Constraining kernel estimators in semiparametric copula mixture models

Gildas Mazo, Yaroslav Averyanov

Constraining kernel estimators in semiparametric copula mixture models

Gildas Mazo[1] Yaroslav Averyanov[2]

[1]MaIAGE, INRA, Université Paris-Saclay, 78350, Jouy-en-Josas, France
[2]MODAL, Inria Lille Nord Europe, Lille, France

Abstract

This paper presents a novel algorithm for performing inference and/or clustering in semiparametric copula-based mixture models. The algorithm replaces the standard kernel density estimator by a weighted version that permits to take into account the constraints put on the underlying marginal densities. Lower misclassification error rates and better estimates are obtained on simulations. The pointwise consistency of the weighted kernel density estimator is established under an assumption on the rate of convergence of the sample maximum.

Keywords: copula; kernel; semiparametric; nonparametric; mixture model; clustering.

1 Introduction

In modern data science, the observations of heterogeneous clusters is not uncommon. An example is given in [11] where one can observe two heterogeneous clusters of data points described by blood pressure and medical costs. The first dimension has a skewed Gaussian distribution and the second a log-normal distribution. The first cluster has negative dependency and the second positive dependency. These data cannot be captured by the standard Gaussian mixture model. The Student-t mixture model [22][27] is not able to deal with heterogeneous clusters either.

Recently more flexible models have been considered. On the one hand, there are copula-based methods [19, 20]. Copula-based methods allow for a separate analysis of the marginals and the dependence structure. They have been successfully applied in Pattern Recognition [36], Machine Learning [35], Knowledge Discovery and Database Management [11]. Copulas allow for concatenating discrete and continuous data, too [25]. In this paper, we only consider continuous data.

On the other hand, there are nonparametric methods. Nonparametric methods do not need to pick parametric families for the component distributions (i.e., the distributions of the clusters) but at the cost of assuming independence within each component [5, 24]. In nonparametric mixture models, the parameters are probability density functions, which are estimated by kernel density estimators embedded in pseudo-EM algorithms [7].

In this paper, following the work in [26], we combine both the copula framework and nonparametric estimation into a single mixture model. This permits to capture a wide spectrum of dependence structures while avoiding the choice of setting up the parametric families for the marginals. However, there is an important difference between the model of [26] and ours. In the former, the distributions in the clusters were not allowed to vary in scale. In the latter, change in scale is possible. This additional degree of freedom induces a structural constraint on the component marginal densities of the mixture. The constraint is not satisfied by the kernel density estimator used in the algorithm in [26]. How can we take the constraint into account? Will the inference be improved? To answer the first question, we have built a random weighted kernel density estimator and proved its pointwise consistency. To answer the second, we compared the algorithms on simulated and real data.

The rest of this paper is as follows. We present the models in Section 2. The first part reviews the paradigms under which one can build mixture models (Gaussian, copula-based, nonparametric and semiparametric) and the second part presents the model of interest in this paper. We give the learning algorithms in Section 3. Section 4 contains the definition and the consistency result for the weighted kernel density estimator. This section is written in a generic framework and therefore can be read independently. Section 5 and Section 6 contain the simulation experiments and the real data analysis, respectively. A Summary closes the paper.

2 Four kinds of mixture models

2.1 A review of paradigms for mixture models

We consider mixture models of the form

$$f(x_1, \dots, x_d) = \sum_{z=1}^{K} \pi_z f_z(x_1, \dots, x_d), \tag{1}$$

where $\pi_1, \dots, \pi_K$ are the proportions of the K components (or clusters) and $f_1, \dots, f_K$ are the corresponding densities. The choice of the structure for the component densities f_z specifies the kind of mixture model.

The first kind of mixture model is as follows. One picks a multivariate parametric family for the component densities and estimate their parameters by maximum likelihood through an EM algorithm. In the majority of cases one usually picks the multivariate Gaussian family, or, perhaps, the multivariate Student-t family. Note that all coordinates of a vector of variables are distributed according to the same distribution up to their parameters. For instance, all the coordinates of a vector distributed according to a Gaussian mixture model are Gaussian. This is an homogeneity assumption. We refer to a standard textbook [27] for further details. We note that, to deal with the complex, high-dimensional and noisy data of modern science, reseachers build more sophisticated models [4, 8, 10, 28, 41]. However, the Gaussian distribution remains at the core of statistics and is often used as an important building block of those [4, 8, 10, 41]. The copula method, presented next, is an interesting alternative.

The second kind of mixture model arises when one chooses to use the copula decomposition for each of the component densities, that is, one writes

$$f_z(x_1, \ldots, x_d) = c_z(F_{1,z}(x_1), \ldots, F_{d,z}(x_d)) \prod_{j=1}^{d} f_{j,z}(x_j), \tag{2}$$

where c_z is the copula density corresponding to f_z and $f_{1,z}, \ldots, f_{d,z}$ are the marginals. Here $F_{1,z}, \ldots, F_{d,z}$ are the corresponding (cumulative) distribution functions. Sklar's theorem [29, 39] states that for any distribution function F_z with continuous marginals $F_{1,z}, \ldots, F_{d,z}$, there exists a function $C_z : [0,1]^d \to [0,1]$, called the copula, such that

$$F_z(x_1, \ldots, x_d) = C_z(F_{1,z}(x_1), \ldots, F_{d,z}(x_d)), \tag{3}$$

for any $(x_1, \ldots, x_d)$ in the domain of definition of F_z. The decomposition (2) follows from Sklar's theorem by differentiation. The copula C_z encodes the dependence structure of a random vector. One easily checks that C_z is the distribution function of the random vector $(F_{1,z}(X_1), \ldots, F_{d,z}(X_d))$ if F_z is the distribution function of $(X_1, \ldots, X_d)$. Copulas are typically parametrized by considering families of the form $\{C_z(\cdot, \ldots, \cdot; \theta_z),\, \theta_z\}$ for some parameters θ_z. An example is given in Section 5. If in (3) $C_z(u_1, \ldots, u_d) = u_1 \cdots u_d$, then $c_z = 1$ in (2). This means that the variables are independent conditionally on belonging to the cluster z. In copula-based models, one can choose different parametric families for the marginals within the same cluster but this heterogeneity property comes at a price. Indeed, the specification of all the parametric families (there are dK marginals) can be a daunting task. Estimation of copula-based mixture models can be performed by EM or EM-like algorithms [20].

The third kind of mixture model is of nonparametric flavor. In nonparametric mixture models, one assumes

$$f(x_1, \ldots, x_d) = \sum_{z=1}^{K} \pi_z \prod_{j=1}^{d} f_{j,z}(x_j).$$

That is, conditionally on the labels (i.e. conditionally on being in a certain cluster), the variables are assumed to be independent. But, in contrast to copula-based mixture models, one does not assume parametric marginals. Nonparametric estimation can be performed with kernel density estimators embedded in EM-like algorithms [5]. In [5], marginals of the form

$$f_{j,z}(x_j) = \frac{1}{\sigma_{j,z}} g_j \left(\frac{x_j - \mu_{j,z}}{\sigma_{j,z}} \right), \tag{4}$$

where $\mu_{j,z}$ and $\sigma_{j,z}$ are location and scale parameters, respectively, are also considered. The case $\sigma_{j,z} = 1$ and $d = 1$ was considered in [7]. This work largely inspired further work on nonparametric mixture models from the kernel density estimation viewpoint. But nonparametric maximum likelihood estimation is also possible if one assumes log-concavity of the component densities [18].

The fourth kind of mixture model combines nonparametric estimation and copula modeling [26]. It is of the form (1), (2) and (4). In (2), the distribution functions $F_{j,z}$ are given by $F_{j,z}(x_j) = G_j((x_j - \mu_{j,z})/\sigma_{j,z})$ and

$$G_j(x_j) = \int_{-\infty}^{x_j} g_j(t)\, \mathrm{d}t. \tag{5}$$

The model [26] is a particular case where $\sigma_{j,z} = 1$. The g_j (hereafter called the *generators*) are estimated in a nonparametric way but the copula is entirely parametric, thus the term semiparametric used for this kind of models. Inference can be performed with essentially the same algorithms as in [5, 7] but with an additional step for estimating the copula parameters. Algorithm 1 in Section 3 is an example of such algorithms.

2.2 The model of interest

We consider a model of the fourth kind, a so called location-scale semiparametric copula-based mixture model of the form

$$f(x_1,\dots,x_d;\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\sigma},\mathbf{g},\boldsymbol{\theta}) = \sum_{z=1}^{K}\pi_z c_z\left(G_1\left(\frac{x_1-\mu_{1,z}}{\sigma_{1,z}}\right),\dots,G_d\left(\frac{x_d-\mu_{d,z}}{\sigma_{d,z}}\right);\theta_z\right) \times \prod_{j=1}^{d}\frac{1}{\sigma_{j,z}}g_j\left(\frac{x_j-\mu_{j,z}}{\sigma_{j,z}}\right),$$

that is, of the form (1), (2) and (4) where the generators g_j, $j = 1,\dots,d$, satisfy

$$\int x_j g_j(x_j)\,\mathrm{d}x_j = 0 \tag{6}$$

and

$$\int x_j^2 g_j(x_j)\,\mathrm{d}x_j = 1. \tag{7}$$

Here $\boldsymbol{\pi} = \{\pi_z\}$, $\boldsymbol{\mu} = \{\mu_{j,z}\}$, $\boldsymbol{\sigma} = \{\sigma_{j,z}\}$, $\mathbf{g} = \{g_j\}$, $\boldsymbol{\theta} = \{\theta_z\}$, $j = 1,\dots,d$, $z = 1,\dots,K$, are the parameters of the model (note that θ_z can be a multivariate parameter). Note that there is no loss of generality in assuming a unit variance in (7). Indeed, if the variance would be σ_j^2, say, then we could find a unique reparametrization (given by $\tilde{g}_j(x_j) = \sigma_j g_j(\sigma_j x_j)$ and $\tilde{\sigma}_{j,z} = \sigma_j\sigma_{j,z}$) so that (7) would be true. The copulas are parametrized by vectors θ_z. No specific parametric families are assumed for the generators.

3 Estimation

Given the model of interest in Section 2.2, one needs to estimate the proportions π_z, locations $\mu_{j,z}$, scales $\sigma_{j,z}$, generators g_j and copulas parameters θ_z for $z = 1,\dots,K$ and $j = 1,\dots,d$. Note that the estimates of the distribution functions G_j can be computed through (5). The sample is denoted by $(x_1^{(i)},\dots,x_d^{(i)})$, $i = 1,\dots,n$. Two learning algorithms are presented in this section. Algorithm 1, is essentially the same as that in [26], which itself is inspired from the algorithms in [5, 7]. Hence we do not consider that Algorithm 1 is a contribution of the paper. The contribution is Algorithm 2.

3.1 Description of the learning algorithms

Building upon the work of [5, 7, 26], the most natural algorithm one can build is Algorithm 1. Algorithm 1 requires initial estimates $\pi_z^0, \mu_{j,z}^0, \sigma_{j,z}^0, g_j^0, \theta_z^0$ and then

Algorithm 1

Given initial estimates $\pi_z^0, \mu_{j,z}^0, \sigma_{j,z}^0, g_j^0, \theta_z^0$ and for $t = 1, 2, \ldots$ (until some stopping criterion has been reached), follow the steps below.

1. Compute (for $i = 1, \ldots, n$ and $z = 1, \ldots, K$)

$$w_{i,z}^t = \frac{\pi_z^t c_z \left\{ G_1^t \left(\frac{x_1^{(i)} - \mu_{1,z}^t}{\sigma_{1,z}^t} \right), \ldots, G_d^t \left(\frac{x_d^{(i)} - \mu_{d,z}^t}{\sigma_{d,z}^t} \right); \theta_z^t \right\} \prod_{j=1}^d \frac{1}{\sigma_{j,z}^t} g_j^t \left(\frac{x_j^{(i)} - \mu_{j,z}^t}{\sigma_{j,z}^t} \right)}{\sum_{z=1}^K \pi_z^t c_z \left\{ G_1^t \left(\frac{x_1^{(i)} - \mu_{1,z}^t}{\sigma_{1,z}^t} \right), \ldots, G_d^t \left(\frac{x_d^{(i)} - \mu_{d,z}^t}{\sigma_{d,z}^t} \right); \theta_z^t \right\} \prod_{j=1}^d \frac{1}{\sigma_{j,z}^t} g_j^t \left(\frac{x_j^{(i)} - \mu_{j,z}^t}{\sigma_{j,z}^t} \right)}$$

2. Process through the following steps ($j = 1, \ldots, d$, $z = 1, \ldots, K$).

   (a) Update the cluster proportions

$$\pi_z^{t+1} = \frac{1}{n} \sum_{i=1}^n w_{i,z}^t.$$

   (b) Update the location parameters

$$\mu_{j,z}^{t+1} = \frac{\sum_{i=1}^n x_j^{(i)} w_{i,z}^t}{\sum_{i=1}^n w_{i,z}^t}.$$

   (c) Update the scale parameters

$$(\sigma_{j,z}^{t+1})^2 = \frac{\sum_{i=1}^n (x_j^{(i)} - \mu_{j,z}^{t+1})^2 w_{i,z}^t}{\sum_{i=1}^n w_{i,z}^t}.$$

3. To update the generators, proceed through the following steps ($j = 1, \ldots, d$).

   (a) Generate a random variable $Z^{(i)}$ from Multi$(w_{i,1}^t, \ldots, w_{i,K}^t)$,

   (b) Define $\tilde{x}_j^i = (x_j^i - \mu_{j,Z^{(i)}}^t)/\sigma_{j,Z^{(i)}}^t$.

   (c) Update the generators

$$g_j^{t+1}(x_j) = \frac{1}{nh_j} \sum_{i=1}^n K \left(\frac{x_j - \tilde{x}_j^{(i)}}{h_j} \right) \tag{8}$$

4. Update the copula parameters ($z = 1, \ldots, K$)

$$\boldsymbol{\theta}_z^{t+1} = \arg\max_{\boldsymbol{\theta}_z} \sum_i w_{i,z}^t \log c_z \left\{ G_1^{t+1} \left(\frac{x_1^{(i)} - \mu_{1z}^{t+1}}{\sigma_{1,z}^{t+1}} \right), \ldots, G_d^{t+1} \left(\frac{x_d^{(i)} - \mu_{dz}^{t+1}{}^{t+1}}{\sigma_{d,z}} \right); \boldsymbol{\theta}_z \right\}$$

produces a sequence $\pi_z^t, \mu_{j,z}^t, \sigma_{j,z}^t, g_j^t, \theta_z^t$, for $t = 1, 2, \ldots$ until some stopping criterion has been reached. The first step is similar to the E step of any EM algorithm. The second step is also similar to the EM algorithm for Gaussian mixture models: the parameters are updated by computing weighted means where the weights $w_{i,z}^t$ relate the observations to their probabilities of belonging to the given clusters. The third step is similar to the computations undertaken in [7]. Given the data $x_j^{(i)}$ and given the weights computed at the t-th iteration, one generates a random label $Z^i \in \{1, \ldots, K\}$ according to a multinomial distribution $\mathrm{Multi}(w_{i,1}^t, \ldots, w_{i,d}^t)$. One then standardizes the data according to these simulated labels, that is, builds a pseudo-sample $\tilde{x}_j^{(1)}, \ldots, \tilde{x}_j^{(n)}$ and constructs a kernel density estimator on the top of it for updating the generators. The kernel density estimator can be constructed by following the guidelines as those in a standard textbook [38]. In (8), the kernel is denoted by K and the bandwidth by h_j. Thanks to a straightforward extension of Lemma 1 in [7], one has that, at each iteration t of the algorithm, $\tilde{x}_j^{(1)}, \ldots, \tilde{x}_j^{(n)}$ is a sample from g_j^t and therefore the choice of the bandwidth can be based on that sample. Finally in the last step, one maximizes a pseudo-likelihood for the copula parameters. See [26] for more details about this step. Algorithm 1 empirically has been found to perform well on simulations (see Section 5) whenever one is concerned with the estimation of the parameters for their own sake. However, when one is interested in the task of clustering instead, Algorithm 1 appears to have no greater value than a standard Gaussian mixture model. See Figure 1 and Section 5.

Interestingly, one can improve on Algorithm 1 by taking the inherent structure of the model into account. Note that in Algorithm 1 the estimator of the generators is not a generator itself. That is, (6) and (7) hold true but in general

$$\int x_j g_j^{t+1}(x_j)\,\mathrm{d}x_j = 0 \text{ and } \int x_j^2 g_j^{t+1}(x_j)\,\mathrm{d}x_j = 1 \tag{9}$$

do *not*. By letting the estimators g^t unconstrained in spite of (6) and (7), information may be lost. To overcome this problem, we propose to base inference on Algorithm 2. Algorithm 2 takes into account the inherent constraints of the model by replacing the standard kernel density estimator (8) by a weighted version (10) satisfying the constraints at each iteration of the algorithm. The proof of pointwise consistency of the weighted kernel density estimator are postponed to Section 4.

Algorithm 2 proceeds as follows. First one follows the instructions of Algorithm 1 till the construction of the pseudo-samples $\tilde{x}_j^{(i)}$. Then one solves an optimization problem for each marginal to get the weights of an adaptive kernel density estimator which, at each iteration of the algorithm, satisfies the constraints (9) (see Section 4). The optimization problem is convex and easy to solve. Consistency of the resulting estimator is studied in Section 4. Finally, once the marginals have been updated, a last step is added to estimate the copula parameters, as in Algorithm 1.

3.2 Heuristics underlying the learning algorithms

Increase of the log-likelihood

The two learning algorithms of Section 3.1 are designed to increase the log-likelihood of the data. They start as the standard EM algorithm. In particular,

Algorithm 2

1. Follow the steps 1 and 2 in Algorithm 1.
2. Generate the random labels $Z^{(i)} \sim$ Multi$(w^t_{i,1}, \ldots, w^t_{i,K})$ and build the pseudo-sample $\tilde{x}^i_j = (x^i_j - \mu^t_{j,Z^{(i)}})/\sigma^t_{j,Z^{(i)}}$ as in Algorithm 1.
3. Compute
$$\widehat{M}_{n,j} = \begin{pmatrix} 1 & \cdots & 1 \\ \tilde{x}^{(1)}_j & \cdots & \tilde{x}^{(n)}_j \\ [\tilde{x}^{(1)}_j]^2 & \cdots & [\tilde{x}^{(n)}_j]^2 \end{pmatrix}, \text{ and } \mathbf{b}_{n,j} = \begin{pmatrix} 1 \\ 0 \\ 1 - h^2_j \end{pmatrix}.$$
4. Solve the optimization problems
$$\min_{\mathbf{p}\in\mathbf{R}^n} \|\mathbf{p}\|^2_2$$
$$\text{such that } \begin{cases} \widehat{M}_{n,j}\mathbf{p} = \mathbf{b}_{n,j} \\ \mathbf{p} \geq \mathbf{0}, \end{cases}$$
and denote the solutions by $\tilde{\mathbf{p}}_j = (\tilde{p}^{(1)}_j, \ldots, \tilde{p}^{(n)}_j)$.
5. Follow step 3 of Algorithm 1 but substitute (8) for
$$g^{t+1}_j(x_j) = \frac{1}{h_j}\sum_{i=1}^{n} \tilde{p}^{(i)}_j K\left(\frac{x_j - \tilde{x}^{(i)}_j}{h_j}\right) \tag{10}$$
6. Follow step 4 of Algorithm 1 to update the copula parameters.

the E step of the EM algorithm and Step 1 of Algorithm 1 are the same. In the M step of the EM algorithm, at each iteration t, one can always write the objective function, given by

$$\begin{aligned}(11)\qquad & \sum_z \sum_i w_{i,z}^t \log c_z \left[G_1\left(\frac{x_1^{(i)} - \mu_{1z}}{\sigma_{1,z}}\right), \dots, G_d\left(\frac{x_d^{(i)} - \mu_{dz}}{\sigma_{d,z}}\right); \theta_z \right] \\ & + \sum_z \sum_i w_{i,z}^t \left\{ \sum_{j=1}^d \log \left[\frac{1}{\sigma_{j,z}} g_j \left(\frac{x_j^{(i)} - \mu_{j,z}}{\sigma_{j,z}} \right) \right] \right\} \\ & + \sum_z \sum_i w_{i,z}^t \log \pi_z,\end{aligned}$$

where $w_{i,z}$ are the weights calculated at the E step. The heuristic is to find values of $\pi_z, \mu_{j,z}, \sigma_{j,z}, g_j, \theta_z$ at which the objective function is high. For π_z, the solution is standard and independent of the other parameters: it is the formula in Step 1 of Algorithm 1. For the other parameters, we use the following tricks.

If updates $\mu_{j,z}^{t+1}, \sigma_{j,z}^{t+1}, g_j^{t+1}$ were available, then they could be plugged in to the first term in (11), which in turn could be maximized over θ_z. To get the updates $\mu_{j,z}^{t+1}, \sigma_{j,z}^{t+1}, g_j^{t+1}$, one tries to "maximize" the second term in (11). But what does it mean to "maximize" the second term? A method of [7] is used. By the decoupling of the marginals, we are left with d univariate problems similar to those of [7]. Their idea consists of updating the location parameters $\mu_{j,z}$ (the scale parameters $\sigma_{j,z}$ were assumed to be one) by proceeding *as if* the generators g_j were Gaussian, leading to the formula in Step 2 (b) in Algorithm 1. Once updates $\mu_{j,z}^{t+1}$ have been obtained, they go on by proposing an elegant way of updating the generators g_j: this is Step 3 of Algorithm 1. Note that step 2 (c) of Algorithm 1 simply incorporates the estimation of the scale parameters $\sigma_{j,z}$ in a straightforward way. The novelty of Algorithm 2 is to replace the standard kernel density estimator of Algorithm 1 by a new one that satisfies identifiability constraints and is consistent, as shown in Section 4.

Let us come back to the "as if the g_j were Gaussian" argument mentioned above. Although the original authors [7] did not mention the following argument, this "Gaussian trick" can be supported for densities symmetric about zero. Indeed, maximization over $\mu_{j,z}$ of the second term in (11) is tantamount to solving

$$\sum_{i,z} \frac{g_j'(x_j^{(i)} - \mu)}{g_j(x_j^{(i)} - \mu)} w_{i,z}^t = 0.$$

Now, if g_j is symmetric and hence an even function, then g_j'/g_j is an odd function and odd functions are quite close to the identity function around zero because the terms of even order in Taylor expansions are zero themselves exactly. Thus, up to quadratic approximation, and assuming that the data observations are tightly concentrated around their mean value within each cluster, the formula in Step 2 (b) is approximately correct.

Initialization of the algorithms

To get initial values $\pi_z^0, \mu_{j,z}^0, \sigma_{j,z}^0, g_j^0, \theta_z^0$, one can proceed as follows. A k-means algorithm is run and the returned centers provide values for the location parameters $\mu_{j,z}^0$. The returned partition of the data is used to estimate the remaining parameters. The proportions of the returned clusters provide values for the proportion parameters π_z^0. The standard errors of the clusters provide values for the scale parameters $\sigma_{j,z}^0$. The estimation of the generators g_j^0 is based on the shuffle of all the K standarized samples. That is, one builds the sample $\{(x_j^{(i)} - \mu_{j,z}^0)/\sigma_{j,z}^0\}_{i,z}$ and performs kernel density estimation on it to get an estimate for g_j^0. Finally, copula parameters θ_z^0 are estimated by standard methods [13, 14], cluster by cluster. Note that, as with the standard EM algorithm, Algorithm 1 and Algorithm 2 may depend on the initialization. It is therefore advisable to test as many starting points as possible.

Check of convergence

The convergence of Algorithm 1 and Algorithm 2 is checked visually. Since the likelihood contains all the information about the parameters, we check that the log-likelihood has increased until stabilization, in average. "In average" means that one must take into account the inherent randomness of the algorithms, which is why the check is done visually. Practically, we set a large number of iterations, let the algorithms run, and inspect the plot afterwards. If the log-likelihood has not stabilized around a mean value, we increase the number of iterations. Examples are given in Section 5 and Section 6.

4 Kernel density estimation under moment constraints

We consider the problem of estimating the common density g of independent random variables $X_1, \ldots, X_n$. We assume that g verifies the regularity conditions in Assumption 1

Assumption 1. *The density g is continuous on $\mathbf{R}$, symmetric about zero and obeys*

$$\int x^2 g(x)\,\mathrm{d}x = 1 \neq \int x^4 g(x)\,\mathrm{d}x < \infty.$$

Note that the assumed symmetry implies

$$\int x g(x) \mathrm{d}x = 0.$$

Continuity is a standard assumption to ensure pointwise consistency of the standard kernel density estimator [30] and the Nadaraya-Watson estimator [42]. The condition on the moment of second order stems from the structure of the model in Section 2.2. The moment of fourth order must have a different value than that of the moment of second order to ensure the convergence of a certain quantity (see the proof of Theorem 1 for details). We view this rather as a technical condition. For instance if g were the Gaussian density, its variance would have to be not equal to 1/3.

As explained in Section 3, our aim is to construct an estimator $\hat{g}$ that obeys

$$\int x\hat{g}(x)\,\mathrm{d}x = 0, \text{ and } \int x^2\hat{g}(x)\,\mathrm{d}x = 1. \tag{12}$$

We define the estimator

$$\hat{g}(x) = \sum_{i=1}^{n} \hat{p}_{n,i} K_{h_n}(X_i - x) \tag{13}$$

where $K_{h_n}(y) = K(y/h_n)/h_n$ is a kernel depending on a positive sequence h_n and where $\hat{\mathbf{p}}_n = (\hat{p}_{n,1}, \ldots, \hat{p}_{n,n})'$ (throughout $'$ stands for the transpose operation) is the unique solution of the random optimization problem

$$\min_{\mathbf{p} \in \mathbf{R}_n} \|\mathbf{p}\|_2^2 \tag{14}$$

$$\text{such that } \begin{cases} \widehat{M}_n \mathbf{p} = \mathbf{b}_n \\ \mathbf{p} \geq \mathbf{0}, \end{cases} \tag{15}$$

where $\mathbf{p} = (p_1, \ldots, p_n)'$ and

$$\widehat{M}_n = \begin{pmatrix} 1 & \cdots & 1 \\ X_1 & \cdots & X_n \\ X_1^2 & \cdots & X_n^2 \end{pmatrix}, \text{ and } \mathbf{b}_n = \begin{pmatrix} 1 \\ 0 \\ 1 - h_n^2 \end{pmatrix}.$$

Each $\hat{p}_{n,i}$ is a function of the random sample. For each realization of the sample, the optimization problem (14) is convex and hence admits a unique solution which is denoted by $\hat{\mathbf{p}}_n$. The constraint (15) ensures that $\hat{g}$ satisfies (12). Indeed, elementary calculations show that (12) holds if and only if

$$\sum_{i=1}^{n} \hat{p}_{n,i} X_i = 0 \text{ and } \sum_{i=1}^{n} \hat{p}_{n,i} X_i^2 = 1 - h_n^2,$$

respectively. The constraints $\sum_i \hat{p}_{n,i} = 1$ and $\hat{p}_{n,i} \geq 0$, $i = 1, \ldots, n$, must always hold to ensure that $\hat{g}$ is a density.

As soon as $n > 3$ the system $\widehat{M}_n \mathbf{p} = \mathbf{b}$ has infinitely many solutions and hence there are infinitely many estimators that satisfy (12). We chose to pick the closest one to the standard kernel density estimator. The standard kernel density estimator is an estimator of the form (13) where $\hat{p}_{n,i} = 1/n$, and the solution of

$$\min_{(p_1, \ldots, p_n)} E \int \left(\sum_{i=1}^{n} p_i K_{h_n}(X_i - x) - g(x)\right)^2 \mathrm{d}x.$$

In our case, we cannot set $\hat{p}_{n,i} = 1/n$ because the constraint (15) would not be satisfied. But we can project $(1/n, \ldots, 1/n)$ onto the feasible space given in (15), which amounts to solve the optimization problem (14) because minimizing $\|\mathbf{p}\|^2$ is the same as minimizing $\|\mathbf{p} - \mathbf{e}\|^2$, where $\mathbf{e} = (1, \ldots, 1)'$. Thus, the minimization of $\|\mathbf{p}\|_2$ is a heuristic justified by an analogy. Moreover, even though one can imagine other criteria [16] for choosing $\mathbf{p}$, the choice of the euclidean norm is the easiest from a theoretical and computational point of view.

Having defined the estimator in (13), it is natural to require at least pointwise consistency. The issue resides in the constraint $\mathbf{p} \geq 0$. Without such a constraint, Lemma 1 states that the solution of the optimization problem is explicit and yields a consistent estimate. In the presence of the constraint, Theorem 1 states that consistency can be achieved under a condition on the tail of the underlying density.

Theorem 1. *Suppose Assumption 1 holds. If $h_n \to 0$, $nh_n \to \infty$ and there exist constants $a_n > 0$, $b_n \in \mathbf{R}$ such that $n^{-1/4}a_n \to 0$, $h_n a_n \to 0$, $n^{-1/4}b_n \to 0$, $h_n b_n \to 0$ and*

$$a_n^{-1}(\max\{X_1, \dots, X_n\} - b_n) \tag{16}$$

converges in distribution, then the estimator (13) is pointwise consistent.

The conditions $h_n \to 0$ and $nh_n \to \infty$ are necessary to ensure pointwise convergence of the standard kernel density estimator [38]. The condition (16) is standard in extreme value theory [34]. The conditions $n^{-1/4}a_n \to 0$ and $n^{-1/4}b_n \to 0$ state that the rate at which the sample maxima grows to infinity must not be too fast. The conditions $h_n a_n \to 0$ and $h_n b_n \to 0$ state that the rate at which the sample maxima grows to infinity must be smaller than the rate at which the bandwidth h_n vanishes. If h_n is the optimal bandwidth, that is if $h_n \propto n^{-1/5}$, then the conditions $n^{-1/4}a_n \to 0$ and $n^{-1/4}b_n \to 0$ are automatically satisfied. Example 1 and Example 3 give distributions which satisfy these conditions. Example 2 is a counter-example. Example 1 and Example 2 are drawn from [9], p. 153–157. The computation of the normalizing constants in Example 3 is given in the Appendix.

Example 1. *Let $h_n \propto n^{-1/5}$. The Gaussian distribution $(2\pi)^{-1/2}\exp(-x^2/2)$, $x \in \mathbf{R}$, satisfies the conditions in Theorem 1 with*

$$a_n = (2\log n)^{-1/2}, \qquad b_n = \sqrt{2\log n} - \frac{\log(4\pi) + \log\log n}{2(2\log n)^{1/2}}$$

Example 2 (Counter-example)**.** *The Cauchy distribution $g(x) = [\pi(1+x^2)]^{-1}$, $x \in \mathbf{R}$, does* not *satisfy the conditions in Theorem 1. Indeed, in addition to have infinite variance, the normalization constants are given by $a_n = n/\pi$ and $b_n = 0$. The sequence (a_n) does* not *verifies $n^{-1/4}a_n \to 0$.*

Example 3. *Let $h_n \propto n^{-1/5}$. The Laplace distribution $g(x) = \exp(-|x|/b)/(2b)$, $b > 0$, $x \in \mathbf{R}$, satisfies the conditions in Theorem 1 with $a_n = b$ and $b_n = b\log(n/2)$.*

5 Computer experiments

In this section, we wish to compare Algorithm 1 (hereafter called cKDE for convenience) and Algorithm 2 (fKDE) in terms of the quality of the obtained estimates. The standard Gaussian Mixture Model (GMM) was also implemented as a benchmark.

We generated 500 datasets of size $n = 300, 500, 700, 900$ according to the following data generating process. The number of clusters was set to $K = 3$ and their proportion parameters were all set of equal value. The Frank family of bivariate copulas, given by

$$C_{\theta_z}(u,v) = \begin{cases} -\frac{1}{\theta_z}\log\left(1 + \frac{(e^{-\theta_z u}-1)(e^{-\theta_z v}-1)}{(e^{-\theta_z}-1)}\right), & \text{if } \theta_z \in (-\infty,\infty)\setminus\{0\}, \\ uv, & \text{if } \theta_z = 0. \end{cases}$$

was chosen for all of the three copulas. The parameters were $\theta_1 = -3.45$, $\theta_2 = 3.45$ and $\theta_3 = 0$, corresponding to negative, positive and null dependence levels, respectively. The generators for the marginals along the first, resp. second, axis (g_1, resp. g_2), were a normal, resp. a Laplace, distribution with zero mean and unit variance. The three clusters had means ($\mu_{1,1} = -3, \mu_{2,1} = 0$), ($\mu_{1,2} = 0, \mu_{2,2} = 3$) and ($\mu_{1,3} = 3, \mu_{2,3} = 0$) respectively. The scale parameters along the first, resp. second, axis were set to $\sigma_{1,1} = 2$, $\sigma_{1,2} = 0.7$ and $\sigma_{1,3} = 1.4$, resp. $\sigma_{2,1} = 0.7$, $\sigma_{2,2} = 1.4$ and $\sigma_{2,3} = 2.8$.

All the three algorithms were run with 100 iterations and initialized according to Section 3.1. The kernel and the bandwidth selection method used for building the kernel density estimators were the Gaussian kernel and the method given by (3.30) in p. 47 of [38]. That is, the bandwidth is $1.06An^{-1/5}$, where A is the minimum between the standard deviation of the data and, the interquartile range divided by 1.34.

In order to compare the algorithms, we computed the mean absolute errors, that is, the differences in absolute value between the true parameters and the estimates. These were averaged over the clusters and the coordinates (if any). For the generators, the L_1 norm was used instead. Only the errors for the location, scale and proportion parameters were computed for GMM. The misclassification rate was computed, too. All these error measures can be computed at each iteration of the algorithms and averaged over the replications. The results are shown in Fig. 1.

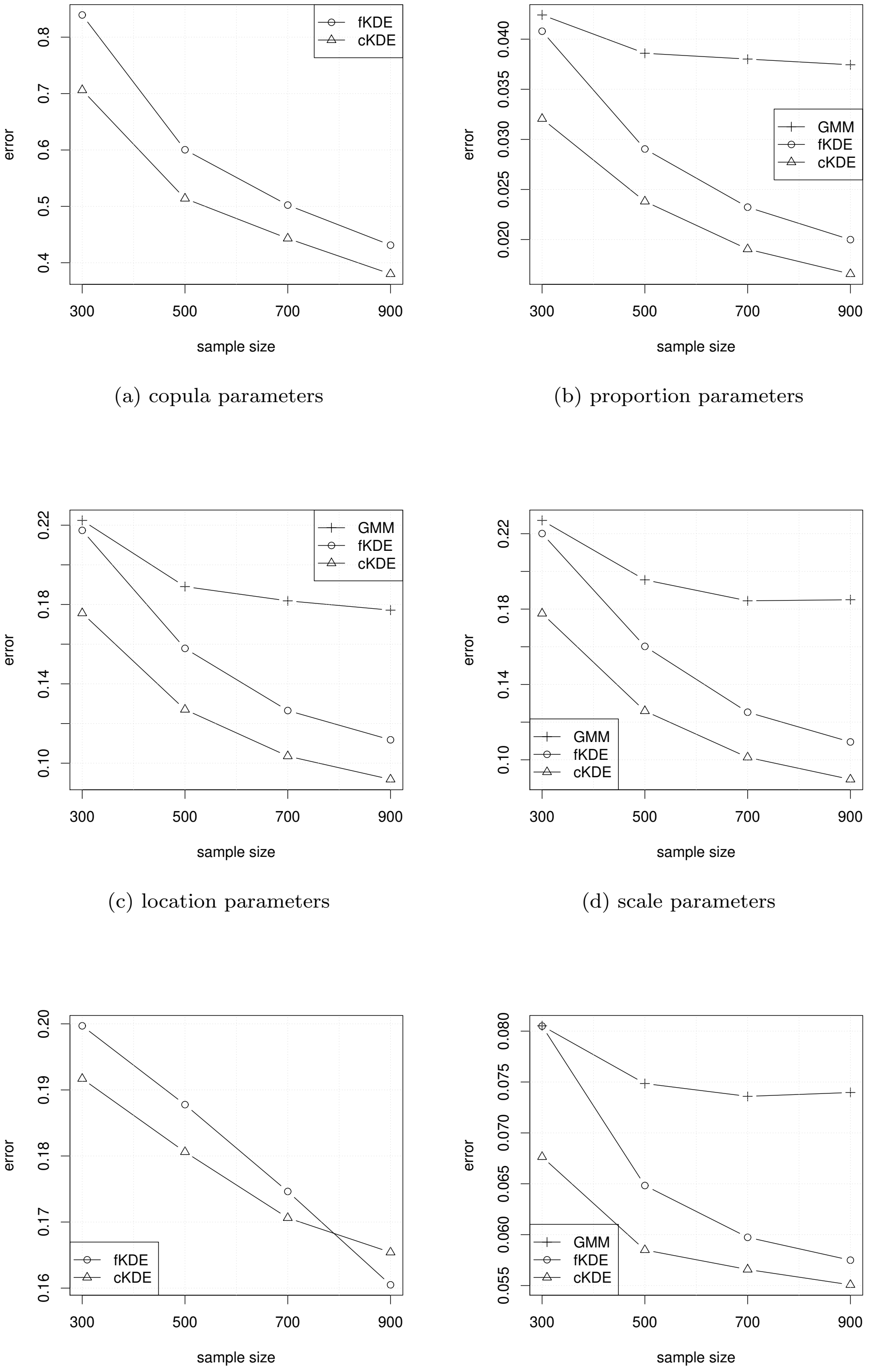
From a clustering point of view, the three learning algorithms can be compared on the basis of the misclassification error rate in Fig. 1 (f). Both the semiparametric algorithms perform better than GMM, especially for large sample sizes, where nonparametric modeling can express its potential (see Fig. 2). When the sample size is not so large, cKDE performs much better than fKDE. This indicates that the new kernel density estimator implemented in cKDE was a good idea. For larger sample sizes, the difference diminishes.

With regard to estimation, cKDE performs better than fKDE but the difference is approximately constant across the sample sizes. Both the semiparametric algorithms perform better than GMM for the proportion (Fig. 1 (b)), location (Fig. 1 (c)) and scale (Fig. 1 (d)) parameters in a way that is similar to the missclassification error. That is, the gain is much more important as the sample size gets larger.

The stability of the algorithms was checked by plotting the log-likelihood and the pointwise averaged error trajectories. Specifically, we focused on the case $n = 300$ and inspected the behaviors of the error trajectories across the iterations. These are displayed in Fig. 3. In Fig. 3 (f), we see that, in average, the log-likelihood increases and stabilizes after 20–30 iterations. This is in agreement with the heuristics discussed in Section 3.2. The log-likelihood of cKDE is higher than that of fKDE, which is in agreement with the results found in Figure 1. In the other panels, note that the method with the lowest trajectory cannot formally be claimed the best because the point at which would converge the algorithms is unknown. Of course if the sample size is large enough then the true parameter would be close to this point. Thus the error trajectories (except the log-likelihood) are here only to inspect convergence of the algorithms.

In view of the panels (a), (c) and (d) of Fig. 3, convergence seems to have been reached after 30 iterations. For the proportion parametres in panel (b), cKDE seems to have reached convergence while it is less clear for fKDE and even GMM. The density generators in panel (e) exhibit a high variability.

To get an idea of the variability of individual trajectories, Fig. 4 shows the



(a) copula parameters

(b) proportion parameters

(c) location parameters

(d) scale parameters

(e) density generators

(f) misclassification errors

Figure 1: Averaged error values for the various parameters and algorithms in terms of the sample size.

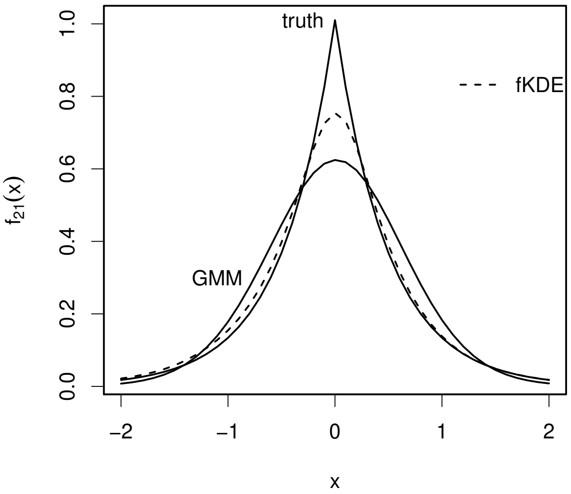


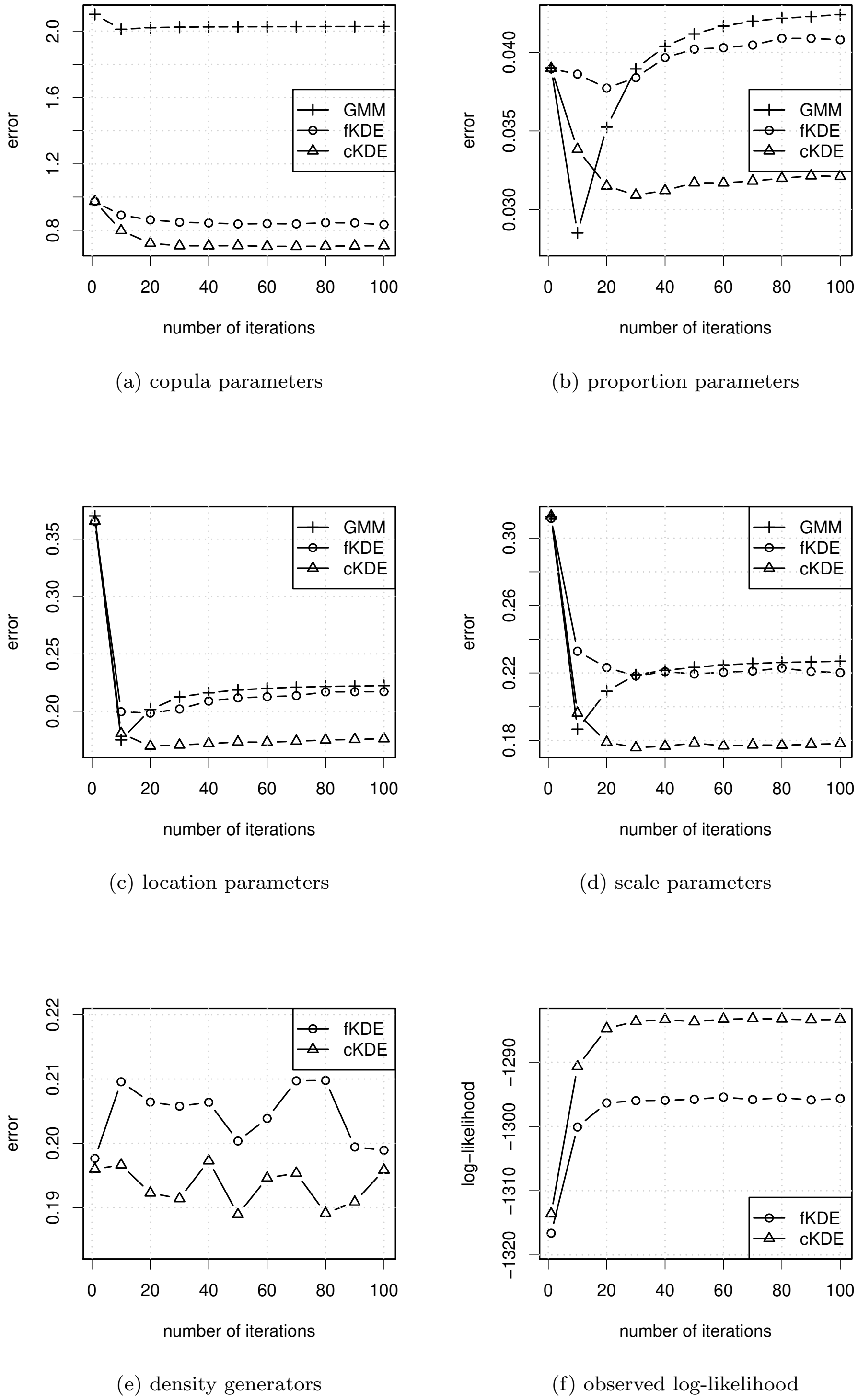
Figure 2: Pointwise averaged marginal density estimates along the second axis in the first cluster for GMM and fKDE. The true underlying density is added for comparison.

trajectories for the first replication of the experiment under focus.

Finally, we repeated the computer experiment described at the beginning of this section with two modifications, one at a time. First, in five successive experiments, the proportion parameters π were set to $(1/4, 3/8, 3/8)$, $(1/5, 2/5, 2/5)$, $(1/6, 5/12, 5/12)$, $(1/7, 6/14, 6/14)$, $(1/8, 7/16, 7/16)$. Second, a fourth component with location parameter $(\mu_{1,4}, \mu_{2,4})$ given by $(0, 8)$, $(0, 6)$, $(0, 4)$, $(0, 2)$, $(0, 0)$ with independent marginals and $\sigma_{1,4} = \sigma_{2,4} = 1$ was added. The cluster proportions were $\pi = (1/4, 1/4, 1/4, 1/4)$. Note that, since the centers of the other clusters are $(-3, 0)$, $(0, 3)$ and $(3, 0)$, the added fourth component gets closer to the other clusters.

The computed missclassification errors averaged over 100 replications are shown in Table 1. We see that the missclassification error seems to remain stable (it exhibits only a very slight increase) as the proportion of observations in the first cluster decreases from $\pi_1 = 1/4$ to $1/6$. Thus, there is some robustness in the clustering in spite of nonparametric estimation.

In general, clustering methods typically perform worst as the amount of separation between clusters decreases. This is also illustrated in Table 1, where the error goes from 7-8% to 21-25%. Compared to GMM, the increase of the error is not worst for semiparametric copula models.



(a) copula parameters

(b) proportion parameters

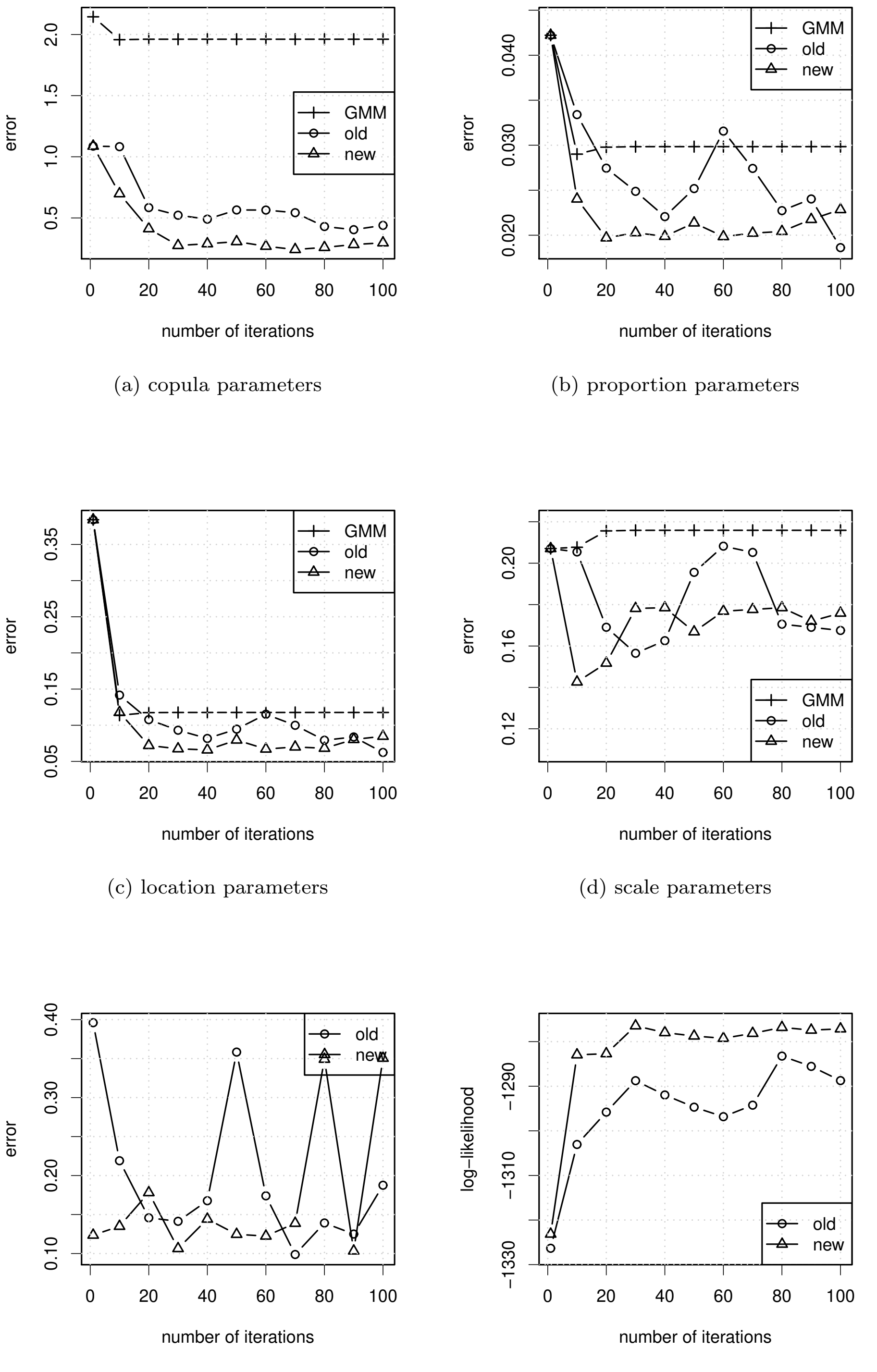
(c) location parameters

(d) scale parameters

(e) density generators

(f) observed log-likelihood

Figure 3: Trajectories of the errors (from (a) to (e)) and log-likelihoods (f), averaged over the replications. The x-line is the number of steps and the y-line the value of the error or observed log-likelihood.



(a) copula parameters

(b) proportion parameters

(c) location parameters

(d) scale parameters

(e) density generators

(f) observed log-likelihood

Figure 4: Trajectories of the errors (from (a) to (e)) and log-likelihoods (f) for the first replication. The x-line is the number of steps and the y-line the value of the error or observed log-likelihood.

	$\pi_1 = 1/4$	$1/5$	$1/6$	$\mu_{2,4} = 8$	6	4	2	0
algo: cKDE	7	7	8	7	10	12	16	21
fKDE	8	8	9	7	11	11	15	25
GMM	9	10	11	8	11	14	18	23

Table 1: Averaged missclassification error of the computer experiment under eight situations, in %.

6 Illustrations on real data

6.1 Illustration on the Iris data

We performed two data analyses of the well-known iris data available in the R software. In the first analysis, we took the first (sepal length) and the third (petal length) variables and ran Algorithm 2 for four families of copulas, namely, the Frank, Clayton, Gaussian copulas and the copula representing independence between the variables. The convergence of the algorithms was checked by inspection of the log-likelihood values, which stabilized after 50 iterations around -250, -245, -246, -288, for the four copula families, respectively. The missclassification errors are given in Table 2. The lowest errors correspond to the highest likelihoods. In terms of the estimated densities, the difference between the copulas is depicted in Figure 5. The Frank and Gaussian copulas seem to best fit the data, in agreement with the results in Table 2.

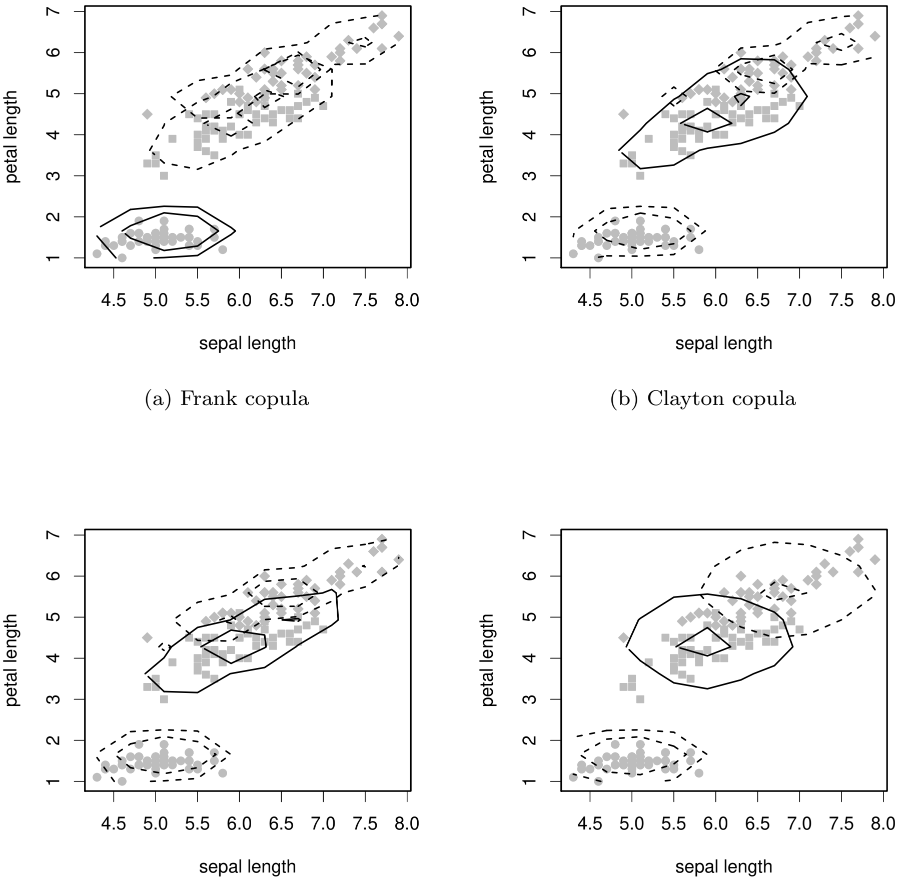
	Frank	Clayton	Gaussian	independence
log-likelihood	-250	-245	-246	-288
missclassification error (%)	16	10	8	14

Table 2: Missclassification errors and estimated log-likelihood for the different copula families

In the second analysis, we ran Algorithm 2 with the complete data, that is, we fitted a semiparametric copula-based mixture model of dimension $d = 4$. Only the Gaussian copula was tested because the other copulas do not generalize easily in higher dimensions. We let the algorithm run 50 iterations and observed that the log-likelihood seem to have stabilized, see Figure 6(a). The isocontours, depicted in Figure 6(b) do not differ much from those in Figure 5(c).

6.2 Illustration on RNA-seq data

The use of high-throughput sequencing technologies to sequence ribonucleic acid content results in the production of RNA-seq data. From a statistical point of view, the observations are (realizations of) random variables $Y_{i,j}$, $i = 1, \ldots, n$, $j = 1, \ldots, d$, each of which is a measure of the digital gene expression (DGE) of the biological entity i (e.g., a gene) for the experimental condition j. For instance, $Y_{i,j}$ may be the number of reads of the ith gene for the jth condition aligned to a reference genome sequence. One question of interest deals with the clustering of DGE profiles [32]. For instance, one may want to discover groups of co-expressed genes.



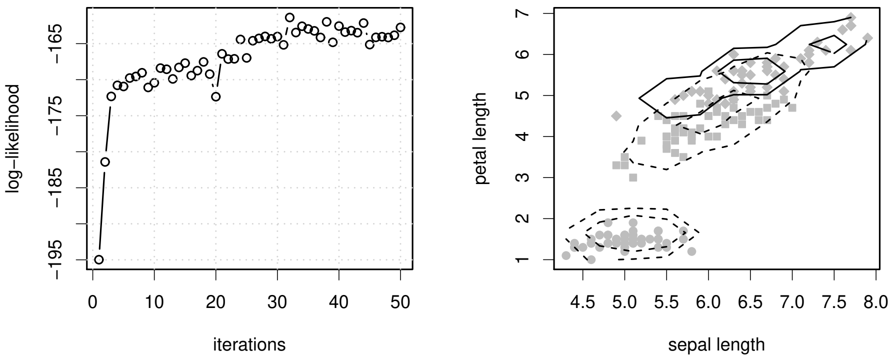
(a) Frank copula

(b) Clayton copula

(c) Gaussian copula

(d) Independence

Figure 5: Isocontours of estimated densities under different copula assumptions: Frank copula (a), Clayton copula(b), Gaussian copula(c) and independence(d).



(a) Log-likelihood

(b) Isocontours

Figure 6: Results of the semiparametric model for the fit in four dimensions: (a)Log-likelihood value of the complete Iris data; (b)isocontours of the estimated density, component by component (the estimated density is built on the four variables of the iris data and this is the coordinatewise projection on the first and third variables).

In recent years several clustering methods have been proposed. Poisson mixture models can be applied but they need to assume that, within a cluster, the DGE measures are independent, a very strong assumption. More precisely, they are of the form [32] $f(y;\psi) = \prod_{i=1}^{n} \sum_{z=1}^{K} \pi_z f_z(y_i;\chi_{iz})$ where $f_z(y_i;\chi_{iz}) = \prod_{j=1}^{d} \prod_{l=1}^{r_j} \mathcal{P}(y_{ijl};\mu_{ijlz})$ and $\chi_{iz} = \{\mu_{ijlz}\}_{jl}$, $\psi = \{\pi_z, \chi_{iz}\}_{i,z}$. Here $r_j = 1$, $j = 1, \ldots, d$ and $\mathcal{P}$ denotes the Poisson density. Another approach consists of applying a transformation $Y_{i,j} \mapsto \tilde{Y}_{i,j}$, $i = 1, \ldots, n$, $j = 1, \ldots, d$, so that the transformed data, or pseudo data, are more appropriate for Gaussian mixture models [31]. One such transformation [12] is given by

$$\tilde{Y}_{i,j} = \log\left(\frac{Y_{i,j}/N_j + 1}{m_i + 1}\right),$$

where $N_j = \sum_{i=1}^{n} Y_{i,j}/10^6$ and $m_i = d^{-1} \sum_{j=1}^{d} N_j^{-1} Y_{i,j}$. This approach essentially amounts to assuming that the data are Gaussian on a log-scale. The semiparametric copula-based mixture models permit to relax this assumption.

In this section, we compare the Poisson mixture model of [32], the Gaussian mixture model and the semiparametric copula-based mixture models with Gaussian and Frank copulas. The data are high-throughput transcriptome RNA-seq data [40] downloaded from the companion `R` package `HTSCluster` of [32]. We removed the biological replicates so that $d = 2$. Estimation in the semiparametric copula-based models was performed with Algorithm 2. Estimation in the Poisson mixture model was performed with the function `PoisMixClus` of the package `HTSCluster`. All the algorithms were run with a fixed number of clusters, set to $K = 10$, corresponding to the number of clusters selected by the integrated completed likelihood criterion in the analysis performed in [33].

In order the compare the models, we reproduced Fig. 2 of [32]. The bar heights in Fig. 7 stand for the quantities

$$\frac{\sum_{i=1}^{n} \hat{w}_{i,z} Y_{i,j}}{\sum_{i=1}^{n} \hat{w}_{i,z} \sum_{j=1}^{d} Y_{i,j}},$$

each of which, according to [32], can be interpreted as the proportion of reads that are attributed to condition j in cluster z. The quantities $\hat{w}_{i,z}$ are estimates of the probability that the i-th observation belongs to the z-th cluster, estimate of which depends on the fitted model (Poisson, GMM, or semiparametric copula-based). Bar widths are proportional to $\hat{\pi}_z$, the estimated cluster proportions. Each bar represents a cluster and each color represents a mean normalized expression profile, the value of which is given by the bar length of a given color. In Figure 7, the results for the Poisson model, the only one which does not take into account the dependence structure within the clusters, differ from all the other models. We note that the copula-based semiparametric models are both similar (compared to the Poisson model) and different from GMM. We take this as an encouragement for copula-based semiparametric models: there are not absurd since similar to GMM; there are potentially of practical interest since they differ from GMM.

7 Open problems and challenges

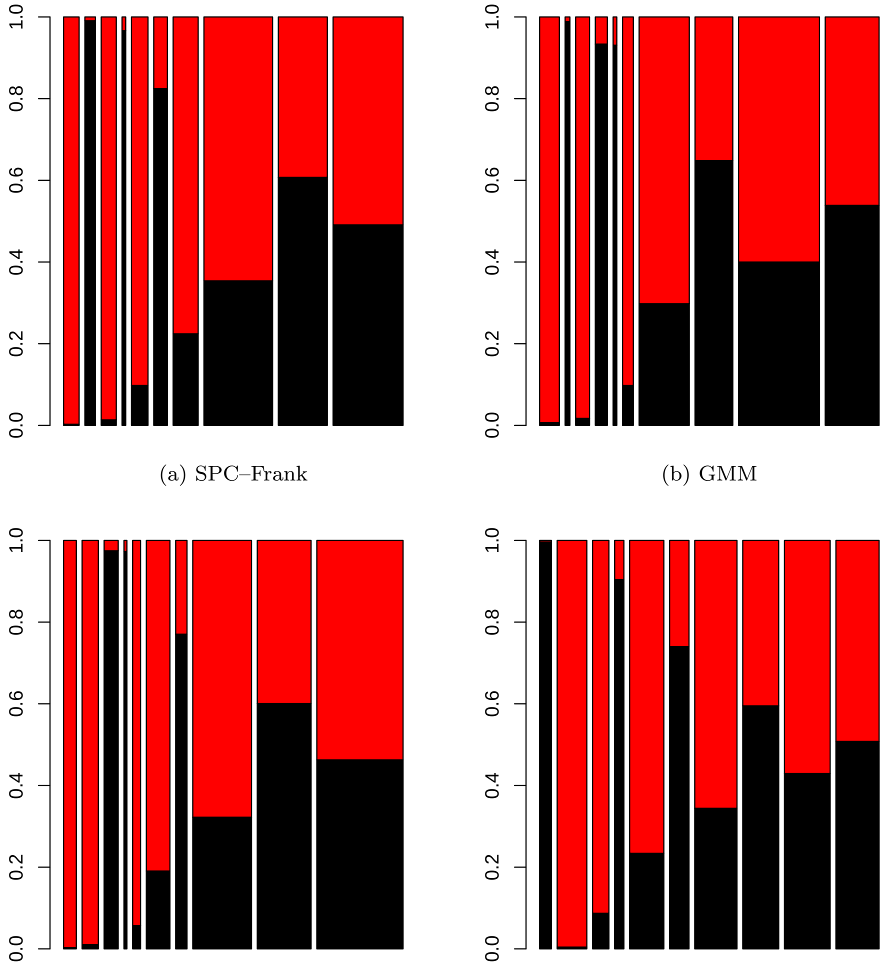
Despite the good results of the learning algorithms in Section 5 and Section 6, there are open problems that need to be addressed to make these algorithms fully applicable in practice.

High dimensions

Are Algorithm 1 and Algorithm 2 applicable to a high-dimensional setting? In principle, the algorithms are written for any dimension d. In practice, however, two issues make the problem challenging. First, as the dimension increases, the number of flexible copula families drops rapidly. Still, there are certain families such as Gaussian copulas, Vines copulas [1] or factor copulas [21] that might be appropriate. But then one must be able to solve the optimization problem in Step 4 of the algorithms.

Number of clusters

An important problem is that of the choice of the number of clusters. To this end, many criteria write as the observed log-likelihood minus a penalization term [2, 6, 15, 37]. But it is unclear what the penalization term should be in semiparametric copula-based mixture models. That said, let us notice that the observed log-likelihoods between different semiparametric copula-based mixture models with the same number of parameters may be compared to at least select the appropriate copula family.



(a) SPC–Frank

(b) GMM

(c) SPC–Gauss

(d) Poisson

Figure 7: Cluster profiles for the Poisson mixture model, the Gaussian mixture model and the semiparametric copula-based mixture models with Frank and Gauss copulas. Each bar represents a cluster and each color represents a mean normalized expression profile, the value of which is given by the bar length of a given color. The bar widths are proportional to the estimated cluster proportions.

Identifiability

As of today, identifiability of semiparametric copula-based mixture models has not been proved. Proving identifiability in general is known to be a difficult problem. In fact, even purely parametric copula-based mixture models, such as those in [20], have not been proved to be identifiable. Identifiability of elliptical mixtures, hence including Student-t mixtures, was not proved until 2006 [17]. Identifiability, in a weaker sense, of nonparametric mixtures with independence components under certain assumptions was proved in 2009 [3]. Identifiability in the weak sense means that non-identifiable parameters in the strong sense belong to a subset of Lebesgue measure zero.

From a statistical perspective, that is, from an estimation point of view, identifiability is an important problem. In this respect, the simulations in Section 5 are reassuring: the true parameters with which the data were simulated could be recovered.

From a learning perspective, that is, from a clustering point of view, identifiability is less important. For example, it is well known that neural networks are not identifiable and still have been enjoying success throughout the sciences.

Convergence

Another important problem is that of checking the convergence of the algorithms given in Section 3.1. The current method is to check visually that the log-likelihood has increased and stabilized, taking into account the inherent stochasticity. The problem is that the standard criterion in EM algorithms, of the form $|\nabla_\phi \sum_i \log f(X_1^{(i)}, \ldots, X_d^{(i)}; \phi)| < \epsilon$, where ∇_ϕ denotes the gradient operator with respect to the parameters ϕ, is not applicable because of the inherent randomness of the algorithms. To address this issue, a smooth summary of the log-likelihood could be considered, as, for instance, its least concave majorant.

8 Summary

We proposed a novel algorithm which permitted to improve the inference in semiparametric copula-based mixture models in which the marginals have a location-scale structure. We did this by replacing the standard kernel density estimator by a weighted one in order to satisfy the inherent constraints of the model. Pointwise consistency of the estimator was proved under mild assumptions. An application to RNA-seq data and a benchmark dataset (the iris data) confirmed the ability of the models to fit real data.

Research on copula-based (and hence genuinely multivariate) semiparametric models has started only recently, and, therefore, many challenges still remain. A list of important open problems is given in Section 7. Among them stands the identifiability problem and the convergence of the algorithms. In fact, with regard to the last point, even for simpler algorithms such as those in [5, 7, 26], the convergence properties are still unknown, even though a first step has been achieved in [23]. Addressing these problems should open the gate for designing sound convergence check methods and performing model selection (including selection of the correct number of clusters) through pseudo-AIC criteria.

Acknowledgment. The author thanks two anonymous referees and an associate editor who made suggestions that helped to improve this paper.

A Appendix

A.1 Computation of the normalizing constants in Example 3

From [9], p. 155, we know that

$$[E(c_n x + d_n; 1/b)]^n \to \Lambda(x), \quad n \to \infty,\, x > 0,$$

where $E(x; 1/b) = 1 - \exp(-x/b)$, $b > 0$ is the distribution function of the exponential distribution, $\Lambda(x) = \exp(-e^{-x})$ is the distribution function of the Gumbel distribution and $c_n = b$, $d_n = b \log n$. Let $L(x; b) = \exp(x/b)/2$, $x > 0$, be the distribution function of the Laplace distribution on the positive real line. Let $a_n = c_n$, $b_n = d_n - b \log 2$ and $x > 0$. By identification of the binomial coefficients in the binomial theorem, we have

$$[L(a_n x + b_n)]^n = [E(c_n x + d_n)]^n \to \Lambda(x),$$

meaning that $a_n = b$ and $b_n = b \log(n/2)$ are the appropriate constants. If $x < 0$, the same formula applies because $a_n x + b_n \to \infty$. □

A.2 Proof of Theorem 1

Theorem 1 shall be proved by first considering the optimization problem (14)–(15) without the constraint $\mathbf{p} \geq \mathbf{0}$. (This shall be called the *simplified* optimization problem.) Throughout the proofs, the bandwidth sequence h_n is simply denoted by h.

Lemma 1. *Let $n \geq 3$. If $h \to 0$ and $nh \to 0$ then the solution $\hat{\mathbf{p}}_n$ of the simplified problem*

$$\min_{\mathbf{p}} \|\mathbf{p}\|_2^2 \tag{17}$$

$$\textit{such that} \left\{ \widehat{M}_n \mathbf{p} = \mathbf{b}_n \right. \tag{18}$$

obeys

$$\hat{\mathbf{p}}_n = \tilde{\mathbf{p}}_n - \frac{(I - \widetilde{H}_n)\mathbf{X}^2}{\mathbf{X}^{2\prime}(I - \widetilde{H}_n)\mathbf{X}^2}(\mathbf{X}^{2\prime}\tilde{\mathbf{p}}_n - 1 + h^2) \tag{19}$$

$$\tilde{\mathbf{p}}_n = \frac{\overline{X^2}\mathbf{e} - \overline{X}\mathbf{X}}{n(\overline{X^2} - \overline{X}^2)} \tag{20}$$

where $\widetilde{H}_n = \widetilde{M}_n'(\widetilde{M}_n\widetilde{M}_n')^{-1}\widetilde{M}_n$ is the projection matrix on the space spanned by $\mathbf{e}$, $\mathbf{X} = (X_1, \ldots, X_n)$, $\overline{X} = n^{-1}\sum_i^n X_i$, and $\overline{X^2} = n^{-1}\sum_i^n X_i^2$. Moreover, the estimator (13) with $\hat{\mathbf{p}}_n$ as in (17)– (18) is pointwise consistent.

Proof of Lemma 1. Since the distribution of X_i has no atom at zero, one has

$$P(\forall \mathbf{y} \in \mathbf{R}^3,\, \widehat{M}_n'\mathbf{y} \neq \mathbf{0} \text{ or } \mathbf{y} = \mathbf{0}) = 1,$$

meaning that $\widehat{M}_n'$ has full rank with probability one. Since $n \geq 3$ this rank must be three. Hence $\widehat{M}_n\widehat{M}_n'$ has full rank equal to three and therfore is invertible. The optimization problem is convex hence there is a unique solution $\hat{\mathbf{p}}_n$ whose expression is easily found: the Lagrangian writes $\mathbf{p}'\mathbf{p} - \lambda(\widehat{M}_n - \mathbf{b}_n)$ for some $\lambda > 0$ and by equating its gradient to zero we get

$$\hat{\mathbf{p}}_n = \widehat{M}_n'(\widehat{M}_n\widehat{M}_n')^{-1}\mathbf{b}_n \tag{21}$$

(and $\lambda = 2(\widehat{M}_n\widehat{M}_n')^{-1}\mathbf{b}_n$).

In order to obtain the desired formulas (19) and (20) it is convenient to introduce

$$\widetilde{M}_n = \begin{pmatrix} 1 & \cdots & 1 \\ X_1 & \cdots & X_n \end{pmatrix} \text{ and } \mathbf{X}^2 = \begin{pmatrix} X_1^2 \\ \vdots \\ X_n^2 \end{pmatrix}.$$

so that we have the decompositions by blocks:

$$\widehat{M}_n = \begin{pmatrix} \widetilde{M}_n \\ \mathbf{X}^{2\prime} \end{pmatrix} \text{ and } \widehat{M}_n\widehat{M}_n' = \begin{pmatrix} \widetilde{M}_n\widetilde{M}_n' & \widetilde{M}_n\mathbf{X}^2 \\ \mathbf{X}^{2\prime}\widetilde{M}_n' & \mathbf{X}^{2\prime}\mathbf{X}^2 \end{pmatrix}.$$

Let $\widetilde{H}_n = \widetilde{M}_n'(\widetilde{M}_n\widetilde{M}_n')^{-1}\widetilde{M}_n$ be the projection matrix onto the linear space spanned by the rows of $\widetilde{M}_n$. With this notation, we have

$$[\widehat{M}_n\widehat{M}_n']^{-1} = \begin{pmatrix} (\widetilde{M}_n\widetilde{M}_n')^{-1} + \frac{(\widetilde{M}_n\widetilde{M}_n')^{-1}\widetilde{M}_n\mathbf{X}^2\mathbf{X}^{2\prime}\widetilde{M}_n'(\widetilde{M}_n\widetilde{M}_n')^{-1}}{\mathbf{X}^{2\prime}(I-\widetilde{H}_n)\mathbf{X}^2} & \frac{-(\widetilde{M}_n\widetilde{M}_n')^{-1}\widetilde{M}_n\mathbf{X}^2}{\mathbf{X}^{2\prime}(I-\widetilde{H}_n)\mathbf{X}^2} \\ \frac{-\mathbf{X}^{2\prime}\widetilde{M}_n'(\widetilde{M}_n\widetilde{M}_n')^{-1}}{\mathbf{X}^{2\prime}(I-\widetilde{H}_n)\mathbf{X}^2} & \frac{1}{\mathbf{X}^{2\prime}(I-\widetilde{H}_n)\mathbf{X}^2} \end{pmatrix}$$

Decomposing $\mathbf{b}_n = (\tilde{\mathbf{b}}_n', 1-h^2)'$ and applying formula (21) then yields (19) with $\tilde{\mathbf{p}}_n = \widetilde{M}_n'(\widetilde{M}_n\widetilde{M}_n')^{-1}\tilde{\mathbf{b}}_n$, this last equality being equivalent to (20).

We now introduce an intermediate lemma in order to facilitate the study of remainder terms which shall appear in the proof of consistency.

Lemma 2. *Let $(Z_{n,1}, \ldots, Z_{n,n})$ be i.i.d. random variables defined on the same probability space as $X_1, \ldots, X_n$. They are assumed to obey $n^{-1}\sum_{i=1}^n Z_{n,i}X_i^k \to c_k$, $k = 0, 1, 2$, in probability as $n \to \infty$ where c_k is some real constant. Then*

$$\frac{1}{n}\mathbf{X}^{2\prime}(I - \widetilde{H}_n)\mathbf{Z}_n \overset{P}{\to} c_2 - c_0, \quad n \to \infty,$$

where $\mathbf{Z}_n = (Z_{n,1}, \ldots, Z_{n,n})'$.

Proof of Lemma 2. Write

$$\frac{1}{n}\mathbf{X}^{2\prime}(I - \widetilde{H}_n)\mathbf{Z}_n = \frac{1}{n}\sum_{i=1}^n X_i^2 Z_{n,i} - \frac{1}{n}\mathbf{X}^{2\prime}\widetilde{M}_n' n(\widetilde{M}_n\widetilde{M}_n')^{-1}\frac{1}{n}\widetilde{M}_n\mathbf{Z}_n \overset{P}{\to} c_2 - c_0.$$

To see why the limit holds, note that $n(\widetilde{M}_n\widetilde{M}_n')^{-1}$ converges elementwise to the identity matrix.

We now prove the consistency statement of Lemma 1. We have $\hat{g}(x) = \tilde{g}(x) + \hat{g}(x) - \tilde{g}(x)$ with $\tilde{g}(x) = \sum_{i=1}^n \tilde{p}_{n,i}K_h(x - X_i)$ and $\hat{g}(x) - \tilde{g}(x) = \sum_{i=1}^n(\hat{p}_{n,i} -$

$\tilde{p}_{n,i})K_h(x-X_i)$. Using (20) and $\sum_{i=1}^n X_iK_h(x-X_i)/\sum_{i=1}^n K_h(x-X_i) \to x$, we easily get that $\tilde{g}(x) \to g(x)$ in probability. Now using (20)–(21) and Lemma 2 we also get

$$\hat{g}(x) - \tilde{g}(x) = \frac{\mathbf{X}^{2\prime}\tilde{\mathbf{p}}_n + 1 - h^2}{\mathbf{X}^{2\prime}(I - \widetilde{H}_n)X^2}\mathbf{X}^{2\prime}(I - \widetilde{H}_n)K \xrightarrow{P} 0,$$

where $K = (K_h(x-X_1), \ldots, K_h(x-X_n))'$. The proof of Lemma 1 is complete. □

Proof of Theorem 1. In this proof, the symbol $\hat{\mathbf{p}}_n$ stands for the solution of the optimization problem (17)–(18), that is, *without* the positivity constraint, and the symbol $\hat{\mathbf{p}}_n^+$ stands for the solution of the optimization problem (14)–(15), that is, *with* the positivity constraint. In view of Lemma 1, it is sufficient to show that

$$P(\hat{p}_{n,i} \geq 0,\, i = 1, \ldots, n) \to 1, \quad n \to \infty,$$

because, by definition of the optimization problems, this implies that

$$P(\hat{p}_{n,i} = \hat{p}_{n,i}^+,\, i = 1, \ldots, n) \to 1$$

and therefore that the estimators are equal with probability tending to one.

We write

$$\hat{p}_{n,i} = \tilde{p}_{n,i}\left(1 + \frac{\hat{p}_{n,i} - \tilde{p}_{n,i}}{\tilde{p}_{n,i}}\right)$$

and the proof will be complete if (i) $P(\tilde{p}_{n,i} \geq 0,\, i = 1, \ldots, n) \to 1$ and (ii) $|(\hat{p}_{n,i} - \tilde{p}_{n,i})/\tilde{p}_{n,i}|$ can be bounded above by a quantity which would not depend on i and would vanish asymptotically.

We first show (i). We have

$$|n\tilde{p}_{n,i} - 1| = \left|\frac{\overline{X}^2 - \overline{X}X_i}{\overline{X^2} - \overline{X}^2}\right| \leq \left|\frac{\overline{X}^2}{\overline{X^2} - \overline{X}^2}\right| + \left|\frac{\overline{X}}{\overline{X^2} - \overline{X}^2}a_n a_n^{-1}X_i\right|.$$

The first term in the right hand side is a $O_P(n^{-1})$ and does not depend on i. Now

$$\begin{aligned} |a_n^{-1}X_i| \leq \vee_i\, &|a_n^{-1}X_i| \\ =&\max\{\vee_i a_n^{-1}X_i, \vee_i - a_n^{-1}X_i\} \\ =&\max\{\vee_i a_n^{-1}(X_i - b_n), \vee_i - a_n^{-1}(X_i + b_n)\} + a_n^{-1}b_n, \end{aligned}$$

where $\vee_i X_i$ is a compact notation for $\max\{X_1, \ldots, X_n\}$. By assumption, $\vee_i a_n^{-1}(X_i - b_n)$ converges in distribution. By symmetry, so does $\vee_i - a_n^{-1}(X_i + b_n)$. Hence, by the continuous mapping theorem, the maximum of $\vee_i a_n^{-1}(X_i - b_n)$ and $\vee_i - a_n^{-1}(X_i + b_n)$ converges in distribution. Thus

$$\begin{aligned} |n\tilde{p}_{n,i} - 1| \leq &\left|\frac{\overline{X}^2}{\overline{X^2} - \overline{X}^2}\right| + \\ &\left|\frac{\overline{X}}{\overline{X^2} - \overline{X}^2}a_n\right| |\max\{\vee_i a_n^{-1}(X_i - b_n), \vee_i - a_n^{-1}(X_i + b_n)\} + a_n^{-1}b_n| \\ =&O_P(n^{-1}) + O_P(n^{-1/2}a_n)(O_P(1) + a_n^{-1}b_n). \end{aligned}$$

The bound does not depend on i and vanishes asymptotically in probability by assumption on the sequences a_n and b_n. This is enough to conclude that (i) holds with probability tending to one.

We finally show (ii). It is convenient to introduce Lemma 3 the proof of which is deferred to the end of this Section.

Lemma 3. *Let v_n be a positive sequence satisfying $v_n^{-1} \to 0$, $v_n^{-1}a_n \to 0$, $v_n^{-1}b_n \to 0$. There exist random quantities A_n, B_n, C_n, D_n, E_n such that, as $n \to \infty$, A_n is $O_P(v_n^{-2})$, B_n, C_n, D_n are $O_P(v_n^{-1})$, E_n tends to a nonzero constant in probability, $P(D_nX_i + E_n > 0,\ i = 1, \dots, n) \to 1$ and*

$$\frac{\hat{p}_{n,i} - \tilde{p}_{n,i}}{\tilde{p}_{n,i}} = \frac{A_n X_i^2 + B_n X_i + C_n}{D_n X_i + E_n} \tag{22}$$

In view of 22, one has

$$\Big|\frac{\hat{p}_{n,i} - \tilde{p}_{n,i}}{\tilde{p}_{n,i}}\Big| \leq \frac{|A_n| \vee_{i=1}^n X_i^2 + |B_n| \vee_{i=1}^n X_i + |C_n|}{E_n - \vee_{i=1}^n - D_n X_i} \tag{23}$$

(we used the fact that $\min\{y_1, \dots, y_n\} = -\max\{-y_1, \dots, -y_n\}$ for the denominator). By assumption and by symmetry, both $\vee_{i=1}^n X_i$ and $\vee_{i=1}^n - X_i$ are $O_P(a_n) + b_n$ and by assumption on v_n,

$$v_n^{-2} \vee_{i=1}^n X_i^2 = \left[\max(v_n^{-1} \vee_{i=1}^n X_i, v_n^{-1} \vee_{i=1}^n -X_i)\right]^2 \xrightarrow{P} 0.$$

Hence the numerator in (23) is $o_P(1)$. The denominator equals $E_n + D_n \vee_{i=1}^n X_i$ if $D_n < 0$ and equals $E_n - D_n \vee_{i=1}^n -X_i$ if $D_n > 0$. Either way, the denominator tends to a constant in probability and

$$\Big|\frac{\hat{p}_{n,i} - \tilde{p}_{n,i}}{\tilde{p}_{n,i}}\Big| \leq$$
$$\max\left\{\frac{|A_n| \vee_{i=1}^n X_i^2 + |B_n| \vee_{i=1}^n X_i + |C_n|}{E_n + D_n \vee_{i=1}^n X_i}, \frac{|A_n| \vee_{i=1}^n X_i^2 + |B_n| \vee_{i=1}^n X_i + |C_n|}{E_n - D_n \vee_{i=1}^n -X_i}\right\}.$$

This upper bound does not depend on i and vanishes asymptotically in probability. This proves (ii). It only remains to prove Lemma 3.

Proof of Lemma 3. Let $\delta_{i,j} = 1$ whenever $i = j$ and $\delta_{i,j} = 0$ whenever $i \neq j$. Let $\widetilde{H}_{i,j}$ denote the element at the i-th row and j-th column of $\widetilde{H}_n$. We have

$$\frac{\hat{p}_{n,i} - \tilde{p}_{n,i}}{\tilde{p}_{n,i}} = \frac{-\sum_{j=1}^n (\delta_{i,j} - \widetilde{H}_{i,j}) X_j^2 \frac{\mathbf{X}^{2\prime}\tilde{\mathbf{p}}_n - 1 + h^2}{\mathbf{X}^{2\prime}(I - \widetilde{H}_n)\mathbf{X}^2}}{\frac{\overline{X^2} - \overline{X}X_i}{n(\overline{X^2} - \overline{X}^2)}}.$$

Standard calculations yield

$$\sum_{j=1}^n (\delta_{i,j} - \widetilde{H}_{i,j}) X_j^2 = \frac{(\overline{X^2} - \overline{X}^2)X_i^2 + (\overline{X}\,\overline{X^2} - \overline{X^3})X_i + \overline{X}\,\overline{X^3} - \overline{X^2}^2}{\overline{X^2} - \overline{X}^2}$$

and hence we can rewrite

$$\frac{\hat{p}_{n,i} - \tilde{p}_{n,i}}{\tilde{p}_{n,i}} = \frac{[-(\overline{X^2} - \overline{X}^2)X_i^2 - (\overline{X}\,\overline{X^2} - \overline{X^3})X_i - \overline{X}\,\overline{X^3} + \overline{X^2}^2][\mathbf{X}^{2\prime}\tilde{\mathbf{p}}_n - 1 + h^2]}{[\overline{X^2} - \overline{X}X_i][n^{-1}\mathbf{X}^{2\prime}(I - \widetilde{H}_n)\mathbf{X}^2]}.$$

This is a ratio of polynomials in X_i that can be identified with (22). One easily sees that $\mathbf{X}^{2\prime}\tilde{\mathbf{p}}_n - 1 + h^2$ is $O_P(n^{-1/2}) + O_P(h^2)$ and hence all the coefficients of the polynomial in the numerator are (at least) $O_P(n^{-1/2}) + O_P(h^2)$. By Lemma 2, $n^{-1}\mathbf{X}^{2\prime}(I - \widetilde{H}_n)\mathbf{X}^2$ tends to $EX_1^4 - 1$ which nonzero by assumption. Therefore the desired equation (22) is satisfied with

$$\begin{aligned}
v_n^{-2} &= n^{-1/2} + h^2,\\
A_n &= -(\overline{X^2} - \overline{X}^2)[\mathbf{X}^{2\prime}\tilde{\mathbf{p}}_n - 1 + h^2]\\
B_n &= -(\overline{X}\,\overline{X^2} - \overline{X^3})[\mathbf{X}^{2\prime}\tilde{\mathbf{p}}_n - 1 + h^2]\\
C_n &= (-\overline{X}\,\overline{X^3} + \overline{X^2}^2)[\mathbf{X}^{2\prime}\tilde{\mathbf{p}}_n - 1 + h^2]\\
E_n &= \overline{X^2}n^{-1}\mathbf{X}^{2\prime}(I - \widetilde{H}_n)\mathbf{X}^2.
\end{aligned}$$

Indeed, $v_n^{-2}a_n^2 = n^{-1/2}a_n^2 + h^2a_n^2 \to 0$ by the assumptions in Theorem 1. Let us show that A_n is $O_P(v_n^{-2})$. We have

$$\begin{aligned}
v_n^2 A_n =&O_p(v_n^2 n^{-1/2}) + O_p(v_n^2 h^2)\\
=&O_p\left(\frac{1}{1 + n^{1/2}h^2}\right) + O_p\left(\frac{1}{1 + n^{-1/2}h^{-2}}\right)\\
=&O_p(1),
\end{aligned}$$

the last equality holding because the sequence $(1 + n^{1/2}h^2)^{-1}$ is bounded. The remaining conditions in Lemma 3 are checked in the same way. The proof of Lemma 3 is complete. Hence the proof of Theorem 1 is complete, too. □

References

[1] K. Aas, C. Czado, A. Frigessi, and H. Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.

[2] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *2nd Inter. Symp. on Information Theory.* Petrov, B. N. and Csaki, F., 1973.

[3] E. S. Allman, C. Matias, and J. A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A):3099–3132, 2009.

[4] G. Anderson, A. Farcomeni, M. G. Pittau, and R. Zelli. Rectangular latent markov models for time-specific clustering, with an analysis of the wellbeing of nations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 2018.

[5] T. Benaglia, D. Chauveau, and D. R. Hunter. An EM-like algorithm for semi- and nonparametric estimation in multivariate mixtures. *Journal of Computational and Graphical Statistics*, 18(2):505–526, 2009.

[6] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725, 2000.

[7] L. Bordes, D. Chauveau, and P. Vandekerkhove. A stochastic EM algorithm for a semiparametric mixture model. *Computational Statistics & Data Analysis*, 51, 2007.

[8] P. Coretto and C. Hennig. Robust improper maximum likelihood: tuning, computation, and a comparison with other methods for robust gaussian clustering. *Journal of the American Statistical Association*, 111(516):1648–1659, 2016.

[9] P. Embrechts, C. Klüppelberg, and T. Mikosch. *Modelling Extremal Events for Insurance and Finance.* Springer, 1997.

[10] M. Fop and T. B. Murphy. Variable selection methods for model-based clustering. *Statistics Surveys*, 12:18–65, 2018.

[11] R. Fujimaki, Y. Sogawa, and S. Morinaga. Online heterogeneous mixture modeling with marginal and copula selection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 645–653, New York, NY, USA, 2011. ACM.

[12] M. Gallopin. *Classification et inférence de réseaux pour les donneées RNA-seq.* PhD thesis, Université Paris-Saclay, 2015.

[13] C. Genest and A.-C. Favre. Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrologic Engineering*, 12(4):347–368, 2007.

[14] C. Genest, K. Ghoudi, and L.-P. Rivest. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3):543–552, 1995.

[15] S. Grønneberg and N. L. Hjort. The copula information criteria. *Scandinavian Journal of Statistics*, 41(2):436–459, 6 2014.

[16] P. Hall and B. A. Turlach. Reducing bias in curve estimation by use of weights. *Computational Statistics & Data Analysis*, 30(1):67 – 86, 1999.

[17] H. Holzmann, A. Munk, and T. Gneiting. Identifiability of finite mixtures of elliptical distributions. *Scandinavian journal of statistics*, 33(4):753–763, 2006.

[18] H. Hu, Y. Wu, and W. Yao. Maximum likelihood estimation of the mixture of log-concave densities. *Computational Statistics & Data Analysis*, 101:137–147, 2016.

[19] D. Kim, J.-M. Kim, S.-M. Liao, and Y.-S. Jung. Mixture of D-vine copulas for modeling dependence. *Computational Statistics & Data Analysis*, 64:1–19, 2013.

[20] I. Kosmidis and D. Karlis. Model-based clustering using copulas with applications. *Statistics and Computing*, pages 1–21, 2015.

[21] P. Krupskii and H. Joe. Factor copula models for multivariate data. *Journal of Multivariate Analysis*, 120:85–101, 2013.

[22] S. Lee and G. J. McLachlan. Finite mixtures of multivariate skew t-distributions: some recent and new results. *Statistics and Computing*, 24(2):181–202, 2014.

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

[24] P. K. Mallapragada, R. Jin, and A. Jain. Nonparametric mixture models for clustering. In *Joint IAPR International Workshop, SSPR & SPR 2010*, volume 6218, pages 334–343. Springer, Hancock, E. R. and Wilson, R. C. and Windeatt, T. and Ulusoy, I. and Escolano, F., 2010.

[25] M. Marbac, C. Biernacki, and V. Vandewalle. Model-based clustering of Gaussian copulas for mixed data. *Communications in Statistics - Theory and Methods*, 46(23):11635–11656, 2017.

[26] G. Mazo. A semiparametric and location-shift copula-based mixture model. *Journal of Classification*, 34(3):444–464, 2017.

[27] G. McLachlan and D. Peel. *Finite mixture models.* John Wiley & Sons, 2004.

[28] S. M. McNicholas, P. D. McNicholas, and R. P. Browne. A mixture of variance-gamma factor analyzers. In *Big and Complex Data Analysis*, pages 369–385. Springer, 2017.

[29] R. B. Nelsen. *An introduction to copulas.* Springer, 2006.

[30] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33:1065–1076, 1962.

[31] A. Rau and C. Maugis-Rabusseau. Transformation and model choice for rna-seq co-expression analysis. *Briefings in Bioinformatics*, page bbw128, 2017.

[32] A. Rau, C. Maugis-Rabusseau, M.-L. Martin-Magniette, and G. Celeux. Co-expression analysis of high-throughput transcriptome sequencing data with poisson mixture models. *Bioinformatics*, 31(9):1420–1427, 2015.

[33] A. Rau, C. Maugy-Rabusseau, M.-L. Martin-Magniette, and G. Celeux. *Co-expression analysis of RNA-seq data with the HTSCluster package.* Version 2.0.8. User guide for the HTSCluster accessible from within R.

[34] S. I. Resnick. *Extreme values, regular variation and point processes.* Springer, 2013.

[35] M. Rey and V. Roth. Copula mixture model for dependency-seeking clustering. In *Proceedings of the 29-th International Conference on Machine Learning*, 2012.

[36] A. Roy and S. K. Parui. Pair-copula based mixture models and their application in clustering. *Pattern Recognition*, 47(4):1689 – 1697, 2014.

[37] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[38] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman & Hall, 1998.

[39] A. Sklar. Fonction de répartition dont les marges sont données. *Inst. Stat. Univ. Paris*, 8:229–231, 1959.

[40] M. Sultan, M. H. Schulz, H. Richard, A. Magen, A. Klingenhoff, M. Scherf, M. Seifert, T. Borodina, A. Soldatov, D. Parkhomchuk, D. Schmidt, S. O'Keeffe, S. Haas, M. Vingron, H. Lehrach, and M.-L. Yaspo. A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, 321(5891):956–960, 2008.

[41] C. Viroli and G. J. McLachlan. Deep gaussian mixture models. *Statistics and Computing*, pages 1–9, 2017.

[42] G. S. Watson. Smooth regression analysis. *Sankhyā Ser. A*, 26:359–372, 1964.