{\text{def}}{=} \sum_{j=1}^m \sum_{\ell=1}^B b_{j\ell}^0(f_{j\ell}) + b_{\lambda_j}^0(\lambda_j),
\end{aligned}
$$

where the right hand side of the last equation is the sum of separate functions of the individual $f_{j\ell}$'s and λ_j's.

Theorem 5. *If we define*

$$\widehat{\lambda}_j = \int_\Omega g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x} \tag{4.7}$$

and

$$\widehat{f}_{j\ell}(\underline{u}_{s_\ell}) = \alpha_{j\ell} \int_\Omega K_{H_\ell}(\underline{x}_{s_\ell} - \underline{u}_{s_\ell}) g(\boldsymbol{x}) w_j^0(\boldsymbol{x}) d\boldsymbol{x}, \quad \underline{u}_{s_\ell} \in \mathbb{R}^{d_\ell}, \tag{4.8}$$

where $\alpha_{j\ell}$ is a constant chosen so that $\int_{\Omega^\ell} \widehat{f}_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} = 1$. Then $b^0(\cdot)$ is minimized by the corresponding piece of $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{\lambda}})$ and the newly updated $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{\lambda}})$ satisfies the descent property:

$$\mathcal{L}(\widehat{\boldsymbol{\theta}}) \leq \mathcal{L}(\boldsymbol{\theta}^0).$$

Proof.

The proof here is nearly identical to a result of Levine et al. [2011] except that we handle here the fact that $\widehat{f}_{j\ell}(\underline{u}_{s_\ell})$ is a $d_\ell-$dim multivariate function as in (4.8).

Fubini's theorem yields

$$\begin{aligned} b^0_{j\ell}(f_{j\ell}) &= -\int_\Omega \left[\int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{s_\ell} - \underline{u}_{s_\ell}) g(\boldsymbol{x}) w^0_j(\boldsymbol{x}) \ \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}\right] d\boldsymbol{x} \\ &= -\int_{\Omega^\ell} \widehat{f}_{j\ell}(\underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} \\ &= \frac{1}{\alpha_{j\ell}} D(\widehat{f}_{j\ell} | f_{j\ell}) - \frac{1}{\alpha_{j\ell}} \int_{\Omega^\ell} \widehat{f}_{j\ell}(\underline{u}_{s_\ell}) \log \widehat{f}_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}, \end{aligned}$$

where the second term on the right hand side does not depend on $f_{j\ell}$. Then $\widehat{f}_{j\ell}$ is the unique (up to changes on a set of Lebesgue measure zero) density function minimizing $b^0_{j\ell}(\cdot)$

To minimize $b^0(\boldsymbol{f}, \boldsymbol{\lambda})$ with respect to the $\boldsymbol{\lambda}$ parameter, we define in the Lagrange multiplier method

$$\underset{\boldsymbol{\lambda}}{\operatorname{argmax}}\, b^0(\boldsymbol{\lambda}) = \sum_{j=1}^m A_j \log(\lambda_j), \quad \text{with constraint } \Psi(\boldsymbol{\lambda}) = \sum_{j=1}^m \lambda_j - 1 = 0.$$

where $A_j = \int_\Omega g(\boldsymbol{x}) w^0_j(\boldsymbol{x}) d\boldsymbol{x}$ and $\sum_{j=1}^m A_j = 1$.

Then the solution differentiating $L(\boldsymbol{\lambda}, \alpha) = b^0(\boldsymbol{\lambda}) + \alpha\Psi(\boldsymbol{\lambda})$ is

$$\widehat{\lambda}_j = \frac{A_j}{\sum_{j=1}^m A_j}, \quad j = 1, \ldots, m.$$

We may also conclude that $b^0(\cdot)$ is minimized by the corresponding piece of $(\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{\lambda}})$ and the newly updated $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{f}}, \widehat{\boldsymbol{\lambda}})$ satisfies the inequality (4.6)

$$\mathcal{L}(\widehat{\boldsymbol{\theta}}) - \mathcal{L}(\boldsymbol{\theta}^0) \leq b^0(\widehat{\boldsymbol{\theta}}) - b^0(\boldsymbol{\theta}^0) \leq 0$$

which proves the descent property.

■

4.4 Estimation of the Parameters

We now assume that we observe a simple random sample $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ distributed according to some r-dimensional density $g(\boldsymbol{x})$. Letting $\tilde{G}_n(\cdot)$ denote the empirical distribution function of the sample and ignoring the term $\int g(\boldsymbol{x}) \log g(\boldsymbol{x}) d\boldsymbol{x}$ that does not involve any parameters, we consider a discrete version of (4.2):

$$\begin{aligned} \mathcal{L}_n(\boldsymbol{\theta}) = \mathcal{L}_n(\boldsymbol{f}, \boldsymbol{\lambda}) &\overset{\text{def}}{=} \int \log \frac{1}{[\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}](\boldsymbol{x})} d\tilde{G}_n(\boldsymbol{x}) = -\sum_{i=1}^n \log\{[\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}](\boldsymbol{x}_i)\}. \\ &= -\sum_{i=1}^n \log \sum_{j=1}^m \lambda_j \prod_{\ell=1}^B \mathcal{N} f_{j\ell}(\underline{x}_{is_\ell}) \\ &= -\sum_{i=1}^n \log \sum_{j=1}^m \lambda_j \prod_{\ell=1}^B \exp \int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{is_\ell} - \underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}. \end{aligned}$$

$\mathcal{L}_n(\boldsymbol{\theta})$ here resembles a penalized loglikelihood function except for the presence of the nonlinear smoothing operator $\mathcal{N}$. We may rewrite $\mathcal{L}_n(\boldsymbol{\theta})$ as

$$\mathcal{L}_n(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \log \sum_{j=1}^{m} \lambda_j \exp\left\{ \sum_{\ell=1}^{B} \int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{is_\ell} - \underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} \right\}. \tag{4.9}$$

We may show that the following algorithm results in an MM algorithm in which the value of $\mathcal{L}_n(\cdot)$ decreases at each iteration.

With initial parameter values $\boldsymbol{\theta}^0 = (\boldsymbol{f}^0, \boldsymbol{\lambda}^0)$ and the fixed $d_\ell \times d_\ell$ bandwidths matrices $H_\ell = \text{diag}(h_1^2, ..., h_{d_\ell}^2)$ for ℓth block, the modified Maximum Smoothed Likelihood (MSL) algorithm iterates the following steps for $t = 0, 1, ...$:

4.4.1 MSL algorithm with conditionally independent multivariate blocks

- **Majorization step:** Define, for each i and j:

$$w_{ij}^{(t)} = \frac{\lambda_j^{(t)} \mathcal{N} f_j^{(t)}(\boldsymbol{x_i})}{\sum_{a=1}^{m} \lambda_a^{(t)} \mathcal{N} f_a^{(t)}(\boldsymbol{x_i})} = \frac{\lambda_j^{(t)} \prod_{\ell=1}^{B} \mathcal{N} f_{j\ell}^{(t)}(\underline{x}_{is_\ell})}{\sum_{a=1}^{m} \lambda_a^{(t)} \prod_{\ell=1}^{B} \mathcal{N} f_{a\ell}^{(t)}(\underline{x}_{is_\ell})}, \tag{4.10}$$

  where

$$\mathcal{N} f_{j\ell}^{(t)}(\underline{x}_{is_\ell}) = \exp \int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{is_\ell} - \underline{u}_{s_\ell}) \log f_{j\ell}^{(t)}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}.$$

- **Minimization step 1:**

$$\lambda_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} w_{ij}^{(t)}. \tag{4.11}$$

- **Minimization step 2:**

$$f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) = \frac{1}{n\lambda_j^{(t+1)}} \sum_{i=1}^{n} w_{ij}^{(t)} \boldsymbol{K}_{H_\ell}(\underline{u}_{s_\ell} - \underline{x}_{is_\ell}), \quad \underline{u}_{s_\ell} \in \mathbb{R}^{d_\ell}, \tag{4.12}$$

where $\boldsymbol{K}_{H_{j\ell}}$ is a multivariate kernel density function, typically Gaussian.

Note that equations (4.10), (4.11) and (4.12) are merely the discrete versions of equations (4.4), (4.7) and (4.8), respectively.

4.4.2 The Descent Property

Theorem 6. *$\mathcal{L}_n(\boldsymbol{\theta}^{(t)})$ is non-increasing in t using the MSL algorithm. In other words, equations* (4.10) *through* (4.12) *ensure the descent property:*

$$\mathcal{L}_n(\boldsymbol{\theta}^{(t+1)}) \leq \mathcal{L}_n(\boldsymbol{\theta}^{(t)}).$$

Proof.
For a given (fixed) $\boldsymbol{\theta}^{(t)}$, let the constants $w_{ij}^{(t)}$ be defined as in Equation (4.10), we first define the finite-sample version of Eq. (4.5) at iteration t:

$$b_n^{(t)}(\boldsymbol{\theta}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \log[\lambda_j \mathcal{N} f_j(\boldsymbol{x}_i)].$$

It is the sum of separate function of the $f_{j\ell}$ and λ_j. Indeed,

$$\begin{aligned}
b_n^{(t)}(\boldsymbol{\theta}) &= -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \log[\mathcal{N} f_j(\boldsymbol{x}_i)] - \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \log \lambda_j \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \log\left[\prod_{\ell=1}^{B} \mathcal{N} f_{j\ell}(\underline{x}_{is_\ell})\right] - \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \log \lambda_j \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \sum_{\ell=1}^{B} \log\left[\exp \int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{is_\ell} - \underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}\right] \\
&\quad -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \log \lambda_j \\
&= -\sum_{j=1}^{m}\sum_{\ell=1}^{B}\sum_{i=1}^{n} w_{ij}^{(t)} \left[\int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{is_\ell} - \underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell}\right] - \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \log \lambda_j.
\end{aligned}$$

Lemma 3. *If* $\boldsymbol{\theta}^{(t+1)} = (\boldsymbol{f}^{(t+1)}, \boldsymbol{\lambda}^{(t+1)})$ *where* $\boldsymbol{f}^{(t+1)}$ *and* $\boldsymbol{\lambda}^{(t+1)}$ *are defined as in* (4.12) *and* (4.11) *respectively, then* $\boldsymbol{\theta}^{(t+1)}$ *minimizes* $b_n^{(t)}(\boldsymbol{\theta})$.

Proof: As a function of $\boldsymbol{\lambda}$, with the constraint $\sum_j \lambda_j = 1$, the general framework of Lagrange multiplier introduced above include that case, with $A_j = \sum_{i=1}^{n} w_{ij}^{(t)}$. Then for each j the minimizer with respect to $\boldsymbol{\lambda}$ of

$$b_n^{(t)}(\lambda_j) \stackrel{\text{def}}{=} -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^{(t)} \log \lambda_j$$

is given by the equation (4.11).

As a function of $f_{j\ell}$ with the definition of $f_{j\ell}^{(t+1)}$ in (4.12), the penalized Kullback-Leibler distance between $f_{j\ell}^{(t+1)}$ and $f_{j\ell}$ is:

$$\begin{aligned}
D(f_{j\ell}^{(t+1)} | f_{j\ell}) &= \int_{\Omega^\ell} \left[f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) \ \log \frac{f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell})}{f_{j\ell}(\underline{u}_{s_\ell})} + f_{j\ell}(\underline{u}_{s_\ell}) - f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell})\right] d\underline{u}_{s_\ell} \\
&= \int_{\Omega^\ell} \left[f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) \ \log f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) - f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) \ \log f_{j\ell}(\underline{u}_{s_\ell})\right] d\underline{u}_{s_\ell}
\end{aligned}$$

$$\text{(since } \int_{\Omega^\ell} f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} = \frac{1}{n\lambda_j^{(t+1)}} \sum_{i=1}^{n} w_{ij}^{(t)} \int \boldsymbol{K}_{H_\ell}(\underline{u}_{s_\ell} - x_{is_\ell}) d\underline{u}_{s_\ell} = 1 \text{ by (4.11))}.$$

Then the piece involving $f_{j\ell}$ may be written

$$\begin{aligned} b_n^{(t)}(f_{j\ell}) &\stackrel{\text{def}}{=} -\sum_{i=1}^{n} w_{ij}^{(t)} \left[\int_{\Omega^\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{is_\ell} - \underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} \right] \\ &= -\int_{\Omega^\ell} \left[\sum_{i=1}^{n} w_{ij}^{(t)} \boldsymbol{K}_{H_\ell}(\underline{x}_{is_\ell} - \underline{u}_{s_\ell}) \right] \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} \\ &= -\int_{\Omega^\ell} n\lambda_j^{(t+1)} f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) \log f_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} \text{ (by (4.12))} \\ &= n\lambda_j^{(t+1)} \left[D(f_{j\ell}^{(t+1)} | f_{j\ell}) - \int_{\Omega^\ell} f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) \log f_{j\ell}^{(t+1)}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} \right], \end{aligned}$$

where the second term on the right hand side does not depend on $f_{j\ell}$. Thus, $f_{j\ell}^{(t+1)}$ is the unique density function minimizing $b_n^{(t)}(f_{j\ell})$ $\square$.

Lemma 4. *Let $\mathcal{L}_n(\boldsymbol{\theta})$ be defined as in Equation (4.9). Then*

$$\mathcal{L}_n(\boldsymbol{\theta}) - \mathcal{L}_n(\boldsymbol{\theta}^{(t)}) \leq b_n^{(t)}(\boldsymbol{\theta}) - b_n^{(t)}(\boldsymbol{\theta}^{(t)}).$$

Proof:

$$\begin{aligned} &\mathcal{L}_n(\boldsymbol{\theta}) - \mathcal{L}_n(\boldsymbol{\theta}^{(t)}) \\ &= -\sum_{i=1}^{n} \log\{[\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}](\boldsymbol{x}_i)\} + \sum_{i=1}^{n} \log\{[\mathcal{M}_{\boldsymbol{\lambda}^{(t)}} \mathcal{N} \boldsymbol{f}^{(t)}](\boldsymbol{x}_i)\} \\ &= -\sum_{i=1}^{n} \log \frac{[\mathcal{M}_{\boldsymbol{\lambda}} \mathcal{N} \boldsymbol{f}](\boldsymbol{x}_i)}{[\mathcal{M}_{\boldsymbol{\lambda}^{(t)}} \mathcal{N} \boldsymbol{f}^{(t)}](\boldsymbol{x}_i)} \\ &= -\sum_{i=1}^{n} \log \sum_{j=1}^{m} \frac{\lambda_j \mathcal{N} f_j(\boldsymbol{x}_i)}{\sum_{a=1}^{m} \lambda_a^{(t)} \mathcal{N} f_a^{(t)}(\boldsymbol{x}_i)} \times \frac{\lambda_j^{(t)} \mathcal{N} f_j^{(t)}(\boldsymbol{x}_i)}{\lambda_j^{(t)} \mathcal{N} f_j^{(t)}(\boldsymbol{x}_i)} \\ &= -\sum_{i=1}^{n} \log \sum_{j=1}^{m} w_{ij}^{(t)} \times \frac{\lambda_j \mathcal{N} f_j(\boldsymbol{x}_i)}{\lambda_j^{(t)} \mathcal{N} f_j^{(t)}(\boldsymbol{x}_i)} \\ &\leq -\sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^{(t)} \log \frac{\lambda_j \mathcal{N} f_j(\boldsymbol{x}_i)}{\lambda_j^{(t)} \mathcal{N} f_j^{(t)}(\boldsymbol{x}_i)} \end{aligned}$$

(by the convexity of the negative logarithm function, since for each $i : \sum_j w_{ij}^{(t)} = 1$)

$$\begin{aligned} &= -\sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^{(t)} \log[\lambda_j \mathcal{N} f_j(\boldsymbol{x}_i)] + \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^{(t)} \log[\lambda_j^{(t)} \mathcal{N} f_j^{(t)}(\boldsymbol{x}_i)] \\ &= b_n^{(t)}(\boldsymbol{\theta}) - b_n^{(t)}(\boldsymbol{\theta}^{(t)}) \quad \square. \end{aligned}$$

Combining two above lemmas, we conclude that

$$\mathcal{L}_n(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}_n(\boldsymbol{\theta}^{(t)}) \leq b_n(\boldsymbol{\theta}^{(t+1)}) - b_n^{(t)}(\boldsymbol{\theta}^{(t)}) \leq 0.$$

This is the same "MM trick" of Jensen's inequality as for the infinite sample case.

■

4.4.3 Some convergence properties

The convergence properties we present in this subsection are the extensions to multivariate density functions $f_{j\ell}$'s of the Appendix in Levine et al. [2011].

We prove that, if we hold $\boldsymbol{\lambda}$ fixed and repeatedly iterate equation (4.8), then the sequence of $\boldsymbol{f}$ functions converges to a global minimizer of $\mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda})$ for that value of $\boldsymbol{\lambda}$. Extend $\mathcal{S}$ to $\mathcal{F}$ by defining $\mathcal{S}\boldsymbol{f} = (\mathcal{S}f_1, ..., \mathcal{S}f_m)^\top$. Assume that $K(\cdot)$ is strictly positive, we define the subset $\mathcal{B} \subset \mathcal{F}$ by

$$\mathcal{B} = \left\{ \mathcal{S}\phi : 0 \leq \phi \in \mathcal{F} \text{ and } \int_\Omega \phi_j(\boldsymbol{x}) d\boldsymbol{x} = 1 \text{ for all } j \right\}.$$

We are defining the suitable $\phi^0_{j\ell}(\underline{u}_{s_\ell})$ such that

$$\widehat{f}_{j\ell}(\underline{u}_{s_\ell}) = \mathcal{S}\phi^0_{j\ell}(\underline{u}_{s_\ell}), \quad \underline{u}_{s_\ell} \in \mathbb{R}^{d_\ell}, \tag{4.13}$$

where $\widehat{f}_{j\ell}(\underline{u}_{s_\ell})$ is defined in equation (4.8).

The integral of the right hand side of equation (4.13) is

$$\int_{\Omega_\ell} \widehat{f}_{j\ell}(\underline{u}_{s_\ell}) d\underline{u}_{s_\ell} = \int_{\Omega_\ell} \left[\int_{\Omega_\ell} \boldsymbol{K}_{H_\ell}(\underline{u}_{s_\ell} - \underline{x}_{s_\ell}) \phi^0_{j\ell}(\underline{x}_{s_\ell}) dx_{s_\ell} \right] d\underline{u}_{s_\ell}$$

and the integral of the left hand side equivalents

$$\int_{\Omega_\ell} \alpha_{j\ell} \left\{ \int_{\bigotimes_{\ell'=1,\ell'\neq\ell}^B \Omega_{\ell'}} \left[\int_{\Omega_\ell} \boldsymbol{K}_{H_\ell}(\underline{x}_{s_\ell} - \underline{u}_{s_\ell}) d\underline{x}_{s_\ell} \right] g(\boldsymbol{x}) w^0_j(\boldsymbol{x}) \left(\prod_{\ell'=1,\ell'\neq\ell}^B d\underline{x}_{s_{\ell'}} \right) \right\} d\underline{u}_{s_\ell}.$$

Combine these above equations and the definition of $\alpha_{j\ell}$ we conclude that

$$\phi^0_{j\ell}(\underline{x}_{s_\ell}) \stackrel{\text{def}}{=} \alpha_{j\ell} \int_{\bigotimes_{\ell'=1,\ell'\neq\ell}^B \Omega_{\ell'}} g(\boldsymbol{x}) w^0_j(\boldsymbol{x}) d\underline{x}_{s_1} ... d\underline{x}_{s_{\ell-1}} d\underline{x}_{s_{\ell+1}} ... d\underline{x}_{s_B}$$

must integrate by one because of the definition of $\alpha_{j\ell}$. Also, $\phi^0_j(\boldsymbol{x}) = \prod_{\ell=1}^B \phi^0_{j\ell}(\underline{x}_{s_\ell})$ must integrate to one.

Therefore, $\mathcal{B}$ will contain the whole sequence $\boldsymbol{f}^0, \boldsymbol{f}^1, \boldsymbol{f}^2, ...$ except possibly the initial $\boldsymbol{f}^0$, where each element in the sequence is defined by applying the formula of (4.8) to the preceding element. Suppose we fix $\boldsymbol{\lambda}^0$ and consider the sequence

$$(\boldsymbol{f}^0, \boldsymbol{\lambda}^0), (\boldsymbol{f}^1, \boldsymbol{\lambda}^0), (\boldsymbol{f}^2, \boldsymbol{\lambda}^0), ... \tag{4.14}$$

For any $(f_1, ..., f_m) \in \mathcal{B}$, f_j is bounded below by $\inf_{\boldsymbol{x}\in\Omega} K(\boldsymbol{x}) > 0$ since $K(\cdot)$ is positive, so the function $\boldsymbol{f} \mapsto \mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda})$ is defined on $\mathcal{B}$ and then $\mathcal{N}\boldsymbol{f}$ is well-defined for $\boldsymbol{f} \in \mathcal{B}$.

The function $\boldsymbol{f} \mapsto \mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda})$ is uniformly bounded from below on $\mathcal{B}$. The lower semi-continuity combined with strict convexity imply that for any fixed $\boldsymbol{\lambda}$, the sequence (4.14) converges to a global minimizer of the functional $(\boldsymbol{f}, \boldsymbol{\lambda})$. These above properties are proven as in the Appendix of Levine et al. [2011], we do not repeat the proof here for

brevity. As a practical matter, this means that we could essentially replace $\mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda})$ by the profile loglikelihood:

$$\mathcal{L}^\star(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \inf_{\boldsymbol{f} \in \mathcal{B}} \mathcal{L}(\boldsymbol{f}, \boldsymbol{\lambda})$$

because the minimization on the right-hand side may be accomplished by iterating the formula of (4.8) until convergence.

The fact that MSL satisfies a monotonic property but for an objective function that is a smoothed loglikelihood, hence convergence in the statistical sense of the MSL estimate to the MLE is not guaranteed. As in Levine et al. [2011], there is still that gap about how the smoothed loglikelihood behaves when the bandwidth h goes to zero as n goes to infinity. This has been addressed by Eggermont [1999] but with a technique that does not pass to mixture. This is where the doubly-smoothed idea can be studied (see in the Discussion and Perspectives Chapter).

4.5 Implementation

The implementation study compares the `mvnpEM` algorithm with the new algorithm `mvnpMSL` using the same examples proposed in Chapter 3 and Chauveau and Hoang [2016]. We recall and list briefly these models in section 4.5.1.

4.5.1 Three simulated and one actual examples

Model A: simple Gaussian data This is a $r = 4$ variates, $B = 3$ blocks, $m = 2$ components Gaussian mixture (see detail in Section 3.5.2).

Model B: Gaussian, heavy-tailed and skewed data This is a $r = 6$-coordinate, $m = 3$-component model with $B = 3$ bivariate blocks using the full potential of our approach: one bivariate Gaussian block, one bivariate block with heavy-tailed (Student) distributions, and one bivariate block with heavy-tailed and severely skewed distributions (see Section 3.5.3).

Model C: Non-linear dependence within clusters (banana data) Here is a non-linear mixture model with $m = 2$ components, $B = 3$ bivariate blocks ($r = 6$ coordinates). The banana-shaped density is described in detail in Section 3.5.4. Figure 4.1 shows a typical pairplot for Model C.

Model D: Wiscosin Diagnostic Breast Cancer (WDBC) data The Wisconsin Breast Cancer datasets from the UCI Machine Learning Repository is used to distinguish malignant (cancerous) from benign (non-cancerous) samples (see more detail in Section 3.1 and Section 3.6). We considered the first ten features of the Mean and keep the same block structure of these feature as in Section 3.6. We then applied our smoothed algorithm `mvnpMSL` on this $r = 10$ dimensional dataset. The total block is $B = 5$.

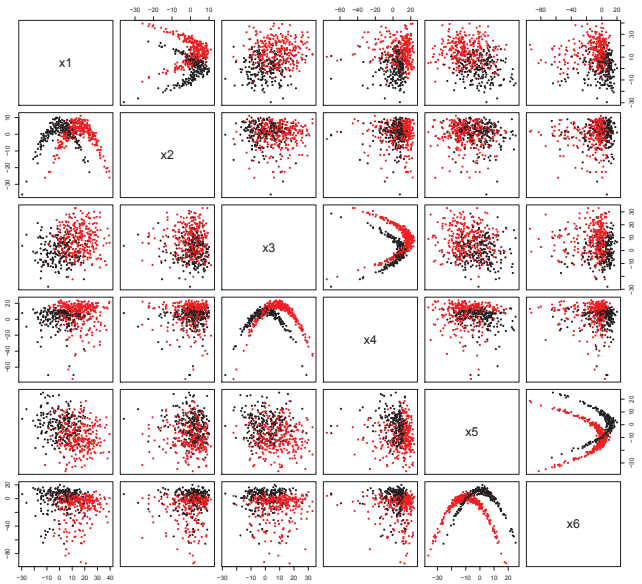


Figure 4.1 – Pairs plot for Model C

4.5.2 The approximating integrals and the monotony of loglikelihood function

The E-step of the `mvnpMSL` algorithm requires the discretization of the intervals over which are approximated the multivariate integrals for non linear smoothing of the log densities. This is the major difference with the previous MSL that already exists in Levine et al. [2011] because of the nonlinear smoothing of multivariate $f_{j\ell}$'s instead of the univariate $\mathcal{N}f_{jk}$'s. It is also the huge difference in computing task between both versions: `mvnpEM` and `mvnpMSL` algorithm. To see that, we made some trials on the simple model B (all gaussians, $r = 6$ coordinates, 3 bivariate blocks with correlation $\rho = 0.25$, $n = 50$, where "easy" means separated enough so that all algorithms require only 3 iterations. Then we compared the CPU time betweeA simulated 6-dimensional example (Model B) (75% within−blocks & component correlation)A simulated 6-dimensional example (Model B) (75% within−blocks & component correlation)n our code versions (these codes are different at step of computing the multivariate integrals which were done in `R` or `C` language):

- Our very first all-in-R code: 1600 s
- Some others improved codes: 360 s
- Our last all-in-C code version: 2.16 s
- `mvnpEM`: 0.004 s

It seems a very good improvement in terms of CPU time.

We define the multivariate grid for integration inside the code, from the min and the max of the data per columns. Let *ngrid* is the same number of points in the discretization of these intervals. We made an experiment to see the influence of *ngrid* by increasing

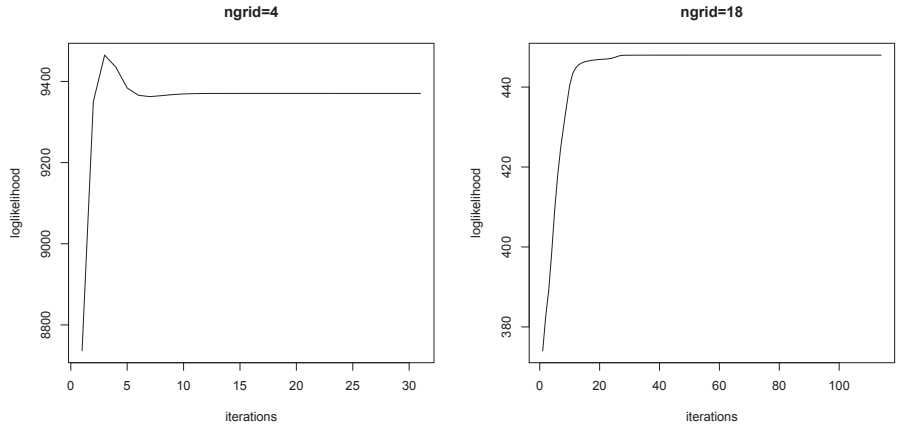


Figure 4.2 – The behavior of the loglik and pseudo-loglik sequence for monotony of WDBC data.

the numbers of the point *ngrid* in each grid dimension as in the Table 4.1. The criterion considered here is corrected classification. Model-Based Clustering and Classification using mixture model is based on the Maximum A Posterior (MAP) strategy.

The results show that the cases of $ngrid = 10$ (for Model A), $ngrid = 50$ (for Model B), $ngrid = 18$ (for Model C) and $ngrid = 18$ (for Model D) are the best cases. In the ℓth block with dimension is d_ℓ, the number of the points in the discretization will be $ngrid^{d_\ell}$. For instance, in Model C the largest dimension of the blocks is 3. Then the best "smooth" grid has $ngrid^3$ meshes. This explains why the CPU time increases nonlinearly when *ngrid* increases and we have to make a trade-off between the CPU times and the grid size to get a workable smooth algorithm.

Additionally, we can see the poor grid size not only give the poor result in the corrected classification but also destroys the ascent property of the loglikelihood objective function as in Figure 4.2.

We also plot the smoothed loglik $\mathcal{L}(\boldsymbol{\theta})$ of Model A along iterations. It increases (ascent property) in same bandwidth case with `mvnpMSL` algorithm. This monotony property will be destroyed in adaptive bandwidth case as in Figure 4.3.

4.5.3 Monte Carlo experiments

In our Monte-Carlo experiments, we computed the errors in terms of the square root of the Mean Integrated Squared Error (MISE) for the densities. and the mean squared error (MSE) for the proportions that are the only scalar parameters in these models

We kept the experiment settings as in Section 3.5 and Chauveau and Hoang [2016]: $S = 300$ replications of $n = 500$ observations each, where λ_1 is varying from 0.1 to 0.4 (for Model A) and S = 300 replications of samples of sizes $n = 400, 600, 800, 1000$ (for Model B). Figure 4.4 and 4.5 point out that our smooth method and `mvnpEM` algorithm gave the same results in term of the errors MISE of the densities estimates and MSE of the scalar parameters estimates. Our smoothed method is also slightly better than `mvnpEM` in

Model A	$\widehat{\lambda}_1$	$\widehat{\lambda}_2$	% correctly-classified	# iterations	CPU times (s)
$ngrid = 2$	0.4577	0.5423	63.4	500	1.392
$ngrid = 3$	0.380	0.620	68.8	56	0.333
$ngrid = 5$	0.3942	0.6058	98.0	26	0.345
$ngrid = 8$	0.3807	0.6193	99.8	18	0.478
ngrid=10	0.3812	0.6188	99.6	14	0.576
$ngrid = 15$	0.381	0.619	99.6	14	1.093
$ngrid = 20$	0.381	0.619	99.6	14	1.508
$ngrid = 50$	0.381	0.619	99.6	14	7.137
$ngrid = 150$	0.381	0.619	99.6	14	60.682
$ngrid = 200$	0.381	0.619	99.6	14	107.272
$ngrid = 300$	0.381	0.619	99.6	14	239.993
$ngrid = 500$	0.381	0.619	99.6	14	663.462
$ngrid = 1000$	0.381	0.619	99.6	14	2639.721

Model B	$\widehat{\lambda}_1$	$\widehat{\lambda}_2$	$\widehat{\lambda}_3$	% correctly-classified	# iterations	CPU times (s)
$ngrid = 2$	0.1567	0.4071	0.4361	16.2	33	0.169
$ngrid = 4$	0.1092	0.2434	0.6474	37.0	24	0.396
$ngrid = 10$	0.1619	0.3945	0.4436	43.6	83	7.987
$ngrid = 15$	0.162	0.3271	0.5109	45.6	62	13.066
$ngrid = 18$	0.162	0.3171	0.5209	94.0	116	34.86
$ngrid = 20$	0.162	0.3127	0.5253	94.6	116	43.352
ngrid=50	0.162	0.2984	0.5396	95.4	174	400.204
$ngrid = 150$	0.162	0.2983	0.5397	95.4	166	3428.401
$ngrid = 200$	0.162	0.2983	0.5397	95.4	165	6051.996
$ngrid = 300$	0.162	0.2983	0.5397	95.4	106	8728.074
$ngrid = 400$	0.162	0.2983	0.5397	95.4	108	15805.470

Model C	$\widehat{\lambda}_1$	$\widehat{\lambda}_2$	% correctly-classified	# iterations	CPU times (s)
$ngrid = 5$	0.4895	0.5105	80.0	50	0.557
$ngrid = 10$	0.4052	0.5948	96.6	71	2.821
$ngrid = 15$	0.4015	0.5985	98.6	61	5.411
ngrid=18	0.3997	0.6003	99.4	60	7.673
$ngrid = 19$	0.4001	0.5999	99.0	83	11.878
$ngrid = 20$	0.4007	0.5993	98.8	80	12.477
$ngrid = 22$	0.4006	0.5994	99.0	46	8.696
$ngrid = 25$	0.4005	0.5995	98.8	83	20.113
$ngrid = 26$	0.4005	0.5995	98.8	83	21.783
$ngrid = 27$	0.4005	0.5995	98.8	84	23.893
$ngrid = 30$	0.4005	0.5995	99	43	15.17
$ngrid = 50$	0.4005	0.5995	99	83	81.018
$ngrid = 70$	0.4005	0.5995	99	43	82.126
$ngrid = 100$	0.4005	0.5995	99	43	170.134

Model D	B (over 357)	M (over 212)	% correctly-classified	# iterations	CPU times (s)
$ngrid = 4$	329	79	71.705	31	0.805
$ngrid = 5$	255	144	70.121	68	3.183
$ngrid = 10$	347	168	90.510	31	10.61
$ngrid = 15$	340	186	92.443	135	152.434
ngrid=18	348	184	93.497	114	229.92
$ngrid = 19$	344	186	93.146	163	436.175
$ngrid = 20$	344	186	93.146	303	755.619
$ngrid = 22$	344	185	92.970	200	754.122
$ngrid = 25$	343	185	92.794	212	1131.697
$ngrid = 30$	343	185	92.794	205	2013.846
$ngrid = 50$	343	185	92.794	204	8451.637

Table 4.1 – Comparing the % of correct classification and the proportion estimates of Model A, Model B, Model C, sample size $n = 500$, Ten first features of Model D (bottom), using `mvnpMSL` (same bandwidth) and the MAP strategy when changing the grid size.

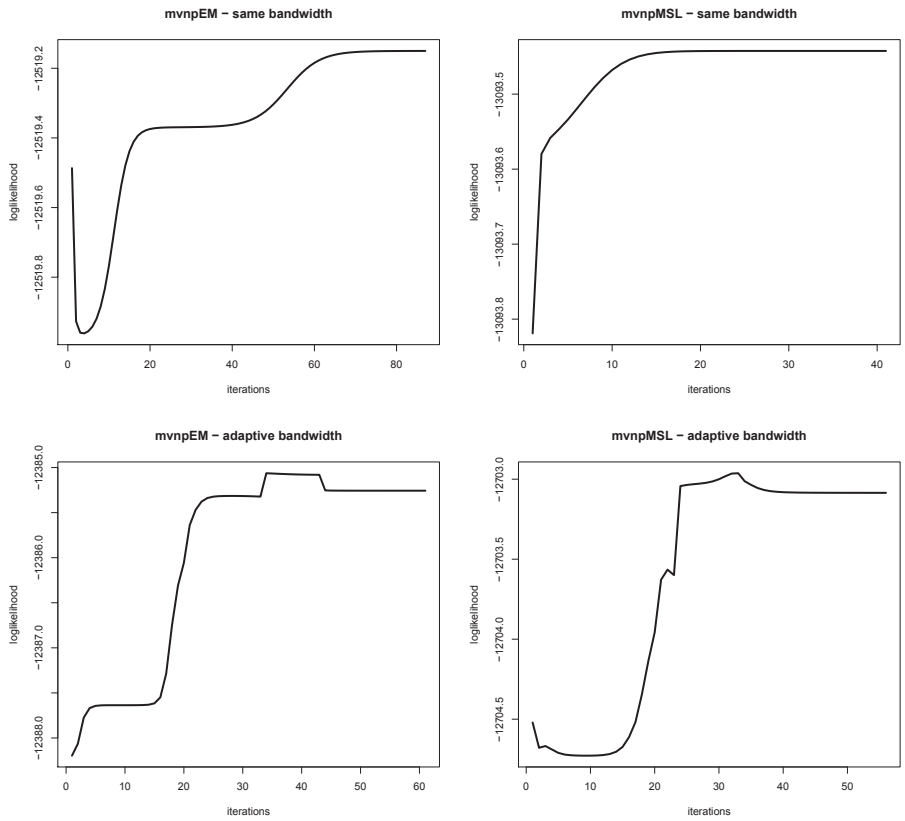


Figure 4.3 – The behavior of the loglik and pseudo-loglik sequence for monotony of Model A (sample size $n = 200$) using `mvnpEM` (left) and `mvnpMSL`, $ngrid = 20$, (right) in 2 cases: same bandwidth (top), adaptive bandwidth (bottom).

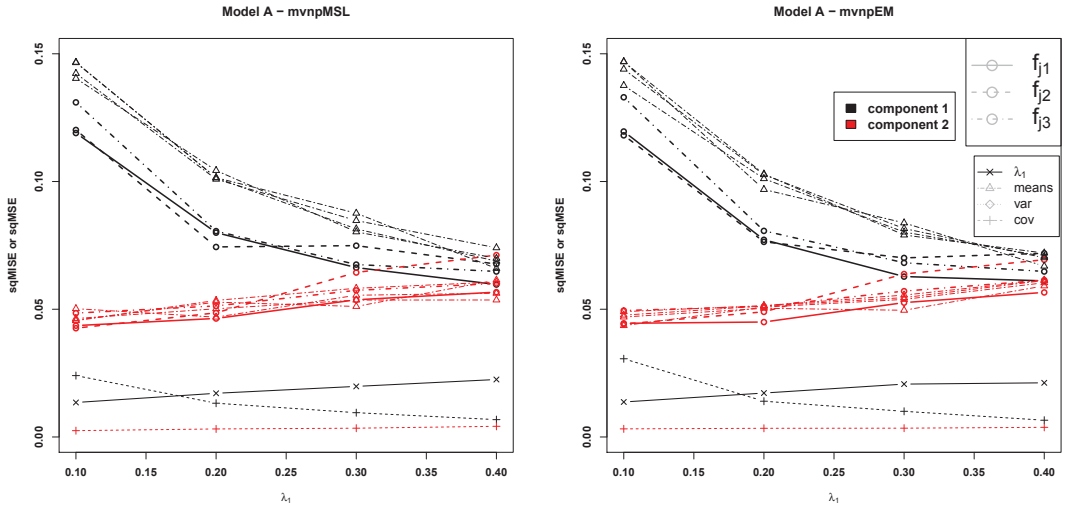


Figure 4.4 – Square roots of MISE for the densities and square roots of MSE for the scalar parameter λ_1, and other scalar measures that are not parameters in the model (means and covariances), as a function of the proportion of the first component λ_1, for Model A, $n = 500$ and $S = 300$ replications, random initialization of 2 algorithms: `mvnpMSL` (on the left) and `mvnpEM` (on the right) in same bandwidth case.

Method	$\lambda_1 = 0.6274$	$\lambda_2 = 0.3726$
`mvnpMSL`	0.6102 0.7137	0.2863 0.3898
`mvnpEM`	0.6130 0.7072	0.2928 0.3870

Table 4.2 – 95 % Confidence Intervals for the true proportion $\boldsymbol{\lambda} = (\lambda_1, \lambda_2) = (0.6274, 0.3726)$, based on $B = 10000$ bootstrap replications, for the WDBC data example, using two methods: `mvnpEM` and `mvnpMSL` with same bandwidth case.

the results of Model B.

For WDBC data, the estimated proportion of these two components for smoothed algorithm (and the corresponding `mvnpEM` estimates in parentheses) are, respectively, 0.338 (0.353), 0.662 (0.647). This observed slight difference between the two algorithms' estimates of these proportions suggests that it might be wise to compute confidence intervals for them. It is possible to obtain confidence intervals on the proportion estimates $\boldsymbol{\lambda}$ using a nonparametric bootstrap approach by repeatedly re-sampling with replacement from the empirical distribution defined by the n observed r-dimensional vectors. Table 4.2 shows the empirical 95% confidence interval for λ_1 and λ_2, respectively using 10000 bootstrap replications.

4.5.4 Clustering efficiency

We compared the performance of `mvnpMSL` algorithm and `mvnpEM` algorithm proposed by Chauveau and Hoang [2016] in terms of the MAP clustering efficiency. Table 4.3 and 4.4 clearly shows that two algorithms give the similar result. It is also better than the result

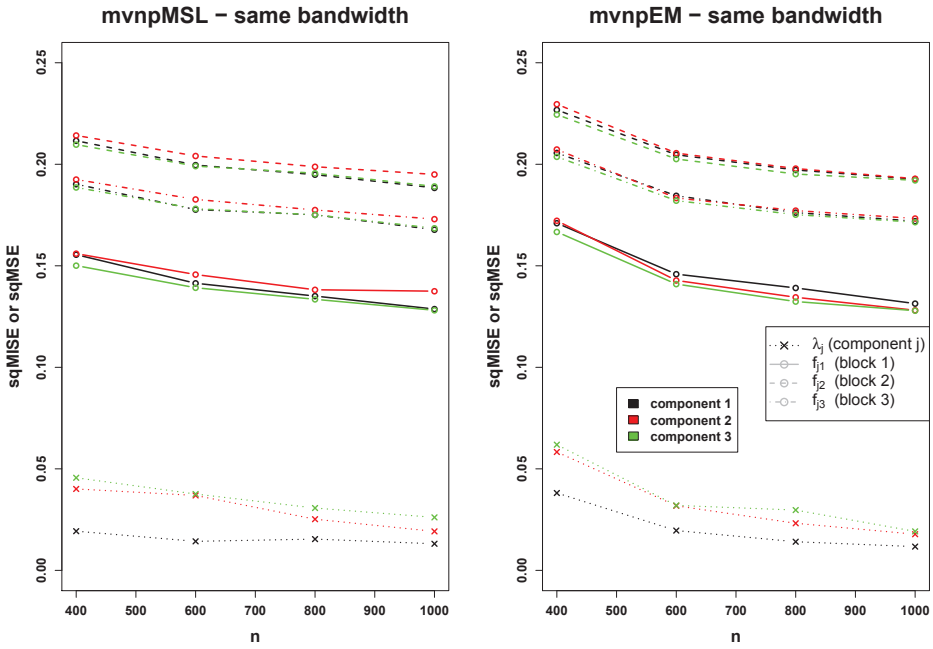


Figure 4.5 – Square roots of MISE's for the densities as a function of the sample size n, $S = 300$ replications, for the two algorithm settings: `mvnpMSL` (left), `mvnpEM` (right) for Model B in same bandwidth case.

Method	%correctly-classified	
	Model B	*Model C*
`mvnpMSL`	92.993	98.99
`mvnpEM`	93.849	98.53
k-means	76.269	68.69

Table 4.3 – The % of correct clustering averaged over $S = 300$ replications for a sample size of $n = 1000$ from Model B and C using `mvnpMSL`/`mvnpEM` and the MAP strategy comparing with the k-means clustering strategy.

from k-means strategy.

Method	B (over 357)	M (over 212)	% correctly-classified
`mvnpMSL`	348	184	93.497
`mvnpEM`	350	183	93.673
k-means	355	122	83.831

Table 4.4 – The % of correct classification of the WDBC data using `mvnpMSL`/`mvnpEM` and the MAP strategy comparing with the k-means clustering strategy.

On several trials we did with the `mvnpMSL` algorithm, we get results very similar to the empirical one `mvnpEM`. So this smoothed version is definitely a usable alternative to the

empirical version. The fact that the smoothed takes more CPU time than the other one requires a possible useful hybrid method to use in practice.

Chapter 5

A multivariate model and mixture approach for FDR estimation

5.1 Introduction

The False Discovery Rate (FDR) is one way of conceptualizing the rate of type I errors in null hypothesis testing when conducting multiple comparisons. FDR controlling procedures are designed to control the expected proportion of rejected null hypotheses that were incorrect rejections. It plays a prominent role in many high dimensional testing and model selection procedures. Several statistical algorithms have been proposed in the literature for estimating the FDR, the recent and unified procedure based on a nonparametric approach from Strimmer [2008b] appearing to be one of the current standards for practitioners.

In hypotheses testing framework, we observe the $p-$value (or probability value), which is the probability of getting a sample statistic (such as the mean) or a more extreme sample statistic in the direction of the alternative hypothesis, given that the value stated in the null hypothesis H_0 is true (not significant or not interesting). Thus, how p-values are defined is important to the performance of the procedure. When we decide whether to retain or reject the null hypothesis, it is possible that a conclusion may be wrong since we are observing a sample and not an entire population. The central problem is the control of type I error (false positive) and type II error (false negative).

		Decision	
		Retain the null	Reject the null
Truth in the population	H_0 True	Correct $1-\alpha$	Type I error α
	H_0 False	Type II error β	Correct $1-\beta$

Table 5.1 – Four outcomes from making a decision.

It is easy to check that, if n independent tests with level of significance α are applied

simultaneously, the achieved Family Wise Error Rate (FWER), that is the probability of observing at least one false rejection among the n tests, is $1-(1-\alpha)^n$ which quickly increases with n and is already $\approx 99\%$ for $n = 100$.

Consider the problem of testing simultaneously n null hypotheses, of which n_0 are true. P/N (Positive/Negative) is the number of hypotheses rejected/accepted. Table 5.2 summarizes the situation of possible outcomes (True Positive, False Positive, True Negative, False Negative) from n testing.

Truth/decision	Accepted H_0	Rejected H_0	Total
H_0 is true	TN	FP	n_0
H_0 is false	FN	TP	$n - n_0$
Total	N	P	n

Table 5.2 – The possible outcomes when testing n hypotheses for H_0.

FDR theory starts with the seminal papers by Schweder and Spjøtvoll [1982], Benjamini and Hochberg [1995],.... Benjamini and Hochberg [1995] suggested that the false discovery rate (FDR), defined as the expected proportion of erroneous rejections among all rejections, may be the appropriate error rate to control the increased type I error when testing simultaneously a family of hypotheses in many applied multiple testing problems.

$$\mathrm{FDR} = \mathbb{E}\left[\frac{\mathrm{FP}}{\max(\mathrm{P},1)}\right] = \mathbb{E}\left[\frac{\mathrm{FP}}{\mathrm{P}}|\mathrm{P}>0\right]\mathbb{P}(\mathrm{P}>0).$$

Storey [2002] defined a new false discovery rate, pFDR, of which the term "positive" has been added to reflect the fact that we are conditioning on the event that positive findings have occurred.

$$\mathrm{pFDR} = \mathbb{E}\left[\frac{\mathrm{FP}}{\mathrm{P}}|\mathrm{P}>0\right].$$

The usual setup for FDR estimation using a mixture model is to consider n iid "cases", where to each case i corresponds the response from a statistical test for some null hypothesis H_0, leading to a p-value $p_i \in]0;1[$. A "small" p-value p_i indicates rejection of H_0, i.e. *significant* cases corresponding to H_1 true, whereas when H_0 is true, $p_i \sim \mathcal{U}_{[0,1]}$. Since it is not observed whether each hypothesis is true or false, we are in the general framework of statistical inference from missing data. Let the missing data $Z_i \in \{1,2\}$ defining the unknown test result, $Z_i = 1$ when H_0 is true and $Z_i = 2$ when it is false.

In multiple testing, we observe a sample of n p-values $p_1, p_2, ..., p_n$ where each individual observation p_i corresponds to the critical probability of the ith test. The FDR control with a mixture model is based on estimation of a $m = 2$–component mixture for the n iid random variable's $\boldsymbol{p} = (p_1, p_2, ..., p_n)$ in which the component pdf associated to H_0 is f_1 known to be Uniform $\mathcal{U}_{[0,1]}$,

$$(p_i|\text{"not interesting"}) = (p_i|H_0 \text{ true}) = (p_i|Z_i = 1) \sim \mathcal{U}_{[0,1]},$$

and the component associated to H_1 is some pdf f_2 with a mass concentrated near 0. The pdf of the p-values is thus

$$f(p) = \lambda + (1-\lambda)f_2(p), \tag{5.1}$$

where λ is the unknown proportion of the true null hypotheses and the FDR control is based on the estimated posterior probabilities.

In a mixture context, the pFDR is given by

$$\text{pFDR}(p_i) = \mathbb{P}(H_0 \text{ true} | P \leq p_i) = \frac{\lambda U(p_i)}{\lambda U(p_i) + (1-\lambda) F_2(p_i)},$$

where U and F_2 are the cumulative distribution functions (cdfs) for densities $\mathcal{U}_{[0,1]}$and f_2, respectively.

Efron et al. [2001] define the local false discovery rate (ℓFDR) to quantify the plausibility of a particular hypothesis being true, given its specific test statistic or p-value. Generally, two distinct types of FDR need to be distinguished: density-based local FDR (ℓFDR) and tail area-based FDR (τFDR). More formally, consider an observed test statistic $y \geq 0$ designed such that a small y indicates an uninteresting null case, and conversely, a large y an interesting alternative case. It is assumed that test statistics y follow a two-component mixture, with density

$$f(y) = \lambda f_1(y|\theta) + (1-\lambda) f_2(y)$$

and distribution function

$$F(y) = \lambda F_1(y|\theta) + (1-\lambda) F_2(y),$$

where θ is the parameters of the pdf and cdf of the null. The local and tail area-based FDR are then defined as follows:

$$\ell FDR = \mathbb{P}(\text{“not interesting”} | Y = y) = \lambda \frac{f_1(y|\theta)}{f(y)},$$

$$\tau FDR = \mathbb{P}(\text{“not interesting”} | Y \geq y) = \lambda \frac{1 - F_1(y|\theta)}{1 - F(y)}.$$

`fdrtool` is a package available online from the Comprehensive R Archive Network (Strimmer [2008a]). This package allows to estimate both tail area-based false discovery rates (Fdr) as well as local false discovery rates (fdr) for a variety of null models (p-values, z-scores, correlation coefficients, t-scores). In contrast to other FDR estimation schemes, in `fdrtool` there is no unnecessary distinction between p-values and other test statistics and regardless of the choice of test statistic, simultaneously both local FDR as well as tail area-based FDR values are estimated.

FDR analysis with `fdrtool` is simple: start the R application (R Development Core Team, 2007), arrange the test statistics in vector format, and run the `fdrtool` command (p is vector of p-values)

```
library(fdrtool)
fdr.out = fdrtool(p, statistic=‘‘pvalue’’)
```

The actual estimated FDR values can be accessed as follows:

```
fdr.out$pval # p-values
fdr.out$lfdr # local FDR
fdr.out$qval # tail area-based FDR
fdr.out$param # estimated parameters
```

Robin et al. [2007] propose to estimate the FDR by computing the average of the $\ell\text{FDR}(p_i)$'s over all the rejected p_i's. Their results were applied in a two component mixture model where one component is known to estimate the posterior population probabilities and the ℓFDR. The unknown part is estimated with a weighted kernel density estimator.

Chauveau et al. [2014] explore a solution for ℓFDR estimation by introducing a specific version of a semi-parametric EM algorithm. It relies on the missing data aspect induced by the mixture. An EM-like algorithm delivers, together with estimates of the mixture parameters, estimates of the posterior probabilities that each p-value comes from each component. The ℓFDR can be computed directly from these posteriors

$$\ell\text{FDR}(p_i) = \mathbb{P}(\text{“not interesting”}|p_i) = \mathbb{P}(Z_i = 1|p_i).$$

Robin et al. [2007]'s work is very close to the semi-parametric EM approach. The difference is that they estimate λ separately, and then estimate f_2 using a weighted kernel density estimate.

To estimate the ℓFDR, it is necessary to estimate the density f_2. Allison et al. [2002] formulate f_2 as a mixture of beta distributions. Liao et al. [2004] proposes a special parametric model tailored to multiple testing by requiring f_2 to be stochastically smaller than f_1, a structure appropriate for multiple testing. A smoothing mechanism is built in. The proposed model provides stable and robust estimation of the ℓFDR for any reasonable form of f_2.

Motivated by the issue of local false discovery rate estimation, Nguyen and Matias [2014] focus on the estimation of the nonparametric unknown component f_2 in the mixture, relying on a preliminary estimator of the unknown proportion θ of true null hypotheses.

Bordes et al. [2006c] considered a special case of model 5.1 where the unknown component belongs to a location family. It was defined as

$$f(p) = \lambda f_1(p) + (1 - \lambda) f_2(p - \mu),$$

where f_1 is known (under the null hypothesis) while f_2 is unknown (under the alternative hypothesis) and symmetric around the non-null location unknown parameter μ. Under some conditions they showed that this kind of model is identifiable and then they proposed an estimation method for the unknown parameters. This model provided a motivation for the work of Shen et al. [2016]. With the assumption that we do not have any informations of the unknown density function Shen et al. [2016] derived a new sufficient identifiability condition and proposed an iterative MM algorithm to estimate the parameters of this model, based on an idea of applying a maximum smoothed likelihood.

Often in experimental design multiple variables are related in such a way that, by analyzing them simultaneously additional information and sometimes essentially information, can be gathered that would be missed if each variable was examined individually.

Hypothesis on high dimensional data involves a sample of random vectors from which a multivariate statistic is derived to capture critical features of the sample.

In this chapter, we first establish a multivariate nonparametric mixture model in multiple testing for False Discovery Rate (FDR) evaluation and then verify its identifiability. We propose in the next section new "EM-like" algorithms, called `mvnpEMN01` (multi-variate non-parametric) since they have one component known and set to multivariate standard normal distribution function. In the implementation section, we conduct numerical study on the FDR control based on the simulated examples. Then, we evaluate the effect of our algorithms on an actual dataset from micro-array experiments.

5.2 Multivariate FDR (mvFDR) model

Assume that for each case $i = 1, 2, ..., n$, $r > 1$ tests are performed (instead of a single test as in the common FDR framework above), these tests being based on r samples, corresponding to different tests of sub hypothesis $H_0^k, k = 1, 2, .., r$. There must be a way to define a "global" hypothesis H_0 of interest with respect to the context, in terms of the individual hypotheses. For instance, a simple model is to asses that H_0 being true for case i means that all the H_0^k are true as well, so that the r individual tests should lead to non significant cases. We can denote this formally by $H_0 = H_0^1 \cdots H_0^r$. Similarly, the global alternative hypothesis has to be specified in the model. For instance, the simplest case is that, when the global H_0 is false, the r tests should lead to rejection: $H_1 = H_1^1 \cdots H_1^r$.

To case i corresponds the r-dimensional observed data $\boldsymbol{p}_i = (p_{i1}, ..., p_{ir})$ and $\boldsymbol{p}$ is the matrix of observations for n cases, with n rows and r columns. A multivariate mixture model can be defined here, and the assumption above can be precised to insure conditional independence assumption required for identifiability of a multivariate nonparametric mixture. Assume that, conditionally to H_0 being true, the r tests responses $(p_{i1}, ..., p_{ir})$ are iid $\sim \mathcal{U}_{[0,1]}$ and that conditionally to H_0 being false they are independent, i.e.

$$(\boldsymbol{p}_i | Z_i = 2) \sim \prod_{k=1}^{r} f_{2k}(p_{ik}),$$

then the pdf of $\boldsymbol{p}_i$ is a multivariate mixture with one component known.

Chi et al. [2008]– the first reference we found of a work related to FDR control with multivariate p-values– give an example to illustrate when multivariate p-values may be useful for pFDR control. They study how to use multivariate statistics to control (p)FDR with good power and propose some rules to reject the nulls. Our approach here is completely different since we use mixture models.

It is easier to work with a transformation that removes the restriction on the range of the values. We consider the probit transform $\boldsymbol{x}_i = \Phi^{-1}(\boldsymbol{p}_i)$ of the p-values since the known component pdf simply becomes $\mathcal{N}(0,1)^{\otimes r}$. Denoting f_k's is the probit transform functions of f_{2k}'s. The model (5.1) may be written

$$g(\boldsymbol{x}_i) = \lambda \prod_{k=1}^{r} \mathcal{N}(0,1)(x_{ik}) + (1-\lambda) \prod_{k=1}^{r} f_k(x_{ik}), \tag{5.2}$$

where component "1" with weight λ is associated to H_0 and component "2", with unspecified nonparametric densities f_k's, to H_1. This multivariate model can be fitted with a specifically designed version of the `npEM` algorithm which has been introduced in Section 2.2.7. Chauveau et al. [2014] proposed two univariate mixture models for FDR estimation which are particular versions of model (5.1).

This $m = 2$-component mixture model, which has the first component is $\mathcal{N}(0,1)$ for all coordinate k, is a special case of model (2.5) which is identifiable under the condition precised in Allman et al. [2009]. The number of coordinates is $r \geq 3$ and for every coordinate $k \in \{1, \cdots, r\}$ the densities $\{\mathcal{N}(0,1), f_k\}$ where f_k is not standard normal distribution, are linearly independent. Such conditions prove the identifiability of the finite mixture (2.5) following Theorem 8 in Allman et al. [2009].

The motivation for this model is that, if multivariate measures and tests are available for a set of global hypotheses H_0 and H_1, then the FDR estimation should be better than the standard univariate framework, since the clustering between interesting/non interesting cases should be more efficient (due to the effect of the conditional independence assumption). In particular, a situation where the global (mv) model should bring some improvement is a situation where the r tests are comparable in terms of power, so that the r p-values are in the same range, and not too obviously leading to rejection, i.e. a situation where the underlying mixtures at the univariate levels are not too obvious, so that a univariate FDR control is not easy, and a multivariate version may take benefit of the "blessing of dimensionality". Conversely, if the significant cases correspond to very small p-values, the mixture at the level of the probit transform is very well-separated, and any univarite FDR control (EM-based or not), should deliver the right answer.

Our first experiments show that this expected behavior hold, i.e. the mvFDR control brings a significative improvement in the case of the simpler $m = 2$-component model for simple global hypotheses (see e.g., example in Section 5.5.1). The question of comparison between uni- and multi-variate FDR is considered in Section 5.4.

5.2.1 More complex mixture models for mvFDR

How many components shoud we set for the mixture model? This is an interesting but delicate question, which is completely determined by the model assumed for the possible behavior of the r tests. In the simplest case as described above, where the r tests correspond to r conditionally independent measures of a same phenomenon, i.e. the corresponding hypothesis are simultaneously either true or false, we have only two possibilities:

$$H_0 = H_0^1 \cdots H_0^r \quad \text{vs.} \quad H_1 = H_1^1 \cdots H_1^r,$$

so that the $m = 2$ components mixture model (5.2) after a probit transform of the $\boldsymbol{p}$'s leading to the data $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ is appropriate.

More complex models in terms of the possible hypothesis (i.e. multivariate test outcomes) can be defined similarly. For instance, in the case of $r = 3$ tests the setup can be

$$H_0 = H_0^1 H_0^2 H_0^3 \quad \text{vs.} \quad H_1 = \{H_1^1 H_1^2 H_1^3 \text{ or } H_0^1 H_1^2 H_1^3\}. \tag{5.3}$$

Setup of the hypothesis	Coord 1	Coord 2	Coord 3
Comp 1 (H_0^1, H_0^2, H_0^3)	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$
Comp 2 $(\overline{H}_0^1, \overline{H}_0^2, \overline{H}_0^3)$	f_{21}	f_{22}	f_{23}
Comp 3 $(H_0^1, \overline{H}_0^2, \overline{H}_0^3)$	$\mathcal{N}(0,1)$	f_{32}	f_{33}

Table 5.3 – an example of 3-coordinate, 3-component mixture of model (5.3).

In this situation, each case i and associated p-value $\boldsymbol{p}_i$ can be either in component 1 with known (uniform) density, or one of the two possible situations considered as the alternative. It is then natural to fit for the probit transform a $m = 3$ component mixture model (detail in Table 5.3)

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \lambda_1 \prod_{k=1}^{r} \mathcal{N}(0,1)(x_{ik}) + \lambda_2 \prod_{k=1}^{r} f_{2k}(x_{ik}) + \lambda_3 \prod_{k=1}^{r} f_{3k}(x_{ik}), \tag{5.4}$$

where component 2 is associated to $H_1^1 H_1^2 H_1^3$ and component 3 to $H_0^1 H_1^2 H_1^3$. In this case the model induces other constraints on the component densities than one component know $(f_{1k} = \mathcal{N}(0,1), k = 1,2,3)$. Precisely here:

$$f_{31} = f_{1k} = \mathcal{N}(0,1), \quad k = 1,2,3.$$

We can also impose the constraint on the pdf of alternative hypotheses H^k in component 2 and 3, i.e.

$$f_{22} = f_{32} \quad \text{and } f_{23} = f_{33}.$$

Unfortunately, this model is not identifiable – that is, that $g_{\boldsymbol{\theta}}$ uniquely determines the parameters appearing in (5.4) – under a mild and explicit regularity condition on $g_{\boldsymbol{\theta}}$ of Allman et al. [2009], as soon as there are at least 3 variates for the mixture of the form (5.4) such that for every $k \in \{1, \cdots r\}$, the densities $\{f_{jk}\}_{1 \le j \le m}$ are linearly independent, (see section 3.3). Indeed, in the first coordinate:

$$k = 1, \quad (f_{11}, f_{21}, f_{31}) = (\mathcal{N}(0,1), f_{21}, \mathcal{N}(0,1)) \text{ are not linearly independent.}$$

On the other hand, if we relax the hypotheses in Theorem 9 from Allman et al. [2009] and compute the sum of Kruskal rank of the set $\{f_{jk}\}_{1 \le j \le m}$ for $k = 1,2,3$– say a finite set of measures has Kruskal rank κ, if κ is the maximal integer such that every κ-element subset is linearly independent. Then, for $m = 3$, straightforward modifications of the proofs establish identifiability provided the sum of the Kruskal ranks of the sets $\{f_{jk}\}_{1 \le j \le m}$ for $k = 1,2,3$ is at least $2m + 2 = 8$. Here the sum of Kruskal rank is $2 + 2 + 2 = 6 < 8$. Hence unfortunately, this relaxation of the proof does not insure identifiability of model (5.4).

The constraint $f_{31} = f_{11} = \mathcal{N}(0,1)$ of the first coordinate violates the identifiable condition for model (5.4). We can overcome this problem without changing the setup of the possible hypothesis as in (5.3) by increasing the dimension of the observed variable and use the dependence within blocks of coordinates to obtain identifiability. In detail, we

build a general mixture model with multivariate blocks (see the definition in Section 3.2):

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \lambda_1 \prod_{\ell=1}^{B} \mathcal{N}_{d_\ell}(\mathbf{0}, \mathbb{I})(x_{is_\ell}) + \sum_{j=2}^{m} \lambda_j \prod_{\ell=1}^{B} f_{j\ell}(x_{is_\ell}), \quad \sum_{j=1}^{m} \lambda_j = 1. \tag{5.5}$$

Setup of the hypothesis	Block 1	Block 2	Block 3
	Coord 1&2	Coord 3	Coord 4
Comp 1 (H_0^1, H_0^2, H_0^3)	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$
Comp 2 $(\overline{H}_0^1, \overline{H}_0^2, \overline{H}_0^3)$	f_{21}	f_{22}	f_{23}
Comp 3 $(H_0^1, \overline{H}_0^2, \overline{H}_0^3)$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & \sigma\\\sigma & 1\end{bmatrix}\right), \sigma \neq 0$	f_{32}	f_{33}

Table 5.4 – an example of 4-coordinate, 3-block, 3-component mixture of model (5.3).

Here, we fix the known component–mutivariate standard normal distribution is the first component (component null) with the proportion λ_1. We denote for each block of coordinate ℓ the subset $J_\ell \subset J = \{1, \cdots, m\}$ storing the indices of components where the alternative hypotheses H_1^ℓ is true and with equality constraint of the densities. For instance, in the setup of model (5.3) we have

$$J_1 = \{2\}, \quad J_2 = \{2, 3\}, \text{ and } J_3 = \{2, 3\}$$

The component densities where the null hypotheses H_0^ℓ is true, have the marginals $\mathcal{N}(0,1)$ and different dependence structure between component. Precisely, the densities of components having index j in $J \setminus J_\ell, \quad \ell = 1, \cdots, B$ are

$$f_{j\ell} = \begin{cases} \mathcal{N}_{d_\ell}(\mathbf{0}, \mathbb{I}) & \text{if} \quad j = 1 \\ \mathcal{N}_{d_\ell}(\mathbf{0}, V) & \text{if} \quad j \neq 1 \end{cases}$$

where $\mathbf{0} = (0, \cdots, 0) \in R^{d_\ell}$ is the d_ℓ-dimensional mean vector, $\mathbb{I} = \text{diag}(d_\ell)$ is $d_\ell \times d_\ell$ identity matrix and $V \neq \mathbb{I}$ is $d_\ell \times d_\ell$ covariance matrix with diagonal elements 1.

Similarly, the component densities where the alternative hypotheses H_1^ℓ is true– i.e. $f_{j\ell}, \ j \in J$, have the same marginal function and different variance structure (and we need at least $r = 6$ coordinates) (see Table 5.10 in section simulation study 5.5) or we can also relax the equal constraint on the pdf of False H^k (at least $r = 4$ necessary coordinates) (see Table 5.4) . Then, the set of $\{f_{j\ell}\}_{1 \leq \ell \leq B}$ is linearly independent for all $j \in \{1, \cdots, m\}$.

Model (5.5) is a special case of the multivariate mixture model that Chauveau and Hoang [2016] and Section 3.2 in the present dissertation presented. Theorem 9 in Allman et al. [2009] proved identifiability of a finite mixture of conditionally independent multivariate nonparametric measures such that for every block $\ell \in \{1, \cdots, B\}$ the densities $\{f_{j\ell}\}_{1 \leq j \leq m}$ are linear independent is whenever the number of blocks $B \geq 3$. For the mvFDR setup (5.3) the mixture model (5.5) has $B = 3$ blocks and in each block the component densities $\{f_{j\ell}\}_{1 \leq j \leq m}$ are different. Then under the condition precised in Theorem 9 of Allman et al. [2009], this model is identifiable.

5.3 The mvnpEMN01 algorithm for a multivariate nonparametric mixture with one component known

Given initial values $\boldsymbol{\theta}^{(0)} = (\boldsymbol{\lambda}^{(0)}, \boldsymbol{f}^{(0)})$, the mvnpEMN01 algorithm consists in iterating the following steps:

1. **E-step:** Calculate the posterior probabilities (conditional on the data and $\boldsymbol{\theta}^{(t)}$), for each $i = 1, \ldots, n$:

   of the component 1: $p_{i1}^{(t)} := \mathbb{P}_{\boldsymbol{\theta}^{(t)}}(Z_{i1} = 1 | \boldsymbol{x}_i) =$

   $$\frac{\lambda_1^{(t)} \prod_{\ell=1}^{B} \mathcal{N}_{d_\ell}(\mathbf{0}, \mathbb{I})(x_{is_\ell})}{\lambda_1^{(t)} \prod_{\ell=1}^{B} \mathcal{N}_{d_\ell}(\mathbf{0}, \mathbb{I})(x_{is_\ell}) + \sum_{j'=2}^{m} \lambda_j'^{(t)} \prod_{\ell=1}^{B} f_{j'\ell}^{(t)}(x_{is_\ell})},$$

   of the component $j \neq 1$: $p_{ij}^{(t)} := \mathbb{P}_{\boldsymbol{\theta}^{(t)}}(Z_{ij} = 1 | \boldsymbol{x}_i) =$

   $$\frac{\lambda_j^{(t)} \prod_{\ell=1}^{B} f_{j\ell}^{(t)}(x_{is_\ell})}{\lambda_1^{(t)} \prod_{\ell=1}^{B} \mathcal{N}_{d_\ell}(\mathbf{0}, \mathbb{I})(x_{is_\ell}) + \sum_{j'=2}^{m} \lambda_j'^{(t)} \prod_{\ell=1}^{B} f_{j'\ell}^{(t)}(x_{is_\ell})}.$$

2. **M-step for λ:**

   $$\lambda_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{(t)}.$$

3. **Nonparametric kernel density estimation step:** For any $\boldsymbol{u}$ in $\mathbb{R}^{d_\ell}$, define for each block $\ell \in \{1, \ldots, B\}$ and each component $j \in \{2, \cdots, m\}$

   $$f_{j\ell}^{(t+1)}(\boldsymbol{u}) = \frac{1}{n\lambda_j^{(t+1)}} \sum_{i=1}^{n} p_{ij}^{(t)} K_{H_{j\ell}}(\boldsymbol{u} - x_{is_\ell}),$$

   where $K_{H_{j\ell}}$ is a multivariate kernel density function, typically Gaussian, and $H_{j\ell}$ is a symmetric positive definite $d_\ell \times d_\ell$ matrix known as the bandwidth matrix.

   For the first component:

   $$f_{1\ell}^{(t+1)}(\boldsymbol{u}) = \mathcal{N}_{d_\ell}(\mathbf{0}, \mathbb{I})(\boldsymbol{u}), \quad \forall \ell = 1, \cdots, B.$$

5.4 Comparing univariate and multivariate FDR

A question is how to compare univariate FDR (univFDR) and multivariate FDR (mvFDR) controls? Before going to answer this question, we want to underline that in any way, it is not possible to achieve a fair comparison since the multivariate $\boldsymbol{p}$-value bring an additional information. The purpose is that we try to obtain some sort of comparisons but obviously the information from each situation is not comparable.

We first tried some simple rules as applying univariate FDR control to the max of the coordinates in $\boldsymbol{p}_i$. The idea is that, if the r tests are similar and the only two possible

situations are TTT or FFF, then univariate FDR will perform equivalently as if only one test was observed, whereas the max is more conservative. A problem with this strategy is that under the null, the max (or min or other rules) are no longer $\mathcal{U}_{[0,1]}$, so that usual FDR controlling procedures fail.

In the univariate case, the procedure for plotting the so-called "local FDR" control is to sort the p-values, and compute

$$i \mapsto \ell FDR(i) = \frac{1}{i}\sum_{l=1}^{i} \hat{p}_{l1},$$

where $\hat{p}_{l1}$ is the (estimated) posterior $P(H_0|p_i)$ that the ith smallest p-value corresponds to a non-significant case (H_0). The decision rule consists then is rejecting the i smallest p-values, such that $\ell FDR(i) \leq \alpha$. In practice, it is used to define how many of the smallest observed p-values have to be rejected, in order to achieve an estimated error level smaller than α. This number of rejections can be defined by the index

$$\widehat{d}_\alpha := \max\{i \in \{1, \cdots, n\} : \ell FDR(i) \leq \alpha\}$$

This index corresponds to the largest ordered p-value before which the estimated FDR crosses the level α for the "last time".

In mutivariate setup it is not possible to order the multivariate $\boldsymbol{p}_i$'s. Some alternatives are possible:

- Order the rows according to the max (or the min, the average, etc) of the $\boldsymbol{p} = (p_{i1}, \cdots, p_{ir})$. But these strategies are not proper way of building a substitute for multivariate FDR. Actually, they are wrong theoretically since for the null cases, the max or the min or any such transformation of the distribution $\mathcal{U}_{[0,1]}^{\otimes r}$ destroys the uniform distribution under the null property. Since `fdrtool` or other local FDR procedures are grounded on the uniform for the null cases, these are theoretically not applicable.

- Order according to the posteriors obtained by the multivariate algorithm. For our algorithm, we use the order of the final values of the p_{i1}'s under the component H_0, i.e. the posterior probabilities after convergence of the algorithm that we denote $\hat{p}_{i1}$'s.

- Other rejection rules as in Chi et al. [2008]'s:
  – "by product": reject a null if $\prod_{k=1}^{r} p_k$ small;
  – "by max": reject H_i if $\max_k T_{ik}$ is large enough where $\boldsymbol{T}_i = (T_{i1}, \cdots, T_{ir})$ are the test statistics;
  – "by sum": reject H_i if $\sum_k c_k T_{ik}$ is large enough where $c_k > 0$ are some positive constants.

  We have not used these rules since they requires constructing p-values via maximization under linear constraints imposed by data's empirical distribution which lead to the case that the rejection associated to the largest p-value is retained or "conservative".

- Use the r results of the r univariate local FDR's, (Positive/Negative for each case) to decide wether the global H_0 or H_1 is true.

Comparison then requires some caution, since the sorting order depends on the vector of p-values used. The number of rejected cases at level α do not correspond to the same cases, depending on which input has been used (univariate p_k's, max of $\boldsymbol{p}$ per rows, order from posteriors, something else, ...). Hence, in a typical FDR plot where the x axis in the index $i = 1, \cdots, n$, the number of rejected cases P (Positives) do not correspond to the *same* cases from the experiment, but to the P cases associated to the smallest p-values in each input vector used to build the FDR.

Observing the complete data $(\boldsymbol{p}, \boldsymbol{Z})$, we can compute the true FDR by

$$i \mapsto \ell FDR_{\text{true}}(i) = \frac{1}{i} \sum_{l=1}^{i} \mathbb{I}_{z(l)1} = 1,$$

where the $z(l)$'s is the lth component membership corresponding to the lth smallest p-value in the ordering of the n cases. The ordering of $\boldsymbol{z}$ vector changes the true FDR of the "smallest" rejected cases.

However, the percentage of wrongly rejected cases, the False Positive (FP) divided by the total number of cases declared Positive (P), if available, is still meaningful for comparisons among several approaches (FP/P when P > 0 is the definition of the FDR from Benjamini and Hochberg [1995]). Similarly, the percentage of False Negative (FN/N) is meaningful to compare the power of the various FDR control methods: we want FN/N as small as possible. Indeed, the four possible outcomes as defined in table 5.2, can be computed on simulated data, from the knowledge of the true component membership $\boldsymbol{z}$. The toy example below on simulated data illustrates this.

The error on the target level α which is supposed to be achieved by an FDR control procedure may also be evaluated in simulation. In our Monte-Carlo experiments, we ran S replications of n such tests. From which we can evaluate the actual error $\ell FDR(\widehat{d}_\alpha^{(s)})$ when the $\widehat{d}_\alpha$ smallest p-values are rejected at replication s, so that

$$\Delta(\alpha) = \frac{1}{S} \sum_{s=1}^{S} \left(\ell FDR(\widehat{d}_\alpha^{(s)}) - \alpha \right)^2$$

can be viewed as a MSE on the target level α over all the replications.

5.5 Simulation study

5.5.1 Simple simulated examples

Model 1 Here is a toy example to illustrate the behavior of our novel approach and the possible comparisons between FDR control methods. The $r = 3$ trivariate data correspond to the simplest model, i.e. $H_0 \equiv TTT$ and $H_1 \equiv FFF$. This is a 2-component mixture with the proportion of true H_0 set to $\lambda_1 = 0.6$. The $n = 1000$ cases are simulated directly

	H_0 True		H_0 False			
	not rejected	rejected	not rejected	rejected	FDR	FN/N
$k=1$	569	9	228	194	0.044	0.286
$k=2$	567	11	227	195	0.053	0.286
$k=3$	569	9	198	224	0.038	0.258
mvFDR	530	48	8	414	0.104	0.015
Sum	578		422			

Table 5.5 – FDR and FN/N using `fdrtool` for each coordinate $k = 1, 2, 3$ in a single sample of $n = 1000$ tests for Model 1, comparing with mvFDR strategy.

at the level of the probit transform, i.e. $\mathcal{N}(0,1)^{\otimes 3}$ for the null, and some distribution located on negative values for the significant cases (H_1), actually here simply $\mathcal{N}(-2,1)^{\otimes 3}$. Then the p-values are obtained by reversing the probit transform, i.e. by applying the normal cdf to the $\boldsymbol{x}$. These are denoted $p_{ik}, i = 1, \cdots, n$ and $k = 1, \cdots, r$ as usual.

Applying the mvFDR algorithm for this model, sorting the n cases by the posteriors and computing $\ell FDR(i)$ returns the plot of the mvFDR control in Fig 5.2 (A). Then computing the ℓFDR using `fdrtool` for each coordinate k, ie each vector of p-values $(p_{1k}, \cdots, p_{nk})$ returns r ways of controlling the FDR. If we want to superimpose these on the same plot, we have to sort each kth coordinate separately, and to plot $i \mapsto FDR(i)$ if we want to display the increasing $FDR(.)$ curves, since each ordering is different. As said above, the plot is then misleading since the P rejected cases are all different. We finally can do that as well for the min and the max of the rows $(p_{i1}, \ldots, p_{ir})$ (remind that the max/min both destroy the uniform distribution of the p-values under the null, so we just use them for illustration). Fig 5.2 shows however that the mvFDR is very accurate here (since the true FDR can be computed from the $\boldsymbol{z}$ for the ordering used in the figure), that the univariate FDR's are more conservative, that the max is way too conservative (conservative in the sense that the rejection associated to the largest p-value is retained) and the min rejects too many cases.

If we want the ordering to be meaningful for each method we can split the plots, as in Fig 5.3, which displays the mvFDR control result, and each of the $r = 3$ univariate FDR controls based on the Benjamini and Hochberg [1995] procedure, and the two fdrtool controlling methods (the one that corresponds to the ℓFDR is the local fdr).

Comparing from the FP and the FN rates. These behavior can then be precised as said above, by computing the exact $FDR = FP/P$ and the false negative rate FN/N as in Table 5.2. For this sample dataset we found, each univariate local FDR control per coordinate at level $\alpha = 10\%$ in $S = 1$ replication as in Table 5.5.

We can see that the three univariate FDR controls are conservative, with actual locFDR $< \alpha$, and more importantly, they miss about 28% of the interesting cases. The same statistics but based on the tail fdr return FDR closer to the target α (between 8% and 10%), but FN/N still large, about 17%. These results have to be compared with the same output for the mvFDR: $FDR = 10.4\%$ and $FN/N = 0.015$.

As already noticed in the plots, the FDR is close to the target level α, and the False

	Coord 1	Coord 2	Coord 3	mvFDR
FDR	0.056	0.056	0.057	0.111
FN/N	0.266	0.267	0.267	0.015
$\Delta(\alpha)$	0.00224	0.00225	0.00220	0.00027

Table 5.6 – Average of FDR, FN/N using `fdrtool` and MSE on the target level α over all the replications for each coordinate $k = 1, 2, 3$ when do $S = 300$ replications of $n = 1000$ tests, for Model 1, comparing with mvFDR strategy.

Negative rate of 1.5% is much smaller than for both univariate FDR strategies: the mvFDR is definitely better. Similar conclusions are drawn when doing Monte Carlo simulation for $S = 300$ replications (the results are as in Table 5.6).

It may seem surprising that the true FDR based on the same vector $\boldsymbol{z}$ looks different in all cases. This is precisely the effect of the ordering of the p-values.

Model 2 Of course, Model 1 above is very simplistic, with the r multivariate tests completely similar (p-values comparable): we know this is the situation for which the multivariate setup helps to recover the mixture better than from univariate's (the blessing of dimensionality). We build another example to see the behaviors of univariate and multivariate FDR controls.

We modified Model 1 by changing the location of the probit transform's densities, i.e. replacing $\mathcal{N}(-2,1)^{\otimes 3}$ by $\mathcal{N}(-1.5,1)\mathcal{N}(-2,1)\mathcal{N}(-3,1)$. The proportion estimates of the 2-component mixture are $\widehat{\lambda}_1 = 0.566, \widehat{\lambda}_2 = 0.434$ and the densities estimates are plotted as in figure 5.1, when we apply `mvnpEMN01` algorithm on a single sample. Fig. 5.1 shows that this constrained model recovered properly component 2 as "FFF", even if this is not specified in the model.

The FP and FN rates for Model 2 are precised in Table 5.7. We can see that, the univariate FDR controls are conservative and the interesting cases they miss are different for each coordinate (decreasing from 35.5%, 23.8% to 4% when the mixture is more and more separated). The mvFDR controls is definitely better with $FDR = 10.9\%$ and the False Negative rate of 0.4%. Similarly, several FDR controls are plotted in Fig 5.2 (B) and Fig 5.3 (B).

	FDR	FN/N
$k = 1$	0.072	0.355
$k = 2$	0.052	0.238
$k = 3$	0.067	0.040
mvFDR	0.109	0.004

Table 5.7 – FDR and FN/N using `fdrtool` for each coordinate $k = 1, 2, 3$ in a single sample of $n = 1000$ tests for Model 2, comparing with mvFDR strategy.

We also made Monte Carlo experiment of $S = 300$ replications for Model 2 and got

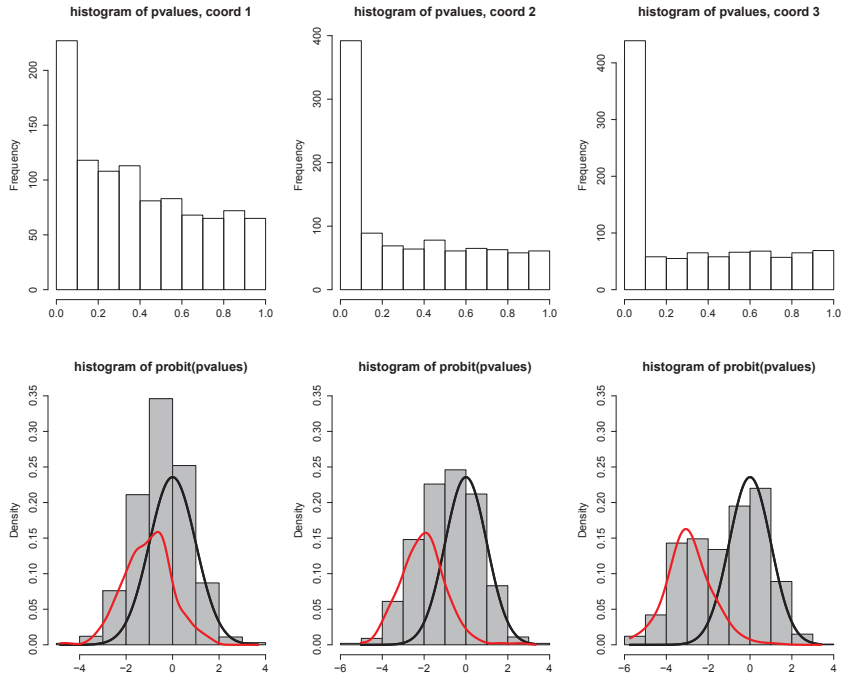


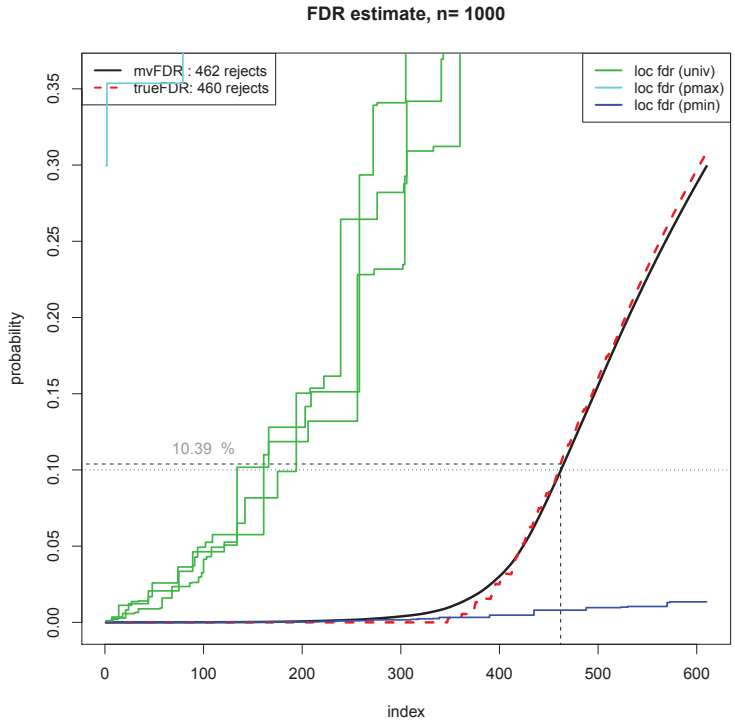
Figure 5.1 – The histogram of p-values (top) and the densities estimates of the probit transform of p-values (bottom) given by the `mvnpEMN01` algorithm, per coordinate, of Model 2.

the results as in Table 5.8.

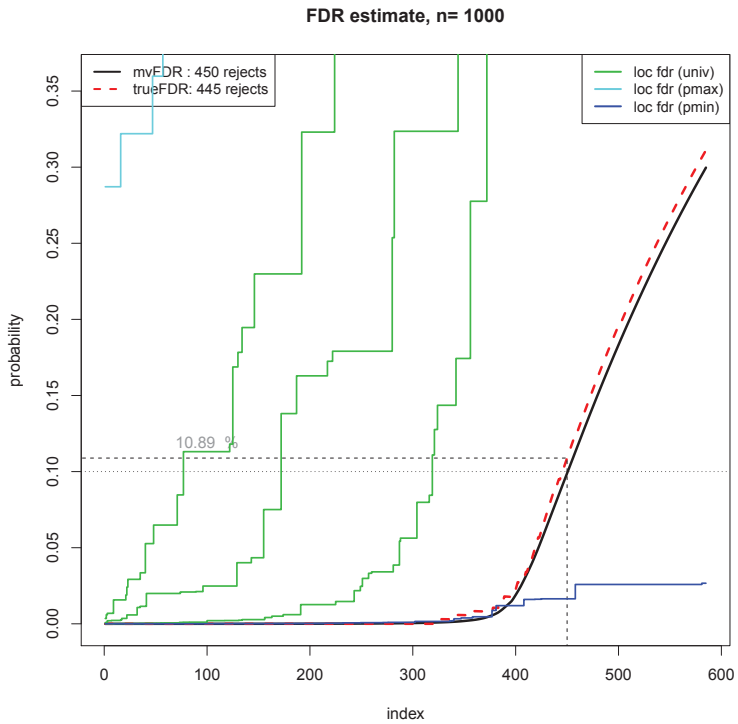
Model 2	Coord 1	Coord 2	Coord 3	mvFDR
FDR	0.216	0.056	0.087	0.111
FN/N	0.374	0.266	0.052	0.005
$\Delta(\alpha)$	0.14290	0.00223	0.00041	0.00025

Table 5.8 – Average of FDR, FN/N using `fdrtool` and MSE on the target level α over all the replications for each coordinate $k = 1, 2, 3$ when do $S = 300$ replications of $n = 1000$ tests for Model 2, comparing with mvFDR strategy.

When the mixture is more and more overlapping (from coordinate 3 to coordinate 1), the univariate FDR control is worst and worst: coordinate 3 has the best FDR control value (0.087) and the false negative rate is smallest (5.2 %), comparing with two other coordinates. For both of two strategies: univ- and multi- FDR control, the mvFDR is definitely better.

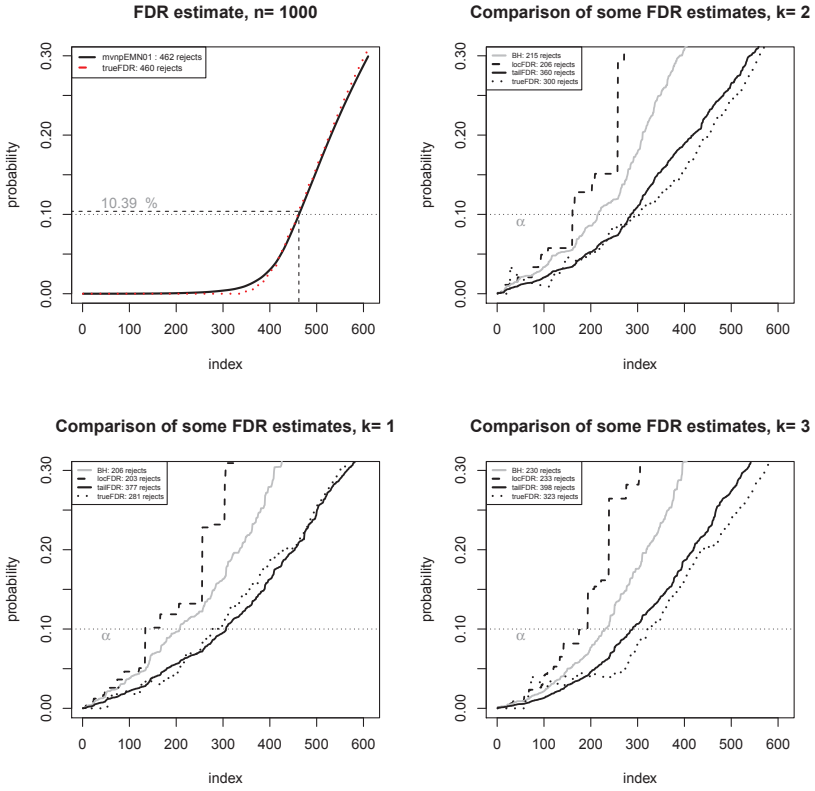


(A)

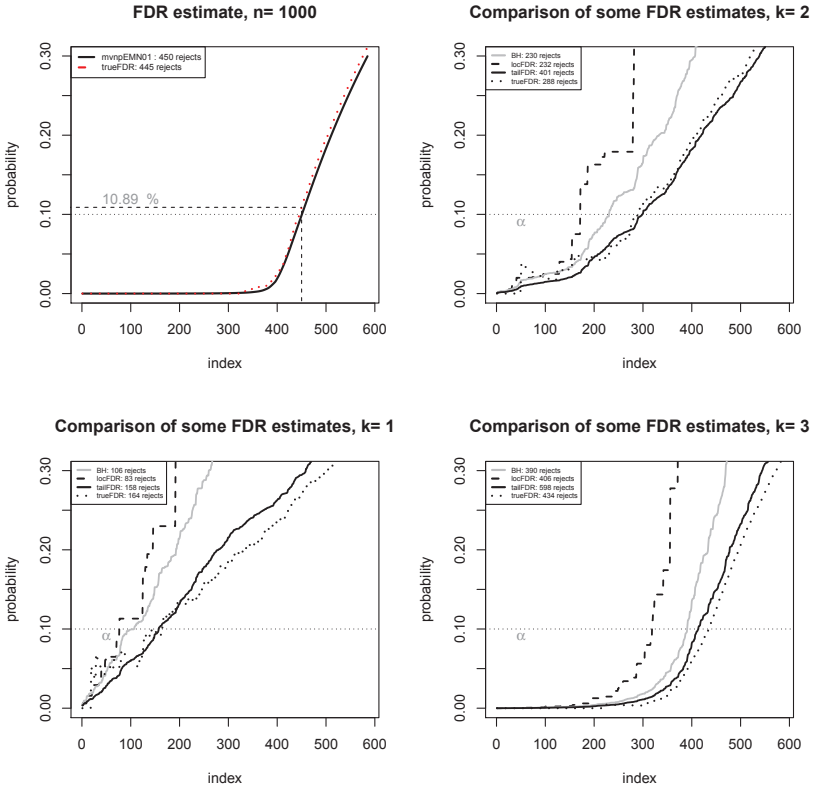


(B)

Figure 5.2 – Example from Gaussian data, several FDR plots in one figure: true FDR, mvFDR, each of $r = 3$ univFDR controls using `fdrtools`, as well the univFDR controls based on the max/min of the rows $(p_{i1}, \cdots, p_{ir})$; Model 1 (A) and Model 2 (B).



(A)



(B)

Figure 5.3 – Example from Gaussian data, several FDR plots separately; topleft: mvFDR and true FDR, the other three panels correspond to the $k = 1, 2, 3$ coordinates for the multivariate p-values. In each plot, the ordering of the n cases is different; Model 1 (A) and Model 2 (B).

MODEL 3	Block 1	Block 2	Block 3
$Probit(\boldsymbol{p})$: $\boldsymbol{y}$	Coord 1&2	Coord 3	Coord 4
Comp 1 (H_0^1, H_0^2, H_0^3)	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$
Comp 2 $(\overline{H}_0^1, \overline{H}_0^2, \overline{H}_0^3)$	$\mathcal{N}_2\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$	$\mathcal{N}(-2,1)$	$\mathcal{N}(-2,1)$
Comp 3 $(H_0^1, \overline{H}_0^2, \overline{H}_0^3)$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$	$\mathcal{N}(-3,1)$	$\mathcal{N}(-3,1)$

Table 5.9 – Model 3: a sample of 4-coordinate, 3-block, 3-component mixture with $\lambda_1 = 20\%$ component 1 (under Gaussian distribution), $\lambda_2 = 35\%, \lambda_3 = 45\%$ respectively for component 2 and 3 (under H_1).

5.5.2 3-component simulated examples

Remind here a complex model in term of the possible hypothesis in case of $r = 3$ tests as setup in model 5.3, i.e. $H_0 \equiv TTT$ and $H_1 \equiv FFF + TFF$. It can be fit with a $m = 3$-component mixture model, where component 1 with known density, component 2 is associated to "FFF", and component 3 to "TFF". The possible constraints on the pdf of the null and the alternative are

$$f_{11} = f_{12} = f_{13} = f_{31},$$

$$f_{22} = f_{32} \text{ and } f_{23} = f_{33}.$$

This model is not identifiable under the condition of Allman et al. [2009] if we have not extended the dimension of coordinates and use the within block of coordinate dependence. Thanks to the nonparametric mixture model with multivariate block which allows the difference in covariance structures whereas preserving the marginals and gives identifiability (see detailed analysis in Section 5.2.1)

We present in this section two examples of extended model depend on considering or not the equal constraints on the pdf of the alternative hypotheses: an extension to $r = 6$-coordinate, $B = 3$-bivariate block mixture model (Model 3) and another one with $B = 3$ blocks (one bivariate and two univariate), $r = 4$ coordinates (Model 4). The $n = 1000$ cases are simulated at the level of the probit transform, i.e. multivariate standard normal distribution for the null, and for the significant cases (H_1) as detail in the Table 5.9 (Model 3) and Table 5.10 (Model 4). Both Model 3 and Model 4 are identifiable because of the different covariances. The difference between them is that we drop or not the constraints on the pdf of the significant cases.

The proportion estimates of 3 components are $\widehat{\lambda}_1 = 0.221, \widehat{\lambda}_2 = 0.344, \widehat{\lambda}_3 = 0.435$ and the densities estimates are plotted as in figure 5.4, for Model 3.

We illustrate the possible comparisons between FDR control methods as we did for the simple examples in previous Section. The behaviors of several FDR controls in Fig 5.5 and Fig 5.6 together with the False negative rate of 0.05% (under mvFDR controls)– which is smaller than other univariate methods (see Table 5.11), precise that the mvFDR control is appropriate and very accurate.

MODEL 4	Block 1	Block 2	Block 3
$Probit(\boldsymbol{p})$: $\boldsymbol{y}$	Coord 1&2	Coord 3&4	Coord 5&6
Comp 1 (H_0^1, H_0^2, H_0^3)	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$
Comp 2 $(\overline{H}_0^1, \overline{H}_0^2, \overline{H}_0^3)$	$\mathcal{N}_2\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$	$\mathcal{N}_2\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$	$\mathcal{N}_2\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)$
Comp 3 $(H_0^1, \overline{H}_0^2, \overline{H}_0^3)$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$	$\mathcal{N}_2\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$	$\mathcal{N}_2\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$

Table 5.10 – Model 4: a sample of 6-coordinate, 3-block, 3-component mixture with $\lambda_1 = 20\%$ component 1 (under Gaussian distribution), $\lambda_2 = 35\%, \lambda_3 = 45\%$ respectively for component 2 and 3 (under H_1).

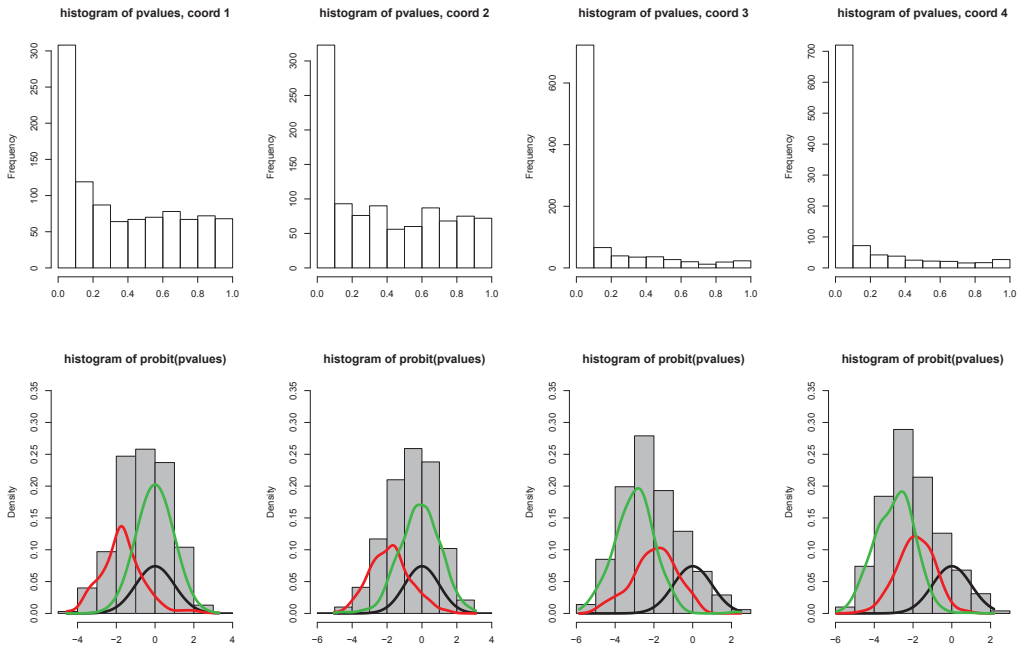


Figure 5.4 – The histogram of p-values (top) and the densities estimates of the probit transform of p-values (bottom) given by the `mvnpEMN01` algorithm, per coordinate, for Model 3.

Model 2	Coord 1	Coord 2	Coord 3	Coord 4	mvFDR
FDR	0.112	0.093	0.082	0.082	0.100
FN/N	0.767	0.767	0.140	0.141	0.00048
$\Delta(\alpha)$	0.07060	0.06624	0.00051	0.00053	0.00022

Table 5.11 – Average of FDR, FN/N using `fdrtool` and MSE on the target level α over all the replications for each coordinate $k = 1, 2, 3, 4$ when do $S = 300$ replications of $n = 1000$ tests for Model 3, comparing with mvFDR strategy.

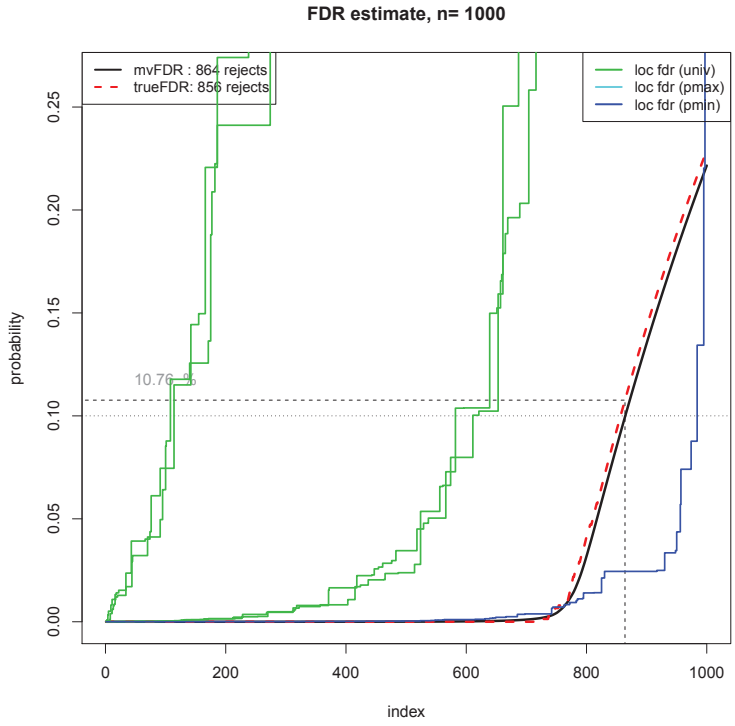


Figure 5.5 – Several FDR plots in one figure for Model 3.

Of course, the comparison between uniFDR's and the mvFDR is unfair since the latter uses all the available information from the 4-dimensional data.

The false negative rate in coordinate 1 and 2 are more than 5 times of coordinate 3 and 4 since the mixture is more overlapping within coordinate 1 and 2. We can also see this result in Fig 5.6.

We made the same computation as in Model 3. Table 5.12 indicates that mvFDR strategy is rather good in all value of FDR, FN/N, $\Delta(\alpha)$ whereas univFDR strategies cannot, even that Model 4 is more overlapping than Model 3.

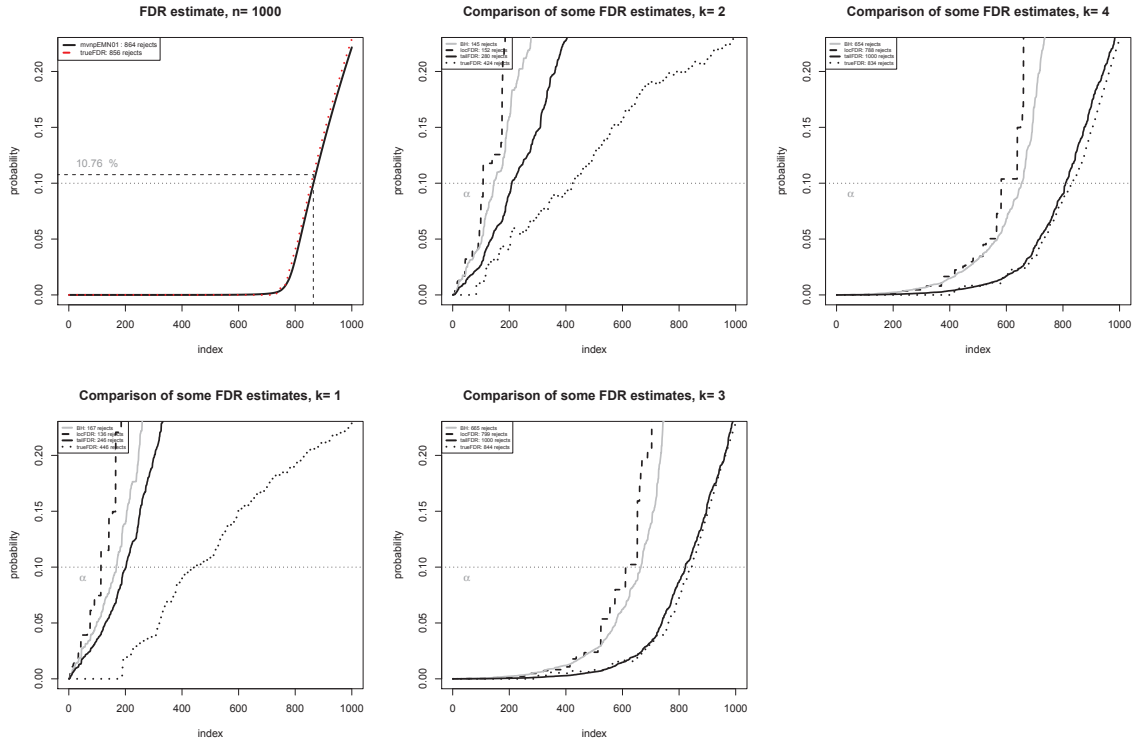


Figure 5.6 – Several FDR plots separately for Model 3; topleft: true FDR and mvFDR, the other four panels correspond to the $k = 1, 2, 3, 4$ coordinates for the multivariate p-values. In each plot, the ordering of the n cases is different.

Model 2	Coord 1	Coord 2	Coord 3	Coord 4	Coord 5	Coord 6	mvFDR
FDR	0.081	0.097	0.067	0.067	0.066	0.067	0.115
FN/N	0.768	0.768	0.318	0.319	0.322	0.322	0.014
$\Delta(\alpha)$	0.05544	0.06796	0.00130	0.00131	0.00134	0.00127	0.00041

Table 5.12 – Average of FDR, FN/N using `fdrtool` and MSE on the target level α over all the replications for each coordinate $k = 1, \ldots, 6$ when do $S = 300$ replications of $n = 1000$ tests for Model 4, comparing with mvFDR strategy.

5.6 Real data example

5.6.1 Real data from large scale micro-array experiments

We propose to illustrate the behavior of our approach using parts of a large dataset from a "Maize Methylation Project" experimented by the group LIMAGRAIN and *Laboratoire de Biologie des Ligneux et des Grandes Cultures (LBLGC) Université d'Orléans*, EA 1207 (agreement 396N). This project is based on microarray experiments involving several hybrids and parental lines of maize. The ultimate purpose of this project is to explain modifications involved in the hybridization process. The base dataset consists in more than 2 millions of responses called "spots" hereafter, with a hundred variables recorded for each spot. From this dataset we only retain $T = 21$ quantitative measures that are log-ratio's between two signals (red and green) from the microarrays, related to T different hybrids and parental lines, that are denoted "treatments" in the sequel. These T treatments can be viewed as multivariate measures on the same "individuals", where the individuals here are the locations along the genome sequence (also called spots).

Briefly, the statistical setup proposed by the biologists consists in the following steps: for each scaffold of the genome,

1. Define a reference value μ_0^t for each treatment t (from prior data or a reference sample);

2. Define a window size, and build samples of spots belonging to each window (consecutive along the sequence). This determines for each scaffold s a certain number n_s of windows, and sample sizes in each window that vary (because of the micro-array setup).

3. For each treatment $t \in \{1, \cdots, T\}$ and window $w = 1, \cdots, n_s$, test the null hypothesis $H_0^{t,w} : \mu^{t,w} = \mu_0^t$ i.e., that the mean of the distribution of the spots in the window w is equal to the reference. Stated like this, we think of applying a standard parametric Student t-test. However, since the sample sizes within each window are often too small (less than 30), and the underlying normality assumption is often violated, a nonparametric version, namely a Wilcoxon signed rank test of a null hypothesis related to localization, i.e. that the distribution of the sample from each window is symmetric about μ_0^t, is used instead.

4. For each treatment, control the FDR of the n_s multiple tests obtained, using a standard univariate FDR procedure like `fdrtool`.

The results from the univariate FDR perspective are summarized in Fig 5.7. It shows in particular that the percentage of rejected cases is rather stable for all the 10 Scaffolds, whereas it clearly depends on the applied treatment. For instance, treatment 3 shows an average 40% of rejected cases, whereas treatment 20 is limited to about 10% of rejections. We do not discuss this in more details here.

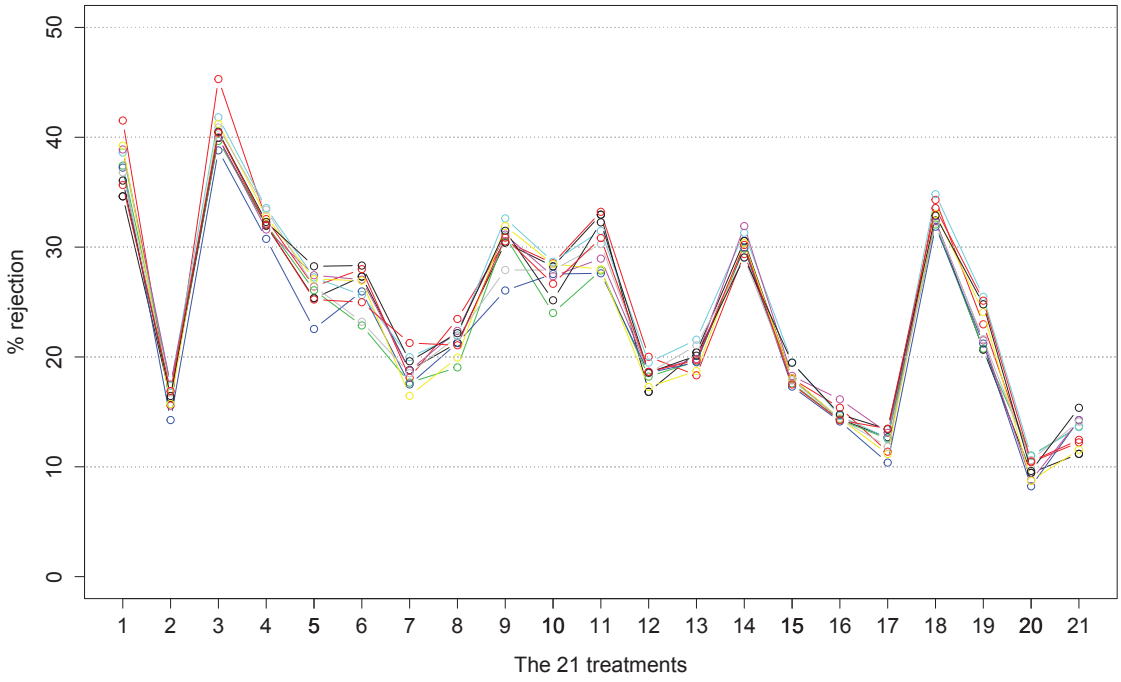


Figure 5.7 – Summarized results as percentages of rejection for each univariate FDR control. Each color line here is only to connect all the treatments for each of 10 scaffolds.

5.6.2 mvFDR for multivariate p-values:

Each test above returns a p-value $p_{w,t}$ for window w, treatment t, this being also for each scaffold. Hence we ultimately have for each scaffold s, n_s T-dimensional p-values as in our multivariate FDR general setup,

$$\boldsymbol{p}_w = (p_{w,1}, \ldots, p_{w,T}), \quad w = 1, \ldots, n_s.$$

The number of windows per scaffold (the n_s's) vary here between 2800 and 5800 with $n = \sum_{s=1}^{S} = 39,141$. That sample size is largely enough to apply nonparametric multivariate mixture estimates using our approach, even if multivariate blocks are required to allow for some dependence.

Our purpose is not to obtain a scientific answer to the ultimate goal of the project, but merely to select some r-dimensional subsets of p-values from these actual multiple tests, upon which a multivariate FDR control can be experimented. We consider here the simplest model with $m = 2$ components, and the control of the FDR for the null hypothesis denoted "T... T" (r times) above, corresponding to the simultaneous non significant answers to the r individual null hypotheses, associated to component $j = 1$. Component $j = 2$ is associated to a nonparametric multivariate distribution which is actually completely unconstrained in the model, except for the conditional independence of blocks (or coordinates) design. Hence, even if we designate component 2 in the initial presentation by the simultaneous rejection of the r univariate hypotheses, any combination might be obtained by the algorithm, in a completely data-driven way.

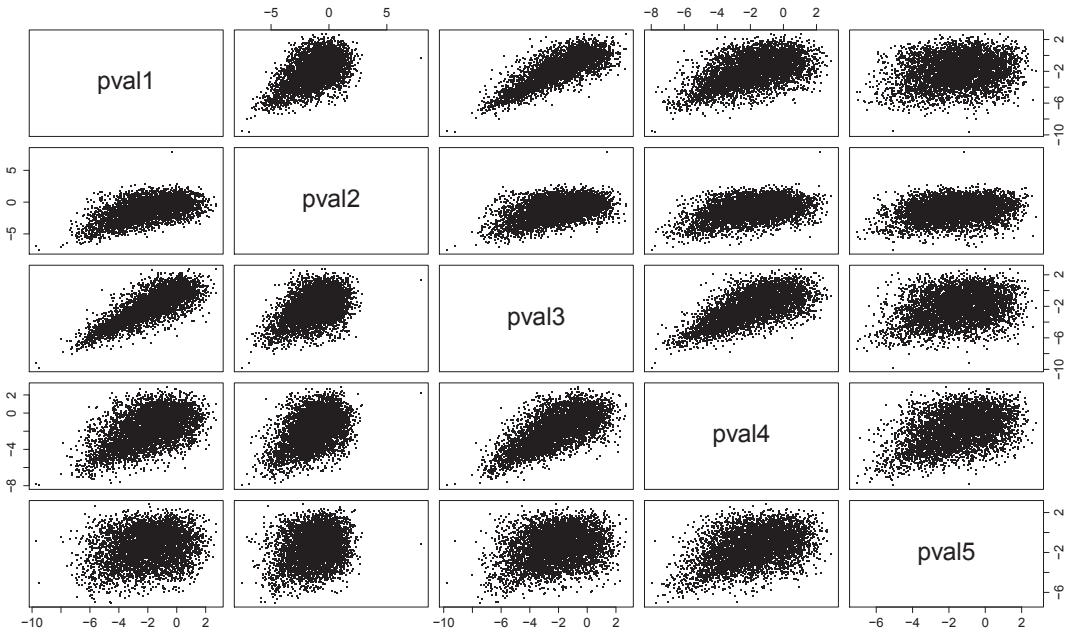


Figure 5.8 – Pairplots of probit transforms from maize data, Scaffold 1, first 5 treatments.

We proceed as follows: (i) apply a probit transform to all the p-values; (ii) remove from the datasets the rows (individual multivariate observations) for which at least one p-values equals 1, since these results correspond to wrong numerical approximations from the test procedures that result in undefined probit transforms (equal to ∞). This procedure discards only about 9% of the data, the final size for Scaffold 1 is $n_1 = 5219$. Of course a more precise way of handling these numerical difficulties could be developed.

An interesting feature of these (probit transform of) p-values is that the conditional independence assumption of coordinates is almost satisfied, except for some couple of variables, as revealed by, e.g., the pairplots of the $r = 5$ first p-values in Fig. 5.8, where a grouping of treatments (1, 3) in a block and keeping the 3 other coordinates as univariate blocks make sense.

We then try the $m = 2$ components simple model with these $r = 5$ coordinates. A typical result for blocks defined as (1, 2, 1, 3, 4) is given in Fig. 5.9. The mvFDR control rejects here the 4170 cases ordered by smallest posterior probability of belonging to component 1, as defined previously.

We use a generic plot to display the marginals of both `mvnpEMN01` and `mvnpEM` algorithm, i.e. the marginals are plotted as wKDE's. Even we know that the theoretical shape for component 1 in the model is Gaussian, we keep that to illustrate the difference between this constrained model and the generic full model. The wKDE's of component 1 densities are "Gaussian-looking", which shows that the algorithm is forced to use $\mathcal{N}(0, 1)$'s as densities, that helps the algorithm retaining the cases corresponding to H_0 : "$TTTTT$". This implies a small component weight $\widehat{\lambda}_1 \approx 21\%$ (`mvnpEMN01` solution). Then component 2 uses the nonparametric flexibility to describe the distribution of all the other cases, that correspond to negative localization and more or less to H_1 : "$FFFFF$", even if this was

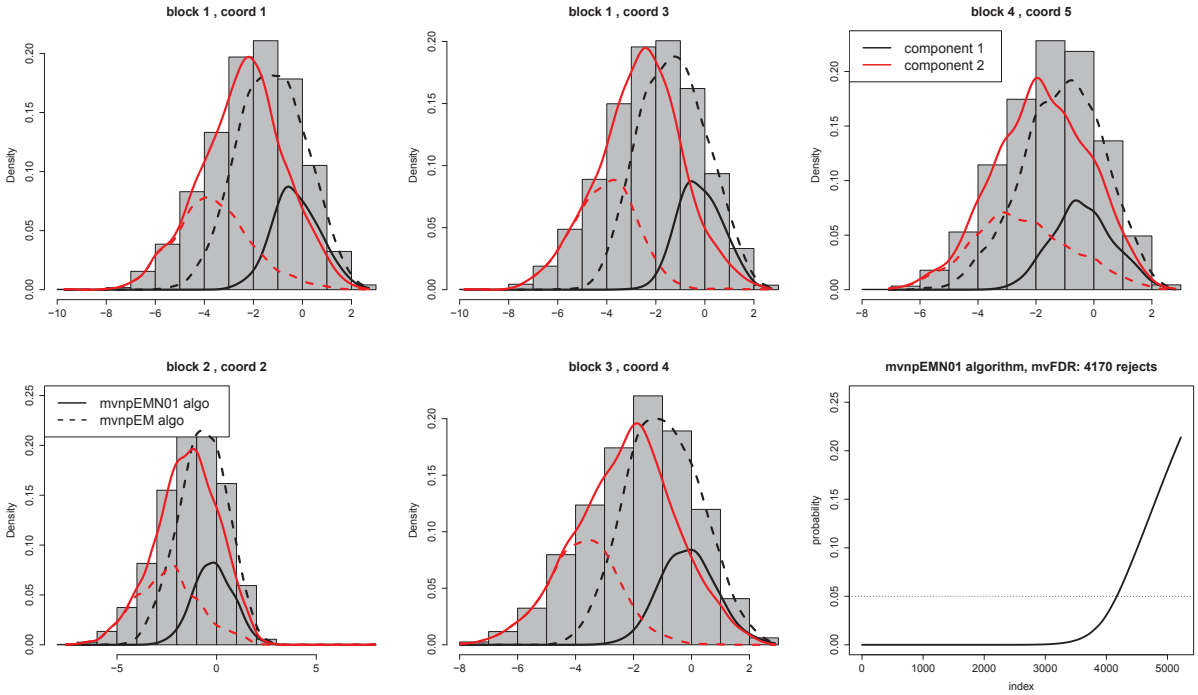


Figure 5.9 – Example from maize data, Scaffold 1, $r = 5$ first probit transforms of p-values with block design (1, 2, 1, 3, 4). Solid line: `mvnpEMN01` solution, dash line: `mvnpEM` solution. First 5 panels: marginal plots; bottom-right panel: the mvFDR control using `mvnpEMN01` algorithm.

not forced in the model. When using the unconstrained model, i.e. the plain `mvnpEM` algorithm, the component labeled "1" (black) corresponds to the largest $\widehat{\lambda}_1 \approx 70\%$, but to a more blurred hypothesis, with pdf estimates definitely not Gaussian-looking: their modes are around -1. So the hypothesis associated to $j = 1$ is not simply H_0 : "$TTTTT$"; even if component 2 is H_1 : "$FFFFF$" where the pdf estimates have negative location (around -4).

The percentage of rejection (about 80%, `mvnpEMN01` solution) is higher for the H_0 : "$TTTTT$" mvFDR control, and in particular higher than the individual percentages given in Fig. 5.7 for the 5 first treatments. This is expected since the global H_0 here involves 5 simultaneous individual hypotheses. It is interesting to see that the known component strategy is really "doing the job" in this real data case, since without it, component 1 is about 70% of the cases and is encompassing much more general cases, whereas component 2 (and mvFDR% of rejection) is smaller. This is because component 1 is not associated to H_0 : "$TTTTT$" here. That's why we have not presented the mvFDR control using the `mvnpEM` algorithm in Fig. 5.9. The two solutions are completely different. This actual data example illustrates the importance of using the constrained algorithm if the model implies that H_0 : "$TTTTT$" is meaningful.

We made an experiment on other coordinates, the last treatments (17, 20, 21) that are associated to small percentages of univariate rejection (see Fig. 5.7). The mvFDR control rejects 2344 cases (`mvnpEMN01` solution, Fig. 5.10). The global mvFDR null "TTT" retains 44,5%, higher than the individual percentage given in Fig. 5.7. In `mvnpEM` solution,

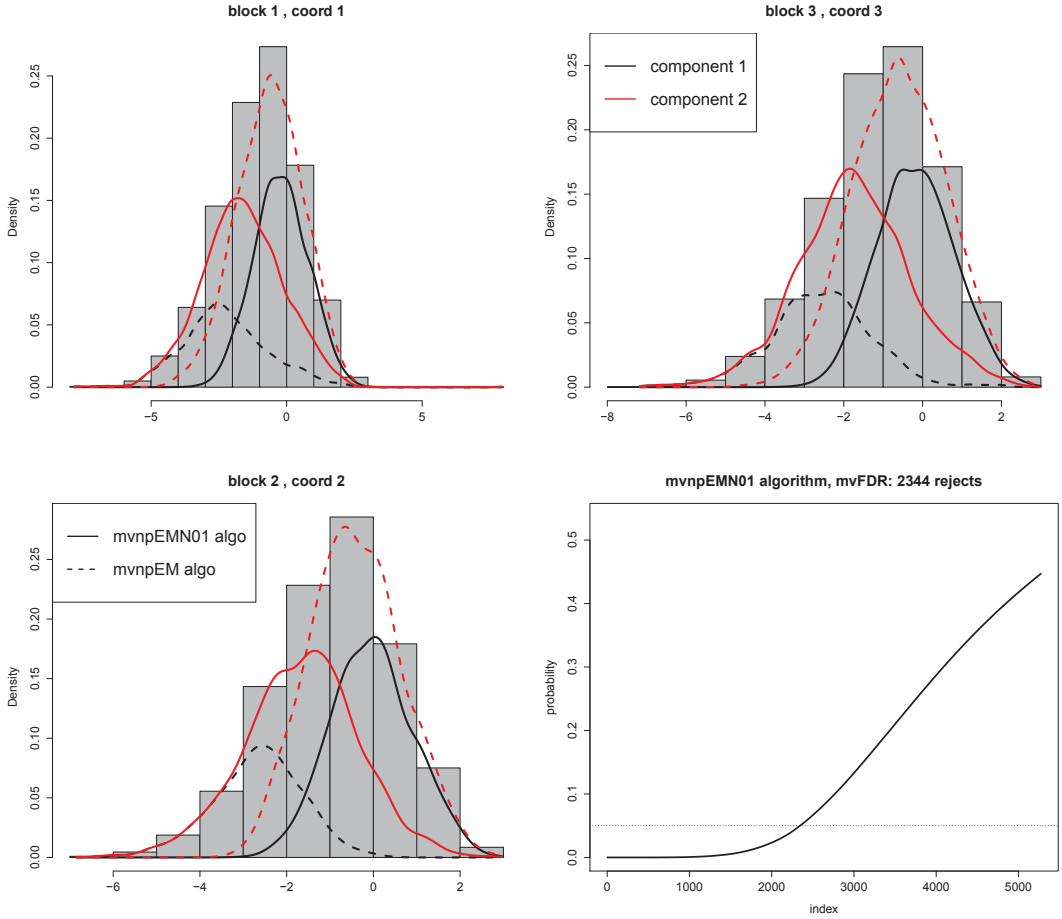


Figure 5.10 – Example from maize data, Scaffold 1, $r = 3$ probit transforms of p-values corresponding to coordinate $(17, 20, 21)$ with block design (1, 2, 3). Solid line `mvnpEMN01` solution, dash line:: `mvnpEM` solution. First 3 panels: marginal plots; bottom-right panel: the mvFDR control using `mvnpEMN01` algorithm.

two components locate around -1 and -3 pdf estimates. Hence the imposed constraints forces the algorithm to identify component $j = 1$ to H_0 : "TTT".

We tried other model on rich data with $B = 3$ bivariate blocks designed as (1, 2, 1, 3, 4, 5, 5, 6, 7, 7) for 10 coordinates corresponding to the treatments 1 to 6 and 10 to 13. The most obvious blocks showing a dependence apart from a possible mixture model are the pairs of p-values: (p_1, p_3), (p_6, p_{10}) and (p_{12}, p_{13}) (see pairplots in Fig. 5.11). The proportion estimate of component 1 using the posterior probability after convergence of `mvnpEMN01` algorithm is 22.26%. The marginal density functions are plotted in Fig. 5.12. The marginal `wKDE` solutions from the constrained model and algorithm show normal-looking densities for component 1, hence this component is associated as expected to the simultaneously non significant 10 univariate hypotheses. The results of mvFDR control are presented in Fig. 5.13. Our constraint model works as well on a rich real dataset, and illustrates the potential of our approach in controlling a multivariate FDR against a simple global null hypothesis ("$TT \dots T$", r times).

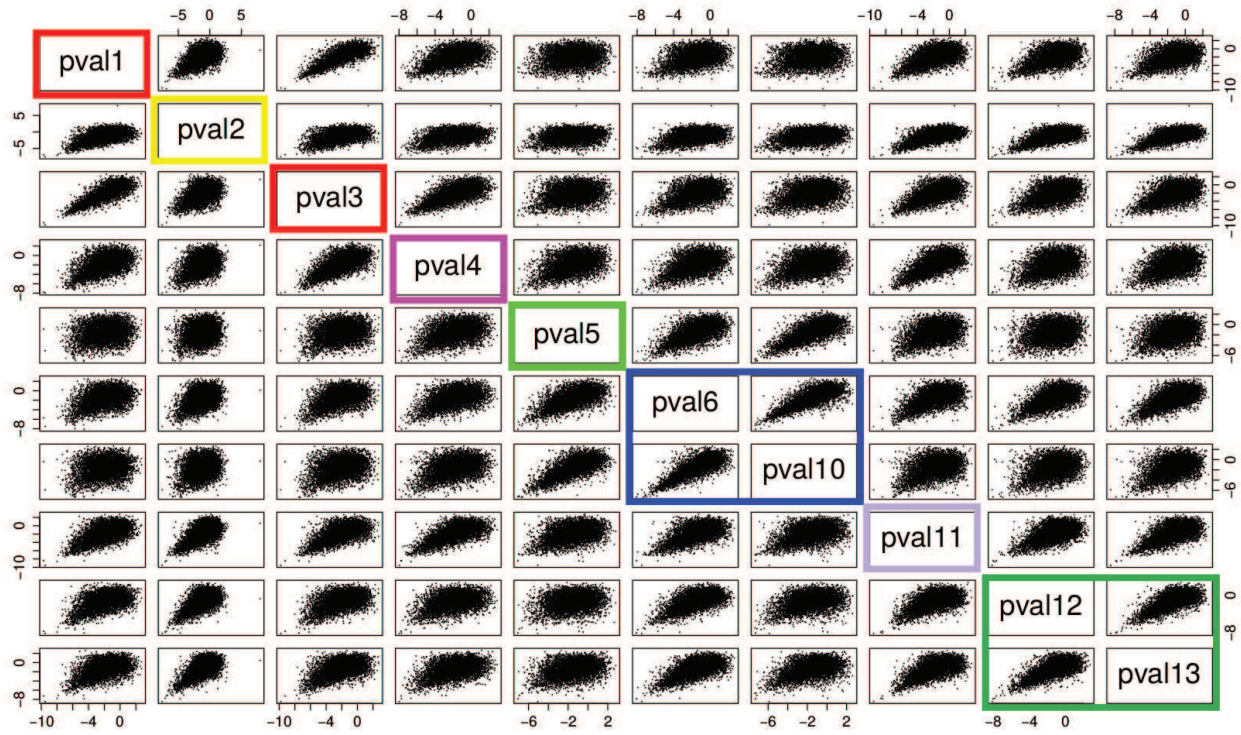


Figure 5.11 – Pairplots of probit transforms from maize data, Scaffold 1, 10 treatments: 1 to 6 and 10 to 13. The 7 colored rectangles show a data-driven possible dependencies.

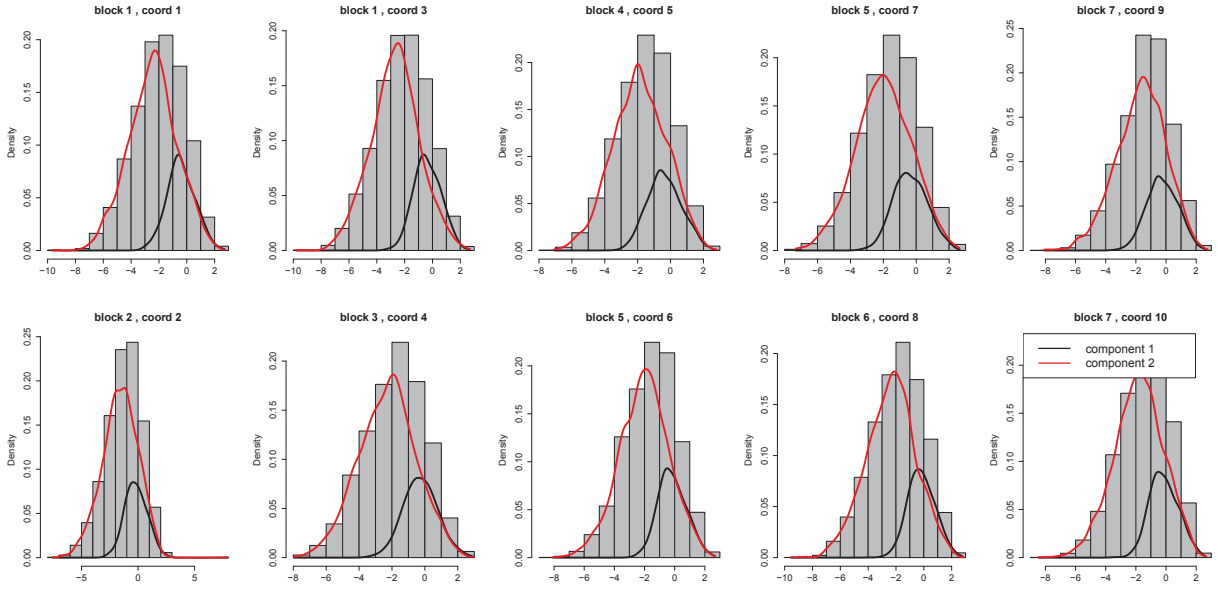


Figure 5.12 – The marginal density estimate plots of an example from maize data, Scaffold 1, $r = 10$ probit transforms of p-values corresponding to coordinate: 1 to 6 and 10 to 13; with block designed as (1, 2, 1, 3, 4, 5, 5, 6, 7, 7); `mvnpEMN01` solution.

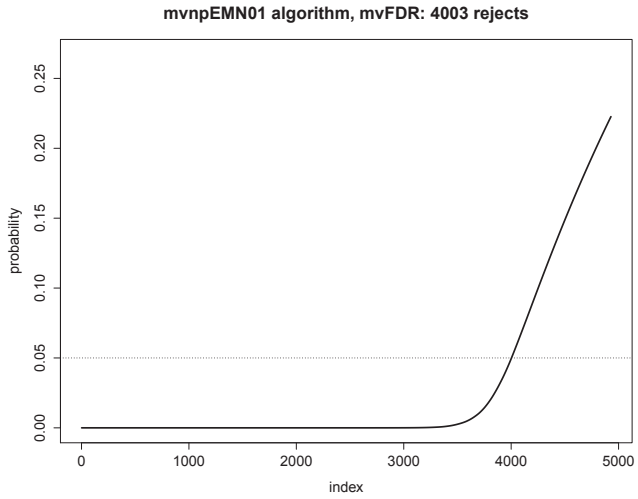


Figure 5.13 – The mvFDR control using `mvnpEMN01` algorithm of an example from maize data, Scaffold 1, $r = 10$ probit transforms of p-values corresponding coordinate (1:6,10:13) with block designed as (1, 2, 1, 3, 4, 5, 5, 6, 7, 7).

Chapter 6

Discussion and Perspective

In this work, we have first proposed (in Chapter 3) a nonparametric mixture model with conditionally independent multivariate blocks of nonparametric components. The conditional independence assumption has been introduced in several works in the literature, as e.g. in Hunt and Jorgensen [2003] in the context of parametric mixtures, but was limited so far in nonparametric mixture models to conditionally independent univariate coordinates. The crucial novelty of our model from a statistical modelling perspective is that it allows the dependence to be due not only to the mixture but also to the internal dependence structure of the multivariate distributions within each block. The identifiability of the parameters of our new model regardless the number of components m comes directly using a results from Allman et al. [2009]: actually we have merely pointed out that our model corresponds exactly to one of the theoretical setup developed in Allman et al. [2009].

We then proposed a multivariate EM-like algorithm for this model, called `mvnpEM` since it extends the `npEM` algorithm from Benaglia et al. [2009a] (the additional "mv" means that this model is "more multivariate" than `npEM` model). We have also introduced and described two strategies to select the bandwidth involved in the kernel density estimation step of this algorithm. The performance of this model has been evaluated through numerical studies with two perspectives. We experimented it focusing on parameter estimation (including the nonparametric multivariate densities), on three synthetic models: one allowing for comparison with the original `npEM` algorithm Benaglia et al. [2009a] and results from Hall et al. [2005] based on an inversion method (both designed for univariate blocks only, and 2 or at most 3 components); another more complex model showing that our algorithm behaves well in case of Gaussian, non-Gaussian with heavy tails, and non-Gaussian with both heavy tails and severely skewed bivariate blocks; a third model illustrating the clustering performance of our algorithm in the presence of strongly non-linear within-group dependencies. We also showed that some better estimates can result from the adaptive bandwidth strategy we have introduced, compared to a more immediate fixed bandwidth approach.

We have then experimented these new model and algorithm on an actual dataset, from the perspective of model-based unsupervised clustering in dimensions from 10 to 30. We compared our approach with the simple k-means algorithm, but also against a recent parametric but non-Gaussian model-based clustering alternative Hennig [2010]. This example allows us to illustrate, from a modelling perspective, the way to choose the

conditionally independent blocks from the structure of the data. By simple exploratory analysis of the data, one can recognize dependences between variables not obviously due to any mixture structure and group these variables in blocks. We have provided general guidelines for this block structure design in Section 3.6. We showed that, for several possible block designs, a clustering based on the Maximum A Posteriori (MAP) strategy using the estimated posterior matrix produced by our algorithm outperformed the two other approaches. Of course, there are many other existing methods to deal with the same problem. The purpose of this example was mostly to illustrate the applicability of our algorithm in real-size datasets and actual multi-dimensional models.

Both strategies about bandwidth selection for the kernel density estimation step of our algorithm use diagonal bandwidth matrices whose elements are computed from a fixed or adaptive weighted Silverman's rule. This rule is known to be somehow motivated by estimation of Gaussian-shaped distributions, which is too restrictive. Other strategies for the smoothing parameter, i.e. non diagonal bandwidth matrices, or cross-validation strategies are interesting perspectives for future investigations (see, e.g., Hyndman et al. [2004] for recent research on multivariate bandwidth selection, or Chauveau et al. [2015] for cross-validation techniques used for the smoothed `npEM` model).

Other extensions to the present model are possible. For instance, it is reasonable to allow the model to encompass the possibility that the block structure could be different in each component. In this case the set of coordinates $\{1, ..., r\}$ would be partitioned for component j into B_j disjoint subsets, i.e. $\{1, ..., r\} = \bigcup_{\ell=1}^{B_j} s_{j\ell}$ for $j = 1, \ldots, m$, where $d_{j\ell}$ would be the ℓth block dimension for component j. This extension would replace (3.1) by

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{\ell=1}^{B_j} f_{j\ell}(x_{is_{j\ell}}),$$

where $x_{is_{j\ell}}$ would now denote the coordinates in $s_{j\ell}$. An algorithm in the spirit of our `mvnpEM` for this extension can conceptually be done. However, there is a major issue with label switching in this model: since the algorithm would depend on the block structure per component, it would be necessary to ensure that "component j" always refers to a same particular component and structure in the model across iterations. This is a perspective for future developments.

Our `mvnpEM` algorithm (detailed in Chapter 3) lacks theoretical justification since its convergence is not proved: proving convergence in this nonparametric setup is difficult because the `wKDE` step is not a genuine maximization step of a $Q(\boldsymbol{\theta}|\boldsymbol{\theta}')$ operator. In Chapter 4, we proposed a maximum smoothed likelihood method and an alternative algorithm, called `mvnpMSL` that can be seen as a first building block towards a full proof of consistency. This algorithm incorporates similar ideas as Levine et al. [2011] for the EM-like algorithm `npEM` Benaglia et al. [2009a]. Under the assumption of conditionally independent blocks of coordinates, a smoothed multivariate finite mixture model was introduced by defining the nonlinear operators for the multivariate density functions. These nonlinear smoothing operators depend on the bandwidth matrix.

We considered maximum smoothed likelihood estimators (Eggermont and LaRiccia [2001]) which maximize a smoothed likelihood function and inherit all the important properties of probability density functions. A Majorization-Minimization (MM) algorithm

based on smoothed likelihood principal is suggested to compute numerically our density estimates. The proposed algorithm is in spirit similar to the majorization-minimization algorithm in Levine et al. [2011]. Extending from their ideas to the multivariate component density functions, we have proved that the `mvnpMSL` with conditionally independent multivariate blocks has the monotony property like any others EM algorithms. We showed that under finite samples, starting from any initial value, this algorithm not only decreases the smoothed likelihood function but also leads to estimates that minimize the smoothed likelihood function. This indicates the convergence of the algorithm but only towards a smoothed version of the log-likelihood, which is not a definitive proof of consistency. Some convergence results associated with `mvnpMSL` algorithm for our multivariate density estimates have been developed from the asymptotic convergence properties of univariate estimators proposed by Levine et al. [2011].

In the implementation, we studied the performance of our smoothed method on the three simulated examples presented in Chapter 3 and in Chauveau and Hoang [2016]. The real data example from the Wisconsin Breast Cancer datasets is also analyzed. Simulation studies show that the proposed method is as efficient as the empirical version in terms of mean integrated squared errors for the density estimates and mean squared errors for the Euclidean parameter estimates. Discretization of the intervals for non linear smoothing of the multivariate log-densities is the huge difficulty in empirical computing task, comparing with the `mvnpEM` algorithm from Chapter 3. This is also the major difference with the previous maximum smoothed likelihood estimators that already exits in Levine et al. [2011]. We made several experiments to define a multivariate grid and studied its effect based on the correct classification of the mixture and computing time.

Although we did serious improvement to get it better in term of CPU time, the smoothed algorithm takes more CPU time than the empirical one. It suggests to use in practice a hybrid method which combines both of two versions: using the `mvnpEM` algorithm to find a good initialization point for the monotone maximum smoothed likelihood algorithm to go to the optimization of the pseudo-loglikelihood. We have not explored this possibility yet. But we could try this interesting perspective in some difficult models.

As we mentioned at the end of Section 2.4.2, we have not yet convergence in the statistical sense. This kind of MSL algorithm does minimize/maximize an objective function which is not a true loglikelihood of the statistical model as commented by Levine et al. [2011]. The maximum smoothed likelihood method fails to yield a consistent estimator. This may impose difficulty in the subsequent technical development. Recently, a doubly smoothed maximum likelihood estimator (DS-MLE) (Seo and Lindsay [2010]) was proposed as a general alternative to the ordinary maximum likelihood estimator. It may be possible to propose a new estimation method using this doubly smoothed maximum likelihood ideas, which can give a potential high efficiency. Yet, the corresponding theoretical properties as well as the numerical performance of these estimates are left unknown. This is an interesting perspective for future work.

In Chapter 5 we have proposed an approach specializing our model and algorithm for False Discovery Rate (FDR) estimation which plays an important role in many high dimensional hypotheses testing framework. Instead of a single test as in common FDR framework in the literature, we assumed a multivariate multiple hypotheses testing framework whose global hypothesis H_0 is precised True if and only if H_0^k in kth test is True for

all k of r tests. The motivation is that multivariate FDR (mvFDR) should bring more improvement than the univariate FDR (univFDR) in case of multivariate statistics available. Then we have r-dimensional observed p-values $\boldsymbol{p}_i$ for each ith case of n cases. We established a simple 2-component mixture model with one component known for the probit transform of the $n \times r$ of matrix of $\boldsymbol{p}$-values. Under conditional independence assumption, the parameters could be estimated by a specific, constrained version of `npEM` algorithm. The identifiability of parameters is inferred directly from the results in Theorem 8 of Allman et al. [2009]. However, this identifiability property is destroyed when we extend to a 3-component mixture model because of the constraint Gaussian density imposed on pdf of the null hypothesis as well as others constraints on the pdf of the alternative hypothesis. Our new model and a constrained version of our `mvnpEM` algorithm allows to build blocks of coordinates which is able not only to impose the constraints for the component densities but also to keep their linearly dependent property, i.e. we can use within blocks dependence to achieve identifiability of the model following the condition of Theorem 9 in Allman et al. [2009]. We specified a multivariate nonparametric EM algorithm with the first component known as d_ℓ-dimensional standard normal distribution for ℓth block, called `mvnpEMN01`, which also uses the multivariate weighted kernel density estimator to estimate the jth component density, for all $j \neq 1$.

To see the convenience of our new algorithm, we have discussed some comparison criteria between our new "mvFDR" strategy (using our `mvnpEMN01` algorithm) and univFDR strategy (using `fdrtools` of Strimmer [2008a]). As we mentioned, we cannot have a fair comparison because the information from multivariate and univariate p-values is not comparable. The percentage of wrongly rejected cases (and of False Negative); the MSE on the target level α over all the replications are however meaningful. The procedure of univFDR control in univariate case has a step of ordering the sequence of p-values which is not possible in multivariate case. Some alternatives have been suggested. In mvFDR control, we apply our `mvnpEMN01` algorithm for the probit transform of multivariate $\boldsymbol{p}$-values and sorted the posterior probability of belonging to component 1 after the convergence.

In the implementation, some examples of a complex hypothesis testing model with one component corresponding to the null and others components to the alternative were simulated at the level of the probit transform of a multivariate $\boldsymbol{p}$-value. The comparisons from the FP and FN rates have precised the difference between univFDR strategy for each coordinate and mvFDR strategy. Monte Carlo simulations also have been done to compare the MSE on the target level α. The mvFDR strategy is definitely more effective and has got the advantage in case of a 3-component model where the `npEM` does not work because of the non-identifiability under some restrictions imposed on the component densities. The `mvnpEMN01` is general for an arbitrary r-dimensional hypothesis testing so that designing more complex models and applying to rich real dataset is an ongoing work. In addition, studying the simulated models and real data to see how would behaves a procedure using r results of the r univFDR control to decide weather the global H_0 or H_1 is true in $m = 2$ case is another area in which further work could bring some light.

We used parts of a large real dataset to compare the behavior of our constrained model with the generic one. This data from a micro-array experiment contains the results from the univFDR which shows the individual percentage of rejected cases for all 10 scaffold and 21 treatments for some maize hybridization experiment. On some 2-component mix-

ture models built by selecting some treatments and designing the meaningful blocks of coordinate according to the pairplots, the number of mvFDR control rejections from applying the constrained `mvnpEMN01` algorithm is more expected and accurate than from the plain `mvnpEM` algorithm. Considering more complex models such as mixture with $m \geq 3$ components in a real data case is an interesting perspective.

Finally, our nonparametric mixture models with conditionally independent multivariate component densities introduced in Chapter 3 is published in Computational Statistics and Data Analysis journal, Vol 103, 1–16. The `mvnpEM` algorithm of this model is also publicly available in the last update (version 1.0.4 released in January 2016) of the mixtools package Benaglia et al. [2009b] for the R statistical software R Core Team [2016]. Future revisions of this package may include the smoothed version `mvnpMSL` of `mvnpEM` as well as the constrained algorithm `mvnpEMN01`.

The idea in relaxing the conditional independence of coordinates and allowing dependence within each block of coordinates and even within component has provided several worth researches which can be applied for real data. Our models have opened many new investigations in EM-like algorithm for mixture model problems and contributed effectively in this direction as well.

Appendix A

The help files from `mixtools`

EM-like Algorithm for Nonparametric Mixture Models with Conditionally Independent Multivariate Component Densities

Description

An extension of the original npEM algorithm, for mixtures of multivariate data where the coordinates of a row (case) in the data matrix are assumed to be made of independent but multivariate blocks (instead of just coordinates), conditional on the mixture component (subpopulation) from which they are drawn (Chauveau and Hoang 2015).

Usage

```
mvnpEM(x, mu0, blockid = 1:ncol(x), samebw = TRUE,
       bwdefault = apply(x,2,bw.nrd0), init = NULL,
       eps = 1e-8, maxiter = 500, verb = TRUE)
```

Arguments

x	An *n x r* matrix of data. Each of the *n* rows is a case, and each case has *r* repeated measurements. These measurements are assumed to be conditionally independent, conditional on the mixture component (subpopulation) from which the case is drawn.
mu0	Either an *m x r* matrix specifying the initial centers for the kmeans function, or an integer *m* specifying the number of initial centers, which are then chosen randomly in kmeans
blockid	A vector of length *r* identifying coordinates (columns of x) that are in the same block. The default has all distinct elements, indicating that the model has *r* blocks of dimension 1, in which case the model is handled directly by the npEM algorithm. See example below for actual multivariate blocks example.
samebw	Logical: If TRUE, use the same bandwidth per coordinate for all iteration and all components. If FALSE, use a separate bandwidth for each component and coordinate, and update this bandwidth at each iteration of the algorithm using a suitably modified bw.nrd0 method as described in Benaglia et al (2011) and Chauveau and Hoang (2015).
bwdefault	Bandwidth default for density estimation,a simplistic application of the default bw.nrd0 for each coordinate (column) of the data.
init	Initialization method, based on an initial *n x m* matrix for the posterior probabilities. If NULL, a kmeans clustering with mu0 initial centers is applied to the data and the initial matrix of posteriors is built from the result.
eps	Tolerance limit for declaring algorithm convergence. Convergence is declared whenever the maximum change in any coordinate of the lambda vector (of mixing proportion estimates) does not exceed eps.
maxiter	The maximum number of iterations allowed; convergence may be declared before maxiter iterations (see eps above).
verb	Verbose mode; if TRUE, print updates for every iteration of the algorithm as it runs

Value

mvnpEM returns a list of class mvnpEM with the following items:

data	The raw data (an *n x r* matrix).
posteriors	An *n x m* matrix of posterior probabilities for each observation (row).
lambda	The sequence of mixing proportions over iterations.
blockid	The blockid input argument. Needed by any method that produces density estimates from the output, like plot.mvnpEM.
samebw	The samebw input argument. Needed by any method that produces density estimates from the output, like plot.mvnpEM.
bandwidth	The final bandwidth matrix after convergence of the algorithm. Its shape depends on the samebw input argument. If samebw = TRUE, a vectors with the bandwidth value for each of the r coordinates (same for all components and iterations). If samebw = FALSE, a *m x r* matrix, where each row is associated to one component and gives the *r* bandwidth values, one for each coordinate. Needed by any method that produces density estimates from the output, like plot.mvnpEM.
lambdahat	The final mixing proportions.
loglik	The sequence of pseudo log-likelihood values over iterations.

References

- Benaglia, T., Chauveau, D., and Hunter, D. R. (2009), An EM-like algorithm for semi- and non-parametric estimation in multivariate mixtures, Journal of Computational and Graphical Statistics, 18, 505-526.
- Benaglia, T., Chauveau, D. and Hunter, D.R. (2011), Bandwidth Selection in an EM-like algorithm for nonparametric multivariate mixtures. Nonparametric Statistics and Mixture Models: A Festschrift in Honor of Thomas P. Hettmansperger. World Scientific Publishing Co., pages 15-27.
- Chauveau, D., and Hoang, V. T. L. (2015), Nonparametric mixture models with conditionally independent multivariate component densities, Preprint under revision. https://hal.archives-ouvertes.fr/hal-01094837

See Also

plot.mvnpEM, npEM

Examples

```
# Example as in Chauveau and Hoang (2015) with 6 coordinates
## Not run:
m=2; r=6; blockid <-c(1,1,2,2,3,3) # 3 bivariate blocks
# generate some data x ...
a <- mvnpEM(x, mu0=2, blockid, samebw=F) # adaptive bandwidth
plot(a) # this S3 method produces 6 plots of univariate marginals
summary(a)
## End(Not run)
```

Plots of Marginal Density Estimates from the mvnpEM Algorithm Output

Description

Takes an object of class `mvnpEM`, as the one returned by the `mvnpEM` algorithm, and returns a set of plots of the density estimates for each coordinate within each multivariate block. All the components are displayed on each plot so it is possible to see the mixture structure for each coordinate and block. The final bandwidth values are also displayed, in a format depending on the bandwidth strategy .

Usage

```
## S3 method for class 'mvnpEM'
plot(x, truenorm = FALSE, mu = NULL, v = NULL,
            lgdcex = 1, ...)
```

Arguments

`x`	An object of class `mvnpEM` such as the output of the `mvnpEM` function
`truenorm`	Mostly for checking purpose, if the nonparametric model is to be compared with a multivariate Gaussian mixture as the true model.
`mu`	true mean parameters, for Gaussian models only (see above)
`v`	true covariance matrices, for Gaussian models only (see above)
`lgdcex`	Character expansion factor for `legend`.
`...`	Any remaining arguments are passed to `hist`.

Value

`plot.mvnpEM` currently just plots the figure.

See Also

`mvnpEM`, `npEM`, `density.npEM`

Examples

```
# example as in Chauveau and Hoang (2015) with 6 coordinates
## Not run:
m=2; r=6; blockid <-c(1,1,2,2,3,3) # 3 bivariate blocks
# generate some data x ...
a <- mvnpEM(x, mu0=2, blockid, samebw=F) # adaptive bandwidth
plot(a) # this S3 method produces 6 plots of univariate marginals
summary(a)
## End(Not run)
```

Bibliography

Allison, D. B., Gadbury, G. L., Heo, M., Fernández, J. R., Lee, C.-K., Prolla, T. A., and Weindruch, R. (2002). A mixture model approach for the analysis of microarray gene expression data. *Computational Statistics & Data Analysis*, 39(1):1–20.

Allman, E. S., Matias, C., and Rhodes, J. A. (2009). Identifiability of parameters in latent structure models with many observed variables. *Ann. Statist*, 37(6A):3099–3132.

Benaglia, T., Chauveau, D., and Hunter, D. R. (2009a). An EM-like algorithm for semi- and non-parametric estimation in multivariate mixtures. *Journal of Computational and Graphical Statistics*, 18(2):505–526.

Benaglia, T., Chauveau, D., and Hunter, D. R. (2011). *Bandwidth selection in an EM-like algorithm for nonparametric multivariate mixtures, in :Nonparametric Statistics and Mixture Models: A Festschrift in Honor of Thomas P. Hettmansperger*, pages 15–27. World Scientific Publishing Co.

Benaglia, T., Chauveau, D., Hunter, D. R., and Young, D. (2009b). mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57:289–300.

Bhattacharya, C. (1967). A simple method of resolution of a distribution into gaussian components. *Biometrics*, pages 115–135.

Biernacki, C., Celeux, G., Govaert, G., and Langrognet, F. (2006). Model-based cluster and discriminant analysis with the mixmod software. *Computational Statistics & Data Analysis*, 51(2):587–600.

Böhning, D., Dietz, E., and Schlattmann, P. (1998). Recent developments in computer-assisted analysis of mixtures. *Biometrics*, pages 525–536.

Bordes, L. and Chauveau, D. (2017). Stochastic EM algorithms for parametric and semi-parametric mixture models for right-censored lifetime data. *Computational Statistics*, 31(4):1513–1538.

Bordes, L., Chauveau, D., and Vandekerkhove, P. (2006a). An EM algorithm for a semi-parametric mixture model. *Comput. Statist. Data Anal.*

Bordes, L., Chauveau, D., and Vandekerkhove, P. (2007). A stochastic EM algorithm for a semiparametric mixture model. *Computational Statistics and Data Analysis*, 51(11):5429–5443.

Bordes, L., Delmas, C., and Vandekerkhove, P. (2006b). Semiparametric estimation of a two-component mixture model where one component is known. *Scand. J. Statistics*, 33:733–752.

Bordes, L., Delmas, C., and Vandekerkhove, P. (2006c). Semiparametric estimation of a two-component mixture model where one component is known. *Scandinavian journal of statistics*, 33(4):733–752.

Bordes, L., Mottelet, S., and Vandekerkhove, P. (2006d). Semiparametric estimation of a two-component mixture model. *Annals of Statistics*, 34(3):1204–1232.

Broniatowski, M., Celeux, G., and Diebolt, J. (1983). Reconnaissance de mélanges de densités par un algorithme d'apprentissage probabiliste. *Data analysis and informatics*, 3:359–373.

Celeux, G., Chauveau, D., and Diebolt, J. (1996). Stochastic versions of the EM algorithm: An experimental study in the mixture case. *J. Statist. Comput. Simul.*, 55:287–314.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational statistics quarterly*, 2(1):73–82.

Celeux, G. and Diebolt, J. (1989). *Une version de type recuit simulé de l'algorithme EM*. PhD thesis, INRIA.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, 28(5):781–793.

Charlier, C. V. L. (1906). Researches into the theory of probability. *Meddelanden fran Lunds Astronomiska Observatorium Serie II*, 4:3–59.

Charlier, C. V. L. and Wicksell, S. (1923). On the dissection of frequency functions. *Meddelanden fran Lunds Astronomiska Observatorium Serie I*, 103:1–64.

Chauveau, D. (1995). A stochastic EM algorithm for mixtures with censored data. *Journal of statistical planning and inference*, 46(1):1–25.

Chauveau, D. and Hoang, V. T. L. (2016). Nonparametric mixture models with conditionally independent multivariate component densities. *Computational Statistics and Data Analysis*, 103:1–16.

Chauveau, D., Hunter, D. R., Levine, M., et al. (2015). Semi-parametric estimation for conditional independence multivariate finite mixture models. *Statistics Surveys*, 9:1–31.

Chauveau, D., Saby, N. P., Orton, T. G., Lemercier, B., Walter, C., and Arrouays, D. (2014). Large-scale simultaneous hypothesis testing in monitoring carbon content from french soil database: A semi-parametric mixture approach. *Geoderma*, 219:117–124.

Chi, Z. et al. (2008). False discovery rate control with multivariate p-values. *Electronic Journal of Statistics*, 2:368–411.

Cohen, A. C. (1967). Estimation in mixtures of two normal distributions. *Technometrics*, 9(1):15–28.

De Leeuw, J. and Heiser, W. J. (1977). Convergence of correction matrix algorithms for multidimensional scaling. *Geometric representations of relational data*, pages 735–752.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Di Marzio, M. and Taylor, C. C. (2004). Boosting kernel density estimates: A bias reduction technique? *Biometrika*, pages 226–233.

Diebolt, J. and Robert, C. P. (1994). Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 363–375.

Efron, B., Tibshirani, R., et al. (1996). Using specially designed exponential families for density estimation. *The Annals of Statistics*, 24(6):2431–2461.

Efron, B., Tibshirani, R., Storey, J., and Tusher, V. (2001). Empirical bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96(456):1151–1160.

Eggermont, P. and Lariccia, V. N. L. (1999). Optimal convergence rates for good's nonparametric maximum likelihood density estimator. *The Annals of Statistics*, 27(5):1600–1615.

Eggermont, P. P. B. (1999). Nonlinear smoothing and the em algorithm for positive integral equations of the first kind. *Applied Mathematics and Optimization*, 39(1):75–91.

Eggermont, P. P. B. and LaRiccia, V. N. (2001). *Maximum Penalized Likelihood Estimation, Volume I: Density Estimation*. Springer.

Elmore, R. T., Hettmansperger, T. P., and Thomas, H. (2004). Estimating component cumulative distribution functions in finite mixture models. *Comm. Statist. Theory Methods*, 33(9):2075–2086.

Frühwirth-Schnatter, S. (2006). *Finite mixture and Markov switching models*. Springer Science & Business Media.

Green, P. J. (1990). Bayesian reconstructions from emission tomography data using a modified em algorithm. *IEEE transactions on medical imaging*, 9(1):84–93.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernouilli*, 7(2):223–242.

Hall, P., Neeman, A., Pakyari, R., and Elmore, R. T. (2005). Nonparametric inference in multivariate mixtures. *Biometrika*, 92(3):667–678.

Hall, P. and Zhou, X. H. (2003). Nonparametric estimation of component distributions in a multivariate mixture. *Annals of Statistics*, 31:201–224.

Hennig, C. (2010). Methods for merging gaussian mixture components. *Advances in data analysis and classification*, 4(1):3–34.

Hettmansperger, T. and Thomas, H. (2000). Almost nonparametric inference for repeated measures in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):811–825.

Hoti, F. and Holmström, L. (2004). A semiparametric density estimation approach to pattern classification. *Pattern Recognition*, 37(3):409–419.

Hunt, L. and Jorgensen, M. (2003). Mixture model clustering for mixed data with missing information. *Computational Statistics & Data Analysis*, 41(3):429–440.

Hunter, D. R. and Lange, K. (2000). Quantile regression via an mm algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60–77.

Hunter, D. R., Wang, S., and Hettmansperger, T. P. (2007). Inference for mixtures of symmetric distributions. *Annals of Statistics*, 35(1):224–251.

Hyndman, R. L., Zhang, X., and King, M. L. (2004). Bandwidth selection for multivariate kernel density estimation using MCMC. Econometric Society 2004 Australasian Meetings 120, Econometric Society.

Jordan, M. and Xu, L. (1995). On convergence properties of the em algorithm for gaussian mixtures.

Krishnan, T. and McLachlan, G. (1997). The em algorithm and extensions. *Wiley*, 1(997):58–60.

Lee, S. X. and McLachlan, G. J. (2013). On mixtures of skew normal and skew t-distributions. *Advances in Data Analysis and Classification*, 7(3):241–266.

Levine, M., Hunter, D. R., and Chauveau, D. (2011). Maximum smoothed likelihood for multivariate mixtures. *Biometrika*, 98(2):403–416.

Liao, J., Lin, Y., Selvanayagam, Z. E., and Shih, W. J. (2004). A mixture model for estimating the local false discovery rate in dna microarray analysis. *Bioinformatics*, 20(16):2694–2701.

Lindsay, B. G. (1995). *Mixture Models: theory, geometry, and applications*. Ims.

Lindsay, B. G. et al. (1983). The geometry of mixture likelihoods: a general theory. *The annals of statistics*, 11(1):86–94.

Louis, T. A. (1982). Finding the observed information matrix when using the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 226–233.

McLachlan, G., Bean, R., and Jones, B.-T. (2006). A simple implementation of a normal mixture approach to differential gene expression in multiclass microarrays. *Bioinformatics*, 22:1608–1615.

McLachlan, G. and Peel, D. (2000). *Finite mixture models*. Wiley Series in Probability and Statistics: Applied Probability and Statistics. Wiley-Interscience, New York.

McLachlan, G. J. and Basford, K. E. (1988). Mixture models. inference and applications to clustering. *Statistics: Textbooks and Monographs, New York: Dekker, 1988*, 1.

McLachlan, G. J. and Krishnan, T. (2008). *The EM algorithm and extensions*. Wiley Series in Probability and Statistics: Applied Probability and Statistics. Wiley-Interscience, New York.

McLachlan, G. J., Peel, D., Basford, K. E., and Adams, P. (1999). The emmix software for the fitting of mixtures of normal and t-components. *Journal of Statistical Software*, 4(2):1–14.

Meilijson, I. (1989). A fast improvement to the em algorithm on its own terms. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 127–138.

Nguyen, V. H. and Matias, C. (2014). Nonparametric estimation of the density of the alternative hypothesis in a multiple testing setup. application to local false discovery rate estimation. *ESAIM: Probability and Statistics*, 18:584–612.

Olkin, I. and Spiegelman, C. H. (1987). A semiparametric approach to density estimation. *Journal of the American Statistical Association*, 82(399):858–865.

Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative solution of nonlinear equations in several variables*, volume 30. Siam.

Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110.

Plasse, J. H. (2013). The EM algorithm in multivariate gaussian mixture models using anderson acceleration. Technical report, Worcester Polytechnic Institute.

Priebe, C. E. and Marchette, D. J. (2000). Alternating kernel and mixture density estimates. *Computational Statistics & Data Analysis*, 35(1):43–65.

Quetelet, L. A. J. (1846). *Lettres... sur la theorie des probabilites appliquee aux sciences morales et politiques*. Hayez.

R Core Team (2016). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna. bibtex: r_core_team_r:_2016.

Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM review*, 26(2):195–239.

Robin, S., Bar-Hen, A., and Daudin, J.-J. (2005). A semiparametric approach for mixture models: Application to local fdr estimation. *Preprint INA/INRIA, France*.

Robin, S., Bar-Hen, A., Daudin, J.-J., and Pierre, L. (2007). A semi-parametric approach for mixture models: Application to local false discovery rate estimation. *Computational Statistics and Data Analysis*, 51.

Roeder, K. (1994). A graphical technique for determining the number of components in a mixture of normals. *Journal of the American Statistical Association*, 89(426):487–495.

Sawant, K. B. (2015). Efficient determination of clusters in k-mean algorithm using neighborhood distance. *International Journal of Emerging Engineering Research and Technology*, 3:22–27.

Schlattmann, P. (2009). *Medical applications of finite mixture models*. Springer Science & Business Media.

Schweder, T. and Spjøtvoll, E. (1982). Plots of p-values to evaluate many tests simultaneously. *Biometrika*, 69(3):493–502.

Scott, D. W. (1992). *Multivariate density estimation*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York. Theory, practice, and visualization, A Wiley-Interscience Publication.

Seo, B. and Lindsay, B. G. (2010). A computational strategy for doubly smoothed mle exemplified in the normal mixture model. *Computational Statistics & Data Analysis*, 54(8):1930–1941.

Shen, Z., Levine, M., and Shang, Z. (2016). A maximum smoothed likelihood based estimation for two component semiparametric density mixtures with a known component. *arXiv preprint arXiv:1611.06575*.

Silverman, B., Jones, M., Wilson, J., and Nychka, D. (1990). A smoothed em approach to indirect estimation problems, with particular, reference to stereology and emission tomography. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–324.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability. Chapman & Hall, London.

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):479–498.

Strimmer, K. (2008a). fdrtool: a versatile r package for estimating local and tail area-based false discovery rates. *Bioinformatics*, 24(12):1461–1462.

Strimmer, K. (2008b). A unified approach to false discovery rate estimation. *BMC Bioinformatics*, 9.

Sundberg, R. (1972). *Maximum Likelihood Theory and Applications for Distributions Generated when Observing a Function an Exponential Family Variable*. Institute of Mathematical Statics, Stockholm University.

Sundberg, R. (1974). Maximum likelihood theory for incomplete data from an exponential family. *Scandinavian Journal of Statistics*, pages 49–58.

Sundberg, R. (1976). An iterative method for solution of the likelihood equations for incomplete data from exponential families. *Communication in Statistics-Simulation and Computation*, 5(1):55–64.

Tanner, M. A. (1991). *Tools for statistical inference*, volume 3. Springer.

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540.

Vandewalle, V. (2009). *Estimation et sélection en classification semi-supervisée*. PhD thesis, Université des Sciences et Technologie de Lille-Lille I.

Wei, G. C. and Tanner, M. A. (1990). A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704.

Wu, C. J. (1983). On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103.

Young, D. S., Benaglia, T., Chauveau, D., Elmore, R. T., Hettmansperger, T. P., Hunter, D. R., Thomas, H., and Xuan, F. (2009). mixtools: Tools for mixture models. R package version 0.3.3.

Zhu, X. and Hunter, D. R. (2015). Clustering via finite nonparametric ica mixture models. *arXiv preprint arXiv:1510.08178*.

Vy-Thuy-Lynh HOANG

Modèles et algorithmes d'estimation pour des mélanges finis de densités de composantes multivariées nonparamétriques et conditionnellement indépendantes

Résumé : Plusieurs auteurs ont proposé récemment des modèles et des algorithmes pour l'estimation nonparamétrique de mélanges multivariés finis dont l'identifiabilité n'est pas toujours assurée. Entre les modèles considérés, l'hypothèse des coordonnées indépendantes conditionnelles à la sous-population de provenance des individus fait l'objet d'une attention croissante, en raison des développements théoriques et pratiques envisageables, particulièrement avec la multiplicité des variables qui entrent en jeu dans le framework statistique moderne. Dans ce travail, nous considérons d'abord un modèle plus général supposant l'indépendance, conditionnellement à la composante, de blocs multivariés de coordonnées au lieu de coordonnées univariées, permettant toute structure de dépendance à l'intérieur de ces blocs. Par conséquent, les fonctions de densité des blocs sont complètement multivariées et non paramétriques. Nous présentons des arguments d'identifiabilité et introduisons pour l'estimation dans ce modèle deux algorithmes méthodologiques dont les procédures de calcul ressemblent à un véritable algorithme EM mais incluent une étape additionnelle d'estimation de densité: un algorithme rapide montrant l'efficacité empirique sans justification théorique, et un algorithme lissé possédant une propriété de monotonie comme certain algorithme EM, mais plus exigeant en terme de calcul. Nous discutons également les méthodes efficaces en temps de calcul pour l'estimation et proposons quelques stratégies. Ensuite, nous considérons une extension multivariée des modèles de mélange utilisés dans le cadre de tests d'hypothèses multiples, permettant une nouvelle version multivariée de contrôle du False Discovery Rate. Nous proposons une version contrainte de notre algorithme précédent, adaptée spécialement à ce modèle. Le comportement des algorithmes de type EM que nous proposons est étudié numériquement dans plusieurs expérimentations de Monte Carlo et sur des données réelles de grande dimension et comparé avec les méthodes existantes dans la littérature. Enfin, les codes de nos nouveaux algorithmes sont progressivement ajoutés sous forme de nouvelles fonctions dans le package en libre accès `mixtools` pour le logiciel de statistique `R`.

Mots clés : Algorithme EM, Estimation non-paramétrique de densité multivariées, Mélanges non-paramétriques multivariés.

Models and estimation algorithms for nonparametric finite mixtures with conditionally independent multivariate component densities

Abstract: Recently several authors have proposed models and estimation algorithms for finite non-parametric multivariate mixtures, whose identifiability is typically not obvious. Among the considered models, the assumption of independent coordinates conditional on the subpopulation from which each observation is drawn is subject of an increasing attention, in view of the theoretical and practical developments it allows, particularly with multiplicity of variables coming into play in the modern statistical framework. In this work we first consider a more general model assuming independence, conditional on the component, of multivariate blocks of coordinates instead of univariate coordinates, allowing for any dependence structure within these blocks. Consequently, the density functions of these blocks are completely multivariate and nonparametric. We present identifiability arguments and introduce for estimation in this model two methodological algorithms whose computational procedures resemble a true EM algorithm but include an additional density estimation step: a fast algorithm showing empirical efficiency without theoretical justification, and a smoothed algorithm possessing a monotony property as any EM algorithm does, but more computationally demanding. We also discuss computationally efficient methods for estimation and derive some strategies. Next, we consider a multivariate extension of the mixture models used in the framework of multiple hypothesis testings, allowing for a new multivariate version of the False Discovery Rate control. We propose a constrained version of our previous algorithm, specifically designed for this model. The behavior of the EM-type algorithms we propose is studied numerically through several Monte Carlo experiments and high dimensional real data, and compared with existing methods in the literature. Finally, the codes of our new algorithms are progressively implemented as new functions in the publicly-available package `mixtools` for the `R` statistical software.

Keywords: EM algorithm, Nonparametric, Mixture models, multivariate component densities.

[MAPMO UMR 6628 - CNRS
Rue de Chartres - BP6759
45067 ORLÉANS CEDEX]