HAL open science

Nonparametric mixture models with conditionally independent multivariate component densities

Didier Chauveau, Vy Thuy Lynh Hoang

▶ To cite this version:

Didier Chauveau, Vy Thuy Lynh Hoang. Nonparametric mixture models with conditionally independent multivariate component densities. 2014. hal-01094837v1

HAL Id: hal-01094837

https://hal.science/hal-01094837v1

Preprint submitted on 13 Dec 2014 (v1), last revised 30 Sep 2015 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Nonparametric mixture models with conditionally independent multivariate component densities

Didier Chauveau Hoang Vy Thuy Lynh

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

December 11, 2014

Abstract

Recent works in the literature have proposed models and algorithms for nonparametric estimation of finite multivariate mixtures. In these works, the model assumes independent coordinates, conditional on the subpopulation from which each observation is drawn, so that the dependence structure comes only from the mixture. Here, we relax this assumption, allowing in the multivariate observations independent multivariate blocks of coordinates conditional upon knowing which mixture component from which they come. Otherwise their 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 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.

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 (d-dimensional) Euclidean 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 paper of Dempster et al. (1977), is more properly understood to be a class of algorithms, a number of which predate even the Dempster et al. (1977) paper 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 of the complete data, which is itself 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

$$\mathbb{P}_{\boldsymbol{\theta}}(Z_{ij} = 1) = \lambda_j, \quad \text{and}(\boldsymbol{X}_i|Z_{ij} = 1) \sim f_j, \quad j = 1, ..., m.$$

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 d-dimensional Euclidean 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 Euclidean 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 an Euclidean 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 Euclidean 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. (2014) 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, nowaday 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 `npEM` algorithm is publicly available in the mixtools package (Benaglia et al., 2009b) for the R statistical software R Core Team (2013), 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. (2012) 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* 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). 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

of Allman et al. (2009) result implying that there is little hope to have an identifiable model for less than 3 independent blocks (see 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 to 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).

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. (2014). 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.

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. (2014).

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 $\lambda_j \stackrel{\text{def}}{:=} P(Z = j)$, 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 Euclidean part 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 $\mathbb{R}^d$, 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 $\boldsymbol{u} = (u_1, u_2, \ldots, u_d)^t \in \mathbb{R}^d$,

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

K is a multivariate kernel function, H is a symmetric positive definite $d \times d$ "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, $H = \mathrm{diag}(h_1^2, h_2^2, \ldots, h_d^2)$ where h_k denotes the kth coordinate bandwidth. Then $|H|^{1/2} = h_1 \cdots h_d$ so that (denoting informally by K both the multivariate and univariate kernels)

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

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

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

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

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

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

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 Core Team, 2013), 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.

We propose in this section some examples illustrating the performances of our algorithm, on two synthetic multivariate models. We first introduce a simple model (model A) with two univariate blocks and one bivariate block. This model is interesting since it is as close as possible to model (2) with conditionally independent marginals. Hence, it allows comparison with the original `npEM` algorithm, and Hall et al. (2005) results based on their estimation approach for the two univariate blocks. It also allows us to compare the estimation quality between these univariate blocks and the bivariate one. Then we introduce a second model, model B, with three bivariate blocks using the full potential of model (3).

Model A: 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 $\mathcal{N}(\mu_{jl}, 1)$ for the univariate blocks, and $\mathcal{N}_2(\mu_{j3}, \Sigma)$ for bivariate block $l = 3$, where the means are given in Table 1, and the common covariance matrix 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 1: Model A

The proportion λ_1 of the first component will vary between 0.1 and 0.4 in the experiment concerning this model. Note that model A is intentionally very similar to the Gaussian trivariate example used first by Hall et al. (2005) to illustrate the performance of their estimation technique based on inverting the mixture, and then used for comparison with the `npEM` estimates in Benaglia et al. (2009a). This model 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. Model A presents however not so well-separated distributions across components than Hall et al. (2005) model.

Model B: This is a higher dimensional model with $r = 6$ variables and $m = 2$ components, where λ_1 =30 % for the proportion of the first component. There are 3 blocks of bivariate Gaussian densities with correlation structure $\mathcal{N}_2(\mu_{jl}, \Sigma_{jl})$, where component means are given in Table 2. This model involves two covariance matrices,

$$\Sigma = \begin{pmatrix} 1 & 1/4 \\ 1/4 & 1 \end{pmatrix}, \quad \text{and} \quad \Sigma_{23} = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 4 \end{pmatrix}.$$

All three blocks of the first component and the first two blocks of the second component share the same covariance $\Sigma_{1l} = \Sigma_{21} = \Sigma_{22} = \Sigma$, whereas Σ_{23}

is used only for the second component, third block. Figure 2 displays the marginal densities for this model, together with a result form a single run of the `mvnpEM` algorithm.

Model B	Block 1	Block 2	Block 3
Coordinates	$\{1,2\}$	$\{3,4\}$	$\{5,6\}$
Component 1	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix},\Sigma\right)$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix},\Sigma\right)$	$\mathcal{N}_2\left(\begin{bmatrix}0\\0\end{bmatrix},\Sigma\right)$
Component 2	$\mathcal{N}_2\left(\begin{bmatrix}1\\1\end{bmatrix},\Sigma\right)$	$\mathcal{N}_2\left(\begin{bmatrix}8\\8\end{bmatrix},\Sigma\right)$	$\mathcal{N}_2\left(\begin{bmatrix}8\\8\end{bmatrix},\Sigma_{23}\right)$

Table 2: Model B

Initialization of the mvnpEM algorithm. To initialize the algorithm, the first E-step requires initial values for the $f_j^{(0)}$'s which themselves require an initial $n \times m$ matrix of posteriors $(p_{ij}^{(0)})$. To obtain this matrix, it is possible to use a k-means clustering algorithm to assign each observation to one initial components. To get a first clustering of the data, we used in our experiments two completely opposite strategies, depending on wether or not we provide k-means with initial clusters centers. Our first setting is " true initialization" i.e. using the true means as the centers for the k-means preliminary binning of the data. The second setting is a data-driven automatic initialization, just specifying the number of clusters (m), so that m random data points are chosen as the initial centers within the kmeans procedure (denoted random initialization hereafter).

Handling the label-switching problem Not surprisingly, the data-driven initialization without specifying centers to the k-means procedure generates more label-switching than the `mvnpEM` true strategy. 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 1st component to 2nd component and inversely.

Experiment on model A: For model A, we ran $S = 300$ replications of $n = 500$ observations each, with proportions $\lambda_1 \in \{0.1, 0.2, 0.3, 0.4\}$. 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):

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

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 Hall et al. (2005) results is that in their work as well as in the comparisons in Benaglia et al. (2009a), the Integrated Squared Errors $ISE_{jl} = \int(\hat{f}_{jl} - f_{jl})^2$ were evaluated using numerical integrations of univariate densities. Here, it appears that estimating $\hat{f}_{jl}^{(s)}$ for strongly correlated blocks 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 3:

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: 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 or covariances (up to 50%). Also, to allow comparison with the original `npEM` and Hall et al. (2005) results for the univariate coordinates, we kept individual densities as in their examples for the first and the second block (see structure of Model A, Table 1).

We also computed the mean squared error (MSE) for the $m-1$ proportions that are the only scalar parameters in these models. In this case for λ_1, the proportion of component 1, we have:

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

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 display as well MSE's for other scalar measures of precision (means, variances,...) but these are not

genuine parameters of the model. 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

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

These statistics are provided in Fig. 1 and Fig. 3 as illustrative indicators of the algorithm efficiency.

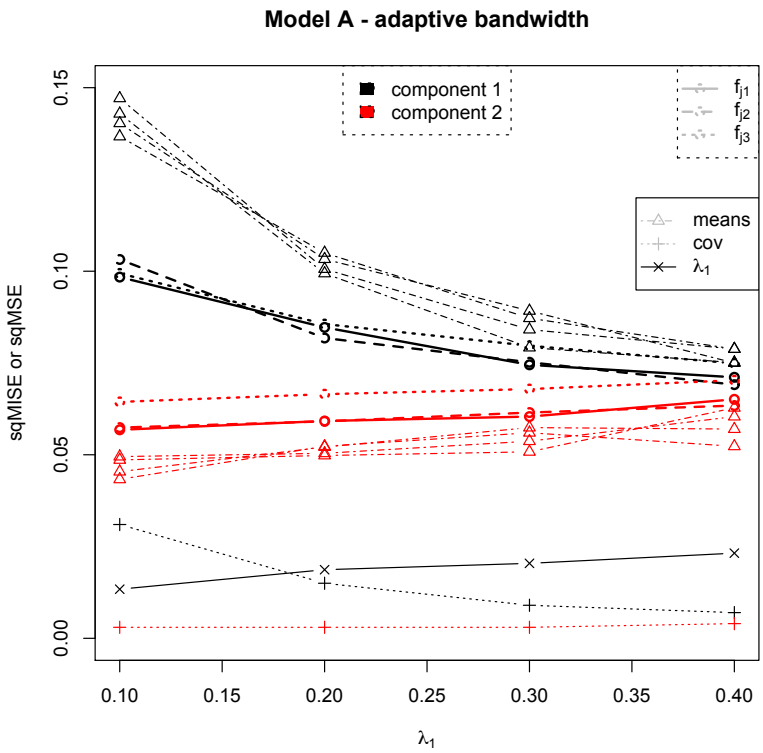

Figure 1: Square roots of MISE for the densities and square roots of MSE for the scalar parameter λ_1, and other scalar measures that are not parameters in the model (means and covariances), as a function of the proportion of the first component λ_1, for Model A, $n = 500$ and $S = 300$ replications, random initialization and adaptive bandwidth.

Results for model A ran with the adaptive bandwidth strategy are given in Fig. 1. We obtained similar results with the same bandwith 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 using the random initialization and checking for label switching. We also obtained similar results with true initialization but omit these results here as well since random initialization is what users do in practice.

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 the plots on page 675, figure 2 of Hall et al. (2005).

Experiment on model B: For model B, we choose three bivariate blocks, but with the same kind of constraints we detailed before on the correlations (see Table 3). Our purpose was also to build a model illustrating the performance of the adaptive bandwidth, which requires typically different ranges of observations per component/coordinates. This model has two "well-separated" blocks, one with different variance matrices, and one severely overlapping block (see Table 2).

Before providing a complete Monte-Carlo experiment as for model A, we start here by presenting in Fig. 2 a result of the algorithm's behavior on a single sample, illustrating the parameters and typical density estimates we can obtain in this case.

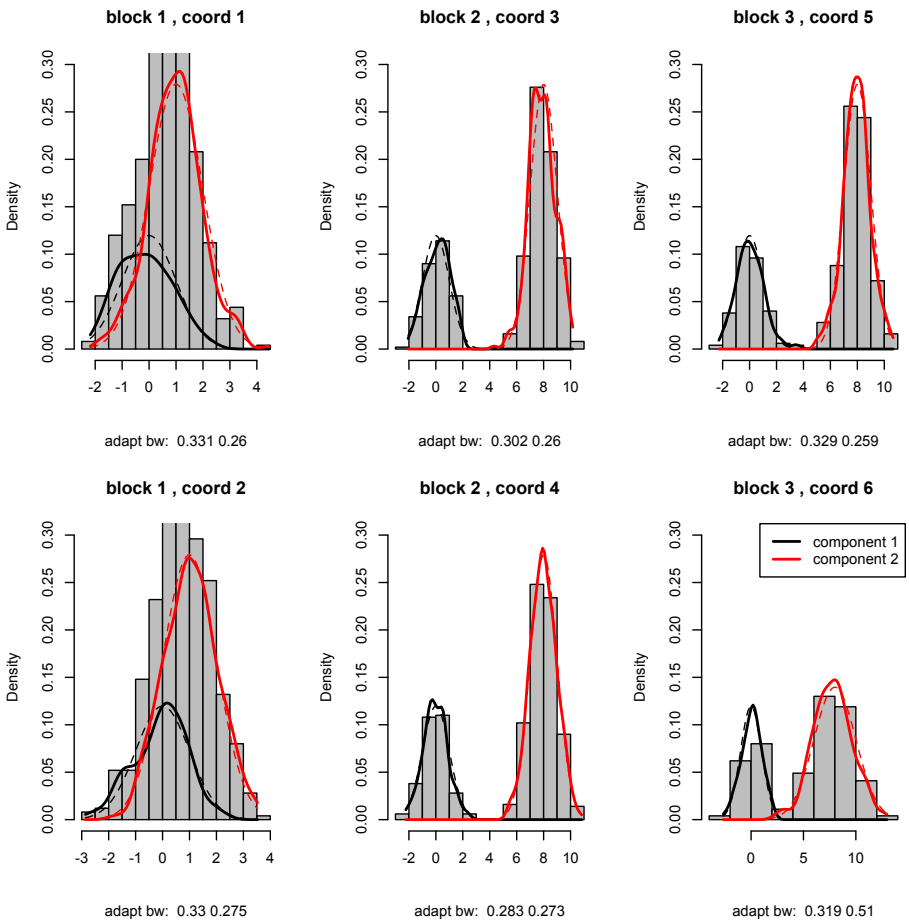

Figure 2: Example of marginal density estimates for Model B, as given by a single run of the `mvnpEM` algorithm for a sample of size $n = 500$, adaptive bandwidth strategy and random initialization. Column l corresponds to the two marginals of the lth bivariate block, $l = 1, 2, 3$. The true (normal) marginal densities are depicted in dashed lines.

We obtained an estimates of the proportion of the first component $\hat{\lambda}_1 = 0.310$ closed to the true value of $\lambda = 0.3$. Estimates of the covariances of the model are also displayed in Table 4. Remind here that only the true covariance for block 3, component 2 equals 0.5, while the others are equal to 0.25.

	Block 1	Block 2	Block 3
Component 1	0.2491	0.2356	0.3291
Component 2	0.2519	0.2502	0.5068

Table 4: The estimates of the covariances for all blocks and 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 the other scalar measures of precision for means and (co-)variances.

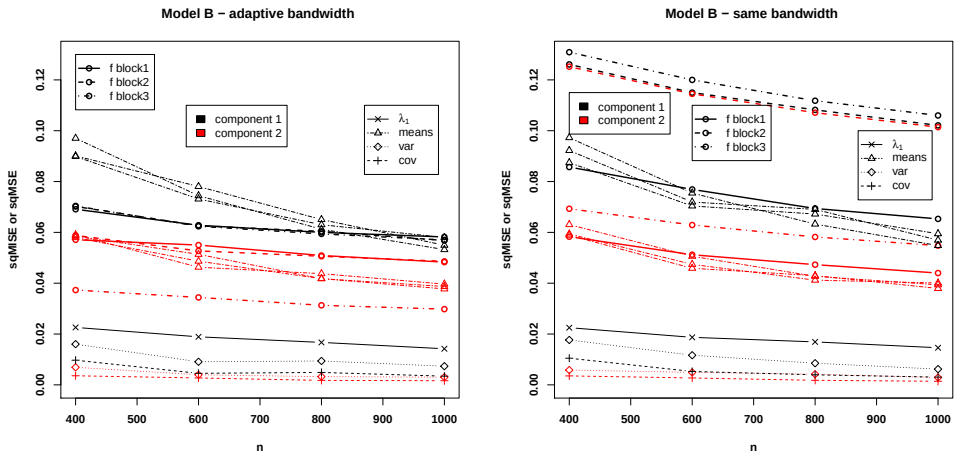

Figure 3: Square roots of MISE's for the densities and square roots of MSE's for the parameters as a function of the sample size n, for the two bandwidth settings for Model B, $S = 300$ replications, random initialization, and two methods for bandwidths settings: adaptive (left) and same (right).

Fig. 3 summarized the results for the two bandwidths tuning strategies presented in Section 3.2. Remember that in model B λ_1 is kept fixed to 0.3, and the horizontal axis shows the effect of increasing the sample size n. We can see that the adaptive bandwidth strategy is definitely better for the estimation of the component densities, than the fixed bandwidth strategy, especially for the well-separated blocks ($l = 2, 3$). In addition, all the MSE's and MISE's decrease when the sample size n increases, as expected.

5 An example on actual data

We consider in this section a real dataset from an experiment involving $n = 569$ instances of Wisconsin Diagnostic Breast Cancer (WDBC). This database is available through the UW CS ftp server[1]. The details of the attributes found

[1] `ftp.cs.wisc.edu`, see `math-prog/cpo-dataset/machine-learn/WDBC/`

in WDBC dataset are: ID number, 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 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. The total number of attributes is 32 (ID, diagnosis, 30 real-valued input features).

This actual dataset is sometimes used as a benchmark for comparing unsupervised clustering methods. As often, the principle of such a study consists in estimating the sub-population to which belongs each individual, and after that to compare these estimates with the true classes, i.e. the Diagnosis variable that is available here. Our motivation in using this dataset is not to find a scientific definitive answer or the best clustering algorithm, but rather to illustrate the potential usefulness of our model and show that our estimation algorithm is usable even for models involving blocks of moderate dimensions.

A mixture model with multivariate, conditionally independent blocks needs to be defined prior to apply the `mvnpEM` algorithm. It looks reasonable at first glance to assume that the ten features lead to conditionally independent responses so that the 30 coordinates corresponding to these features (from 3rd coordinate to 32th coordinate) could be organized into $B = 10$ blocks of dimension three each, i.e. $\{3, ..., 32\} = \bigcup_{l=1}^{10} s_l$, defined by $s_1 = \{3, 13, 23\}$, $s_2 = \{4, 14, 24\}, ..., s_{10} = \{12, 22, 32\}$.

However, from the description of the data we can see that the feature compactness is defined from two other features: the perimeter and the area ($compactness = perimeter^2/area - 1.0$). Therefore, to obtain a more reasonable model, we decided to group these three features in one block. Hence, we applied the `mvnpEM` algorithm to model (3) with $B = 8$ blocks (1 block of size 9 and 7 blocks of size 3), and $m = 2$ components (corresponding to the two possible Diagnosis: M = malignant and B = benign). Note that running the algorithm on that fairly high dimensional model and $n = 569$ individuals only took about 1 minute on a common laptop computer.

Maximum A Posteriori strategy and EM classifier In addition to the parameter estimates $\hat{\boldsymbol{\theta}}$, any EM-like algorithm can provide a classification of the individuals using a Maximum A Posteriori (MAP) strategy. This (unsupervised) clustering, which is the object of interest here, 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 previously noted the posterior probabilities after convergence of the algorithm. The MAP classifier is then compared with the true classes $\boldsymbol{Z}$ corresponding to the Diagnosis variable $\in \{B, M\}$. This true known classification gives 62.741% instances in the Benign group (B) and 37.259% instances in the Malignant group (M).

% classification	B (Benign)	M (Malignant)	Sum
Group (1)	62.214	9.842	72.056
Group (2)	0.527	27.417	27.944
Sum	62.741	37.259	100

Table 5: The cross-classification of the WDBC data using `mvnpEM` and the MAP strategy compared with the true classes.

As displayed in Table 5, we see that the MAP from the `mvnpEM` algorithm correctly classify 62.214% of the individuals of the whole dataset into component B and 27.417% into component M. 10.369% instances were incorrectly classified between the two components. We also indicate the true $\boldsymbol{Z}$ marginal (in the last row) and the MAP marginal distribution (the last column).

Let p denote the % of the correct classification. If there is no "label-switching", we have:

$$p = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{I}_{\{Z_{ij}=1 \text{ and } \hat{Z}_{ij}=1\}}$$

where $\hat{\boldsymbol{Z}}$ is the MAP estimate from the posterior probabilities after convergence. We computed the correct classification of the EM classifier p and the distribution of MAP given by each subpopulation; then we compared them with a k-means classification as in Table 6:

% correctly-classified	B (Benign)	M (Malignant)	$p(\%)$
`mvnpEM` MAP	99.16	73.585	89.631
k-means	99.72	61.321	85.413

Table 6: The % of correct classification of the WDBC data using `mvnpEM` and the MAP strategy and using k-means clustering.

The solution given by the `mvnpEM` using the MAP classifier is better than the solution provided by the k-means strategy: 89.631% of the individuals classified correctly into the two components, in comparison with 85.413%. The percentage of instances correctly classified within component B is similar for both strategies, whereas our mixture model approach outperforms the common k-means approach for recovering the individuals from component M.

6 Discussion

We have proposed in this paper a nonparametric mixture model with conditionally independent multivariate blocks. The crucial novelty of our model from a statistical modeling 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 conditional independence assumption has been introduced in several works in the literature. Allman et al. (2009) proved the strong condition for identifiability of the parameters regardless the number of components m. We used a more general result from Allman et al. (2009) proving that our model is also identifiable in their sense.

We then proposed a multivariate, `mvnpEM` EM-like algorithm for this model. 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 on two synthetic multivariate models: one allowing for comparison with the original `npEM` algorithm and Hall et al. (2005) results based on an inversion method (both designed for univariate blocks only); the other one illustrating some better estimates that can result from the adaptive bandwidth strategy we have introduced, compared to a more direct fixed bandwidth approach. These new model and algorithm have also been experimented on an actual dataset which corresponds to a mixture of two sub-populations related to the diagnostic of breast cancer. We obtained that an estimation of the component memberships based on the `mvnpEM` algorithm and MAP strategy behaves well, in comparison with a (model-free) k-means strategy. 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 adaptively weighted Silverman's rule. This rule is known to be somehow motivated by estimation of Gaussian-shaped (tails) distributions, which is not necessarily the case here. 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), provides numerical evidence of consistency but lacks any theoretical justification. In a recent work, Levine et al. (2012) has introduced a smoothed loglikelihood objective function and developped an iterative algorithm to correct this. Both algorithms also produced similar results in some test cases. This smoothed EM idea is the subject of an ongoing work that is beyond the scope of the present paper.

Note finally that the `mvnpEM` algorithm introduced in this work will be publicly available in a future update version of the mixtools package (Benaglia et al., 2009b) for the R statistical software R Core Team (2013).

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. (2014). Semi-parametric estimation for conditional independence multivariate finite mixture models. Preprint hal-00558834v2.

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.

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. (2012). Maximum smoothed likelihood for multivariate maximum. *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 Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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.