Nonparametric mixture models with conditionally independent multivariate component densities

Didier Chauveau, Vy Thuy Lynh Hoang

Nonparametric mixture models with conditionally independent multivariate component densities

Didier Chauveau* Vy Thuy Lynh Hoang †

June 21, 2015

Abstract

Recent works in the literature have proposed models and algorithms for nonparametric estimation of finite multivariate mixtures. In these works, independent coordinates conditional on the subpopulation from which each observation is drawn is assumed, so that the dependence structure comes only from the mixture. Here this assumption is relaxed, allowing for independent multivariate *blocks* of coordinates, conditional on the subpopulation from which each observation is drawn. Otherwise the blocks density functions are completely multivariate and nonparametric. We propose an EM-like algorithm for this model, and derive some strategies for selecting the bandwidth matrix involved in the nonparametric estimation step of it. The performance of this algorithm is evaluated through several numerical simulations. We also experiment this new model and algorithm on an actual dataset from the model based, unsupervised clustering perspective, to illustrate its potential.
keywords. EM algorithm, multivariate kernel density estimation, multivariate mixture, nonparametric mixture.

1 Introduction

Populations of individuals may often be divided into subgroups. The task in 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 literature, and in fact mixture models may be generally thought of as comprising the subset of clustering methods known as model-based clustering.

Finite mixture models may also 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 form and quite often this form is univariate or multivariate normal.

*Univ. Orléans, CNRS, MAPMO, UMR 7349, Orléans, France, didier.chauveau@univ-orleans.fr
†vy-thuy-lynh.hoang@etu.univ-orleans.fr

The most general model for nonparametric multivariate 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

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

where $\boldsymbol{x}_i \in \mathbb{R}^r$, and $\boldsymbol{\theta} = (\boldsymbol{\lambda}, \boldsymbol{f}) = (\lambda_1, \ldots, \lambda_m, f_1, \ldots, f_m)$ denotes the parameters of the statistical model. In this model 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 multivariate density functions $\mathcal{F}$ absolutely continuous with respect to Lebesgue measure. Note that the univariate ($r = 1$) case will only be briefly considered, since this paper focus on multivariate extensions.

Model (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 (1) and also that we do not consider that "label-switching" — i.e., reordering the m pairs $(\lambda_1, f_1), \ldots, (\lambda_m, f_m)$ — produces a distinct representation. 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. The most used and studied parametric mixture model is the Gaussian mixture, where f_j is the density of a (eventually multidimensional) Gaussian distribution with mean μ_j and variance (matrix) Σ_j. Section 1.2 presents various ways of relaxing this parametric assumption while preserving some sort of identifiability property.

1.1 The EM algorithm

Mixture models are deeply connected to the EM algorithm. 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 in missing data problems, of which finite mixture problems 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 EM principle, properties and generalizations can be found in McLachlan and Krishnan (2008), and mixture models are deeply detailed in McLachlan and Peel (2000).

In a missing data setup, the n-fold product of the pdf of the observations $g_{\boldsymbol{\theta}}$ corresponds to the *incomplete* data pdf, associated to the log-likelihood $\ell_{\boldsymbol{x}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log g_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$. In mixture models and many other missing data situations, maximizing $\ell_{\boldsymbol{x}}(\boldsymbol{\theta})$ 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. Given an arbitrary starting value $\boldsymbol{\theta}^{(0)}$, the EM algorithm generates a sequence $(\boldsymbol{\theta}^{(t)})_{t \geq 1}$ by iterating the following steps:

1. E-step: `compute` $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$

2. M-step: `set` $\theta^{(t+1)} = \text{argmax}_{\boldsymbol{\theta}\in\Theta} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$.

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. Notationally, 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 (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,

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

Conveniently, 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

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

where $p_{ij}^{(t)} := \mathbb{P}_{\boldsymbol{\theta}^{(t)}}(Z_{ij} = 1|\boldsymbol{x}_i)$ is the *posterior probability* that the individual i comes from component j. The 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 is often easy, and can be looked like weighted MLE estimates. This is the case for, e.g., Gaussian mixtures.

1.2 Previous work on semi- and non-parametric mixtures

In this work, the term "nonparametric" means 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 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. (2006) 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 (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.

On 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 (1) becomes

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jk}(x_{ik}). \tag{2}$$

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 but not in fewer than three. Hall et al. (2005) extended this result, suggesting that the condition on m gets less restrictive as r increases; intuitively, dimensionality together with conditional independence help for identifiability. This results, nowadays known as the "curse of dimensionality in the reverse", says that for a given number of components m, there is a lower bound r_m that the dimensionality of observations must exceed for the model to be identifiable. Allman et al. (2009) finally established the fundamental result of identifiability for model (2) 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. (2006) and Hunter et al. (2007) both propose estimators based on a minimum contrast approach, a method impossible to extend beyond $m = 2$ components. For the multivariate model (2), Hall et al. (2005) give estimators based on inversion of the mixture, that apply only in the case when $m = 2$ and $r = 3$, due to analytical difficulties appearing beyond this case.

The difficulties associated to these theoretically well grounded approaches encourage the development of estimation strategies based on the EM principle. In the univariate case, Bordes et al. (2007) first propose a univariate semiparametric (and stochastic) "EM-like" algorithm for a location-shift semiparametric mixture model

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

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 gives kernel-density-like estimates 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. This assignment is 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.

For the multivariate model (2), an empirical "EM-like" algorithm for statistical estimation of its parameter has been introduced in Benaglia et al. (2009a). This algorithm called `npEM` (non-parametric EM) eliminates the stochasticity of the univariate algorithm from Bordes et al. (2007), but also relies on a WKDE step for the updates of the f_{jk}'s. The corresponding `npEM` function for this algorithm is publicly available in the mixtools package Benaglia et al.

(2009b) for the R statistical software R Development Core Team (2010), and is designed to estimate $\boldsymbol{\theta}$ in model (2), and in some refinements of it. However, despite its empirical success, this algorithm lacks any sort of theoretical justification; indeed, it can only be called "EM-like" because it resembles an EM algorithm in certain aspects of its formulation. Levine et al. (2011) corrects this shortcoming by introducing a smoothed loglikelihood function and formulating an iterative algorithm with a provable monotonicity property that happens to produce results that are similar to those of Benaglia et al. (2009b) in practice.

This article describes a new nonparametric mixture models that extends model (2) in the sense that it allows for conditionally independent *multivariate and nonparametric* component densities. Importantly, this extensions 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). 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 2, and verify 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 these parameters in Section 3. 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 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 3.2. Section 4 is devoted to implementation considerations and a study of the algorithm through large scale Monte-Carlo simulations. Section 5 describes an analysis, using our model, of an actual dataset from the machine learning community. The perspective there is unsupervised model-based clustering, illustrating the potential usefulness of our new mixture model approach relaxing the conditional independence assumption.

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. lth block dimension. Actually, we will impose $B \geq 3$ in practice in view of Allman et al. (2009) result implying that there is little hope to have an identifiable model for less than 3 independent blocks (see Section 2.1 below).

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 $1 \leq l \leq B$ (m, r, B and n stand for the number of mixture components, repeated measurements, blocks, and the sample size). Suppose f_j is equal to the product of $f_{jl}-$the multivariate density function of jth component and lth block.

Then model (1) becomes

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{l=1}^{B} f_{jl}(x_{is_l}), \tag{3}$$

where $x_{is_l} = \{x_{ik}, k \in s_l\}$ is the multivariate variable which have its coordinates in 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 the model of conditional independence (2) introduced by Hall and Zhou (2003): here the dependence structure does not come only from the mixture structure, since some 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) (see, for instance, our actual data application Section 5).

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

2.1 Identifiability considerations

As reviewed briefly in Section 1.2, Hall et al. (2005) explored the identifiability question related to model (2) 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). 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. Their results are based on an algebraic result of Kruskal (1976, 1977), who describes a 3-way contingency table that cross-classifies a sample of n individuals with respect to three categorical variables, say X_k, $k = 1, 2, 3$, each X_k taking value in a state space $\{1, \ldots, \kappa_k\}$ with κ_k possible categories. This model assumes existence of a latent (unobservable) variable Z with values in $\{1, \ldots, m\}$ that is just an alternative coding of our binary variables Z_{ij}'s. It is also assumed that conditionally on knowing the exact class $\{Z = j\}$, the 3 observed variables are mutually independent. This model is thus precisely a version of model (2) for per-components and coordinate finite measures. Allman et al. (2009) denote this m-class, $r = 3$-features model $\mathcal{M}(m; \kappa_1, \kappa_2, \kappa_3)$. The full details are in their article, and a survey-like shorter description for application to model (2) can be found in Chauveau et al. (2015). We only summarize briefly this technique here, focusing on results concerning our model (3).

The representation of the $r = 3$ conditionally independent finite measures is done by defining matrices A_k of size $m \times \kappa_k$, $k = 1, 2, 3$, where each A_k's row j describes the probability distribution of $(X_k | Z = j)$. Defining $\lambda_j \overset{\text{def}}{:=} P(Z = j)$, and $\tilde{A}_1 = diag(\boldsymbol{\lambda}) A_1$, the probability distribution of the latent class model (the finite mixture) is associated to the $\kappa_1 \times \kappa_2 \times \kappa_3$ tensor $[\tilde{A}_1, A_2, A_3]$, that is the three-dimensional array whose element with coordinates (u_1, u_2, u_3) is a sum of products of elements of these three matrices, with column numbers u_1, u_2, u_3

respectively, added up over the m rows:

$$\left[\tilde{A}_1, A_2, A_3\right]_{u_1,u_2,u_3} = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{3} \mathbb{P}(X_k = u_k | Z = j).$$

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, and let $I_k = rank_K(A_k)$. Kruskal established that, if $I_1 + I_2 + I_3 \geq 2m+2$, then $[A_1, A_2, A_3]$ uniquely determines the A_k's, up to simultaneous permutation and rescaling of rows. Kruskal's result is a cornerstone of several subsequent results establishing identifiability criteria for various latent structure models. Allman et al. (2009) first reformulate it, proving identifiability of model $\mathcal{M}(m; \kappa_1, \kappa_2, \kappa_3)$ (up to label switching), providing that all entries of $\boldsymbol{\lambda}$ are positive. Then they extend that to the r-variate model $\mathcal{M}(m; \kappa_1, ..., \kappa_r)$ with $r \geq 3$, under the condition that there exists a tripartition of $\{1, ..., r\}$ into three disjoint nonempty subsets S_1, S_2, S_3, such that $\sum_{l=1}^{3} \min(m, \tau_l) \geq 2m + 2$, where $\tau_l = \prod_{k \in S_l} \kappa_k$.

Extension of Kruskal's work to finite mixtures of conditionally independent univariate nonparametric measures, that is model (2), is based on a judicious use of cut points to discretize the distributions associated to the f_{jk}'s (Theorem 8). Considering 3 random variables at a time only, each X_k is associated to $Y_k = \{\mathbf{1}_{\{X_k \in I_k^1\}}, ..., \mathbf{1}_{\{X_k \in I_k^{\kappa_k}\}}\}$, where $\mathbb{R}$ is partitionned into κ_k consecutive intervals $(I_k^l, 1 \leq l \leq \kappa_k)$. Stochastic matrices are built from this construction, using the f_{jk}'s associated c.d.f.s. It is possible to build these partitions general enough and well-chosen so that Kruskal's result applies to these matrices, and that identifiability for the continuous model can be linked to identifiability of the discrete one. This requires equivalence between linear independence of probability distributions and their corresponding c.d.f.s.

Finally, the case of multidimensional blocks of conditionally independent measures, model (3), is covered using a similar but more cumbersome construction (Theorem 9 in Allman et al., 2009). Discrete random variables Y_k's are defined based on indicator functions of d_l-product intervals, where d_l is the lth block dimension. The equivalence between linear independence of the probability distributions and corresponding multidimensional c.d.f.'s remains valid, so that model (3) is identifiable in general.

3 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). 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 whole 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 where this algorithm becomes "EM-like", since kernel density estimation is not a genuine maximization step.

3.1 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 $j = 1, \ldots, m$:

$$p_{ij}^{(t)} := \mathbb{P}_{\boldsymbol{\theta}^{(t)}}(Z_{ij} = 1 | \boldsymbol{x}_i) = \frac{\lambda_j^{(t)} f_j^{(t)}(\boldsymbol{x}_i)}{\sum_{j'=1}^{m} \lambda_{j'}^{(t)} f_{j'}^{(t)}(\boldsymbol{x}_i)}, \tag{4}$$

   where $f_j^{(t)}(\boldsymbol{x}_i) = \prod_{l=1}^{B} f_{jl}^{(t)}(x_{is_l})$.

2. **M-step for λ:**

$$\lambda_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{(t)}, \quad j = 1, \ldots, m. \tag{5}$$

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

$$f_{jl}^{(t+1)}(\boldsymbol{u}) = \frac{1}{n\lambda_j^{(t+1)}} \sum_{i=1}^{n} p_{ij}^{(t)} K_{H_{jl}}(\boldsymbol{u} - x_{is_l}), \tag{6}$$

   where $K_{H_{jl}}$ is a multivariate kernel density function, typically Gaussian, and H_{jl} is a symmetric positive definite $d_l \times d_l$ 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.

3.2 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. Firstly, as in Benaglia et al. (2009a) it is possible to 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). Secondly, we 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).

Multivariate Kernel Density Estimation (KDE) has been used since a long time in multivariate data analysis (see, e.g., Scott, 1992). Forgetting for now about blocks and components, and 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

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

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

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

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, $H = \mathrm{diag}(h_1^2, h_2^2, \ldots, h_r^2)$ where h_k denotes the kth coordinate bandwidth. Then $|H|^{1/2} = h_1 \cdots h_r$ so that (denoting informally by K both the multivariate and univariate kernels)

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

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

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

In our mixture model with multivariate blocks, we propose to consider two cases for the $d_l \times d_l$ diagonal bandwidth matrix associated to 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: $H_l = \mathrm{diagonal}(\boldsymbol{h}_{s_l}^2)$, where $\boldsymbol{h}_{s_l} = (h_k)_{k \in s_l}$. The multivariate kernel for block l becomes

$$K_{H_l}(\boldsymbol{u}) = \frac{1}{\prod_{k \in s_l} h_k} K(H_l^{-1/2}.\boldsymbol{u}), \quad \boldsymbol{u} \in \mathbb{R}^{d_l},$$

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

$$h_k = 0.9 \min\{SD_k, \frac{IQR_k}{1.34}\}(n)^{-1/5}, \tag{7}$$

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 scalar bandwidths for each coordinates in the block, but it depends also on component j and current algorithm iteration t:

$$H_{jl}^{(t)} = \mathrm{diagonal}((\boldsymbol{h}_{js_l}^{(t)})^2), \quad \text{where } \boldsymbol{h}_{js_l}^{(t)} = (h_{jk}^{(t)})_{k \in s_l}.$$

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

$$K_{H_{jl}^{(t)}}(\boldsymbol{u}) = \frac{1}{\prod_{k \in s_l} h_{jk}^{(t)}} K\left((H_{jl}^{(t)})^{-1/2}.\boldsymbol{u}\right), \quad \boldsymbol{u} \in \mathbb{R}^{d_l}.$$

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 has 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:

$$h_{jk}^{(t+1)} = 0.9\min\{\sigma_{jk}^{(t+1)}, \frac{IQR_{jk}^{(t+1)}}{1.34}\}(n\lambda_j^{(t+1)})^{-1/5}, \tag{8}$$

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

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

are the weighted empirical means and variances.

To define the iterative interquartile range $IQR_{jk}^{(t+1)}$ appearing in (8), 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 functions for computing these quantiles are provided in the mixtools package (Benaglia et al., 2009b).

4 Implementation and simulated examples

We propose in this section some examples illustrating the performances of our algorithm, on two synthetic multivariate models, after some details about implementation and experiment settings. The `mvnpEM` algorithm defined in Section 3.1 has been implemented in the development version of the mixtools package Benaglia et al. (2009b) for the R statistical software R Development Core Team (2010), and will be made publicly available in a future version of it. In particular, the step requiring nonparametric multivariate WKDE's has been coded in `C` to speed up the CPU time.

4.1 Initialization of the mvnpEM algorithm.

As it is typically the case 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 location of 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 assign each observation to one initial components as, e.g., in Benaglia et al. (2009a). At this point, the parallel 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. We never experienced any difficulties (such as, e.g., the algorithm emptying one component after few iterations due to a very poor initialization). Note also that this k-means based initialization is also often used in standard EM algorithms for, e.g., multivariate Gaussian mixtures, where clusters means and (co)-variances are used as initialization means and variances for the component Gaussian distributions.

In even more complex situations where the above initialization strategies fail we can proceed by analogy with the parametric space exploration: drawing $\boldsymbol{P}^{(0)}$ posterior matrices randomly (uniformly) several times, and run several `mvnpEM` algorithm 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 ModelsA and B, and obtained the same results as with the k-means initialization.

Handling the label-switching problem Not surprisingly, the data-driven initialization without specifying centers to the k-means procedure generates more label-switching than when proper centers are provided. As explained in Section 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 component index does not change 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 $\hat{\lambda}_1 < \hat{\lambda}_2$. Here, we choose to detect and "switch-back" the estimates (the final matrix of posteriors here, from which the other estimates are computed) to be in accordance with the initial representation. Since in all our experiments we set $\lambda_1 < \lambda_2$, we decide that a switching occured after a replication if $\hat{\lambda}_1 > \hat{\lambda}_2$, in which case we switch the parameters from 1^{st} component to 2^{nd} component and inversely.

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):

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

where the integral over $\mathbb{R}^{d_l}$ is computed numerically and $\hat{f}_{jl}^{(s)}$ is the density estimate at replication s, computed from (6) 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.

A difference with both Hall et al. (2005) and Benaglia et al. (2009a) results is that in their work the Integrated Squared Errors $ISE_{jl} = \int(\hat{f}_{jl} - f_{jl})^2$ were evaluated using numerical integrations of univariate densities (since the f_{jl}'s were univariate only). Here, it appears that estimating f_{jl} for multivariate densities with strong dependence structure using a kernel density estimate (KDE) with diagonal bandwidth matrix is more difficult, and this difficulty may results 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 (7). Results are in Table 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 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 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 $MISE_{jl}$'s between univariate and multivariate blocks, we selected variances matrices Σ_j's with not too strong correlations (up to 50%).

We also computed the mean squared error (MSE) for the proportions that are the only scalar parameters in these models. In our models with just $m = 2$ components, we have for

λ_1:

$$MSE_{\lambda_1} = \frac{1}{S}\sum_{s=1}^{S}(\hat{\lambda}_1^{(s)} - \lambda_1)^2,$$

where, at replication s, $\hat{\lambda}_1^{(s)}$ is computed using (5) 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

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

4.2 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) (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 1 multivariate block, i.e. $B = 3$ blocks of coordinates with $s_1 = \{1\}, s_2 = \{2\}, s_3 = \{3,4\}$. Densities f_{jl} are univariate normals for $l = 1, 2$, and bivariate Gaussian for block $l = 3$, where the means are given in Table 2, and the common covariance matrix of the bivariate block is

$$\Sigma = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}.$$

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

Table 2: Parameters for Model A.

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.

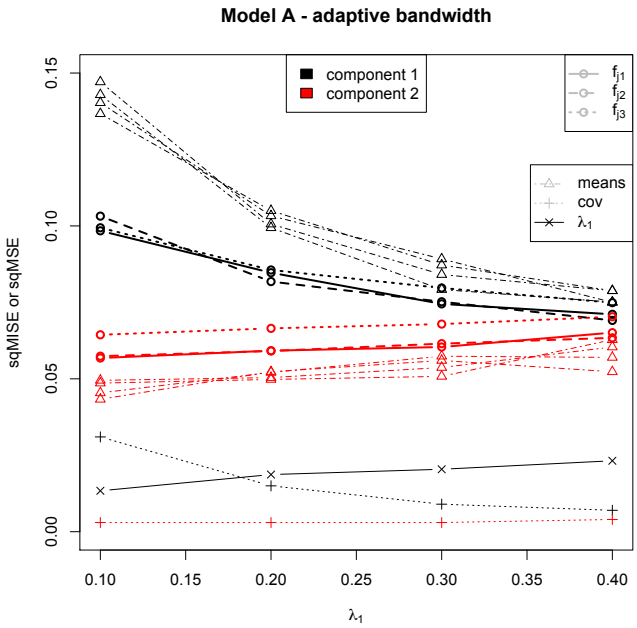

Figure 1: Square roots of MISE for the densities and square roots of MSE for the scalar parameter λ_1, and other scalar measures that are not parameters in the model (means and covariances), as a function of the proportion of the first component λ_1, for Model A, $n = 500$ and $S = 300$ replications, adaptive bandwidth. The gray line types in the legend are identifying densities and scalar criterions, that are plotted colored by component.

Results for model A ran with the adaptive bandwidth strategy are given in Fig. 1. We obtained similar results with the same bandwidth setting; these results are omitted here for brevity. 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 4.1, 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 estimates associated to component 1 (black curves) decrease when λ_1 increases, as expected since the proportion of data actually coming from this component increases with λ_1. Simultaneously, the estimates associated with component 2 increase (red curves). Moreover, the results for the $\sqrt{MISE}_{f_{jl}}$'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.

4.3 Model B: Gaussian, heavy-tailed and skewed data

We also experiment our method on a second model, with three bivariate blocks using the full potential of our approach. We wanted here to show that our algorithm can compete to some extent with fitting Gaussian mixtures 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, $m = 2$ components, where λ_1 is kept fixed to 30 %, 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 $t_2(\mu_{j2}, \Sigma)$'s with same correlation structure. The component densities of block 3 are themselves mixtures of bivariate Gaussian contaminated by bivariate Student's, thus generating skewed densities. This model involves two covariance matrices,

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

where Σ' is used only in block 3, component 2. The other parameters are given in Table 3.

Model B	Block 1	Block 2	Block 3
s_l	$\{1, 2\}$	$\{3, 4\}$	$\{5, 6\}$
$j = 1$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \Sigma\right)$	$t_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \Sigma, df = 4\right)$	$87\% t_2\left(\begin{bmatrix}0\\0\end{bmatrix}, \Sigma, df = 4\right) + 13\% \mathcal{N}_2\left(\begin{bmatrix}4\\6\end{bmatrix}, \Sigma\right)$
$j = 2$	$\mathcal{N}_2\left(\begin{bmatrix}1\\5\end{bmatrix}, \Sigma\right)$	$t_2\left(\begin{bmatrix}2\\5\end{bmatrix}, \Sigma, df = 4\right)$	$87\% t_2\left(\begin{bmatrix}2\\8\end{bmatrix}, \Sigma, df = 4\right) + 13\% \mathcal{N}_2\left(\begin{bmatrix}5\\14\end{bmatrix}, \Sigma'\right)$

Table 3: Parameters for Model B.

Before presenting a full Monte-Carlo experiment as for model A, we display in Figure 2 the true marginal densities of this model, together with a result from a single run of the `mvnpEM` algorithm and a result given by a standard Gaussian EM algorithm, the `mvnormalmixEM` function from the mixtools package Benaglia et al. (2009b).

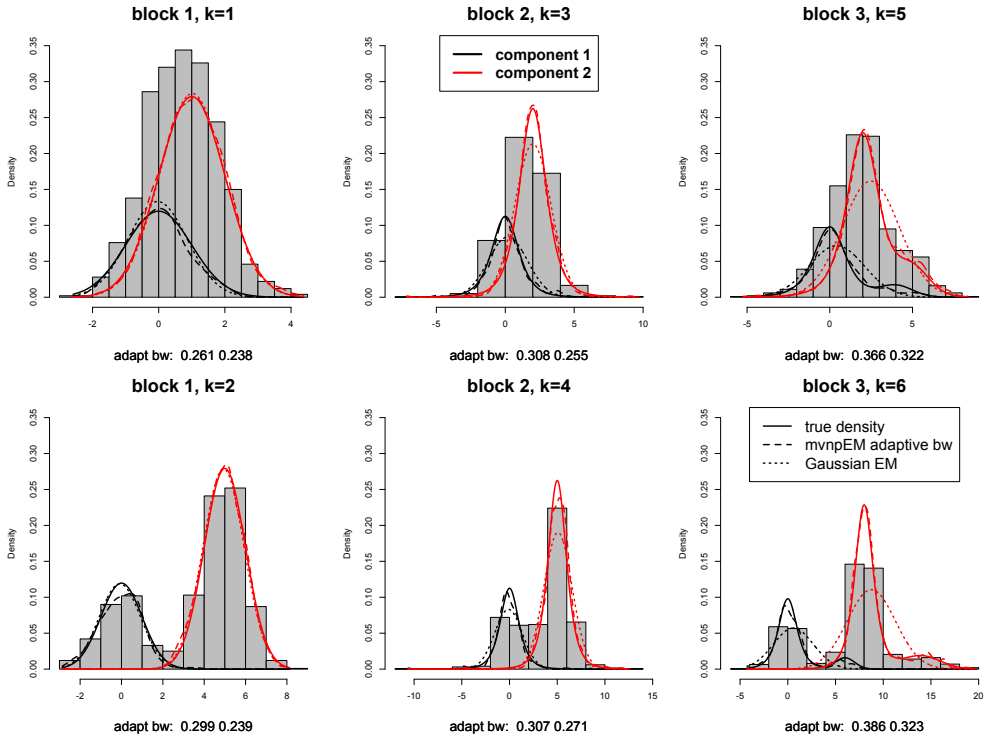

Figure 2: Marginal density estimates for Model B, where column l corresponds to the two marginals of the lth bivariate block, $l = 1, 2, 3$. Each plot shows the true marginals (solid lines), `mvnpEM` with adaptive bandwidth estimates (dashed lines), Gaussian EM estimates (dotted lines). Estimates are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

The estimate for the proportion in this run has been $\hat{\lambda}_1 = 0.284$ for our nonparametric method, and 0.282 for the Gaussian EM. We also computed the estimates of the covariances for block 1 (the one for which the true values are given in Σ). Both `mvnpEM` and the Gaussian EM gave estimates of about 0.236 in both components.

We then ran $S = 300$ replications of samples of sizes $n = 400, 600, 800, 1000$. As for model A, we computed the $MISE$ of the densities and the mean squared error (MSE) of the scalar parameter (λ_1) and some other descriptive scalar measures (for the Gaussian block 1 only).

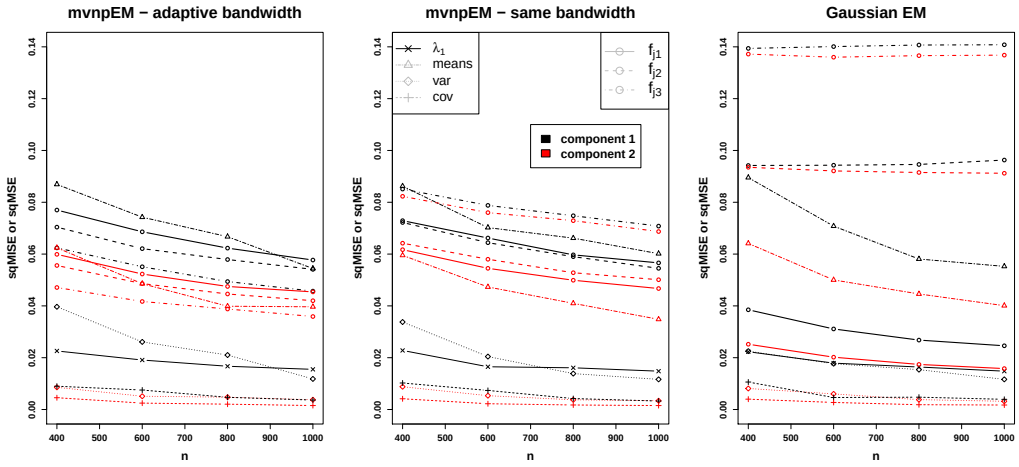

Figure 3: Square roots of MISE's for the densities as a function of the sample size n, $S = 300$ replications, for the two algorithm settings for Model B: `mvnpEM` adaptive bandwidth (left), `mvnpEM` same bandwidth (middle) and Gaussian EM (right). MISE for Densities are plotted in circles and solid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

Our purpose was also to build a model illustrating the performance of the adaptive bandwidth strategy (Section 3.2), which is appropriate typically for models with different ranges of observations per components and coordinates. Figure 3 confirm this behavior, with a slight advantage for the adaptive bandwidth, particularly in block 3, as expected. Figure 3 also shows that the MISE's decrease when the sample size n increases, which can be understood as numerical evidence of "convergence" (see the Discussion about these convergence-related questions). Finally we can see, by comparing marginal densities in the single example Fig. 2 and MISE's Fig. 3 across the nonparametric and parametric solutions block per block, that our method performs slightly less better than the Gaussian method for the Gaussian block 1, but better than it for the heavy-tailed block 2, and 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.

5 An example on actual data

We consider in this section a real dataset from an experiment involving $n = 569$ instances about Wisconsin Diagnostic Breast Cancer (WDBC). This database is available through the UW CS ftp server. The attributes in the WDBC dataset are the Diagnosis (M = malignant, B = benign) and 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.

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 compare these estimates with the observed classes. For model-based clustering

using mixture models, the clustering 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

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

where the $\hat{p}_{ij}$'s are as before the posterior probabilities after convergence of the algorithm. The MAP classifier is compared here with the classes given by Diagnosis variable (62.74% B and 37.26% M).

Our motivation in using this dataset is not to find a scientific definitive answer or the best clustering algorithm. We have choosen 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. 4 below); (iii) it has been used recently in Hennig (2010), who propose 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 first features (means) of the WDBC dataset. Hence we first tried our approach on this $r = 10$ dimensional dataset. We had to define multivariate conditionally independent blocks prior to apply our `mvnpEM` algorithm. A simple exploration of the data shows that there are some obvious correlations across coordinates, not due to a mixture. Fig. 4 displays the most obvious such dependences 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.

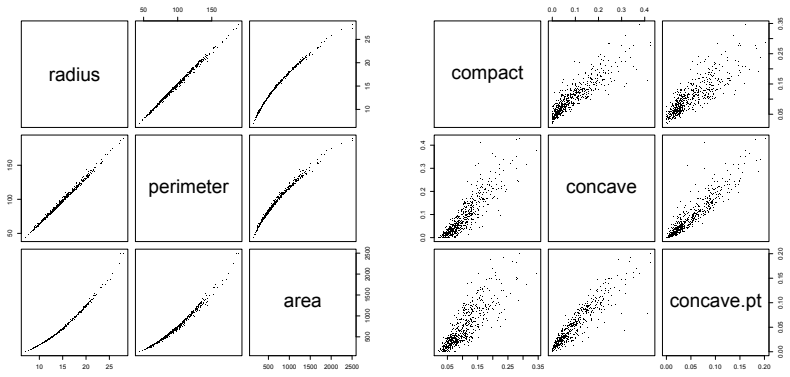

Figure 4: Pairs plots for selected "mean" features from the WDBC database; $s_1 = \{1, 3, 4\}$ for block 1 (left), and $s_2 = \{6, 7, 8\}$ for block 2 (right).

Proceeding like this, we are able to design some plausible models. One of the best ones in terms of clustering precision is made of $B = 5$ blocks: the two trivariate blocks from Fig. 4, a block of size 2 (symmetry and fractal dimension), and two remaining blocks of size 1. The results are given in Table 4, together with the basic k-means that we tried as well (and that is used in our initialization of the `mvnpEM`, see Section 4.1).

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 4: 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).

In experimenting some alternative block designs we somehow proceed like Hennig (2010) who try several (heuristical) merging criterions and reports the best result they obtained. However, in our case, these alternative models (e.g., merging smoothness with the block in Fig. 4, right) always showed results between 92.5 and 93.67% i.e. better than Hennig (2010).

We also tried more complex models by adding the other groups of available features, the 10 standard errors (se), or the 10 "worst" variables keeping the same block structure (also supported by the exploration of the scatterplots) but merging together corresponding features. 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 means and se's) 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 model and $n = 569$ individuals only took a couple of minute on a common laptop computer.

6 Discussion

We have proposed in this paper 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 (covariance) 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). We have also introduced and described two strategies to select the bandwidth involved in kernel density estimation step of this algorithm. The performance of this model have been evaluated through numerical simulations with two perspectives. First we experimented it focusing on parameter estimation (including the nonparametric multivariate densities), on two synthetic models: one allowing for comparison with the original `npEM` algorithm and results from Hall et al. (2005) based on an inversion method (both designed for univariate blocks only); the other showing how

our algorithm behaves on Gaussian, non-Gaussian with heavy tails, and non-Gaussian with both heavy tails and severely skewed data. We also showed there that some better estimates can result from the adaptive bandwidth strategy we have introduced, compared to the more immediate fixed bandwidth approach.

Second, we have 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 showed that, for several possible blocks design, a clustering based on the Maximum A Posteriori (MAP) strategy using final posterior matrix produced by our algorithm outperformed the two other approaches. The purpose of this example was also 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).

Our algorithm, like the original `npEM` algorithm for univariate blocks from Benaglia et al. (2009a) has not yet theoretical justification, since it is not proved to maximize any objective function, and since 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 the scalar and density estimates "numerically converge" to the true values for MSE or MISE criterion, when we let n increase, for all the models we tried. An ongoing work that is beyond the scope of the present paper is precisely to show some type of convergence, extending the ideas from Levine et al. (2011) introducing a non-linearly smoothed log-likelihood objective function and developping an iterative algorithm with a monotony property as for a genuine EM.

Finally, we reiterate that the `mvnpEM` algorithm introduced in this work will be publicly available in a future version of the mixtools package Benaglia et al. (2009b) for the R statistical software R Development Core Team (2010).

Acknowledgements We thank the Associate Editor and a Reviewer for pointing us to references Hennig (2010) and Hunt and Jorgensen (2003), and for their valuable comments helping us improving the original manuscript and the simulation and real data sections.

References

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*, pages 15–27. Number Nonparametric Statistics and Mixture Models: A Festschrift in Honor of Thomas P. Hettmansperger. 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.

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., Mottelet, S., and Vandekerkhove, P. (2006). Semiparametric estimation of a two-component mixture model. *Annals of Statistics*, 34(3):1204–1232.

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.

Chauveau, D., Hunter, D. R., and Levine, M. (2015). Semi-parametric estimation for conditional independence multivariate finite mixture models. *Statist. Surv.*, 9:1–31.

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.

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.

Hunt, L. and Jorgensen, M. (2003). Mixture model clustering for mixed data with missing information. *Comput. Stat. Data Anal.*, 41(3-4):429–440.

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.

Kruskal, J. B. (1976). More factors than subjects, tests and treatments: an indeterminacy theorem for canonical decomposition and individual differences scaling. *Psychometrika*, 41(3):281–293.

Kruskal, J. B. (1977). Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138.

Levine, M., Hunter, D. R., and Chauveau, D. (2011). Maximum smoothed likelihood for multivariate mixtures. *Biometrika*, 98(2):403–416.

Lindsay, B. G. (1995). *Mixture Models: Theory, Geometry, and Applications*. Ims.

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 Krishnan, T. (2008). *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics: Applied Probability and Statistics. Wiley-Interscience, New York.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

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.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability. Chapman & Hall, London.