



**HAL**  
open science

# A smoothed semiparametric likelihood for estimation of nonparametric finite mixture models with a copula-based dependence structure

Michael Levine, Gildas Mazo

► **To cite this version:**

Michael Levine, Gildas Mazo. A smoothed semiparametric likelihood for estimation of nonparametric finite mixture models with a copula-based dependence structure. *Computational Statistics*, In press. hal-03900661v2

**HAL Id: hal-03900661**

**<https://hal.inrae.fr/hal-03900661v2>**

Submitted on 21 Feb 2024 (v2), last revised 21 Mar 2024 (v3)

**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.

A smoothed semiparametric likelihood for  
estimation of nonparametric finite mixture models  
with a copula-based dependence structure

Michael Levine<sup>1\*</sup> and Gildas Mazo<sup>2†</sup>

<sup>1</sup>Department of Statistics, Purdue University, 150 N. University Street,  
West Lafayette, 47907, Indiana, USA.

<sup>2</sup>INRAE, MaIAGE, Université Paris-Saclay, Jouey-en-Josas, 78350,  
France.

\*Corresponding author(s). E-mail(s): [mlevins@purdue.edu](mailto:mlevins@purdue.edu);

Contributing authors: [gildas.mazo@inrae.fr](mailto:gildas.mazo@inrae.fr);

†These authors contributed equally to this work.

**Abstract**

In this manuscript, we consider a finite multivariate nonparametric mixture model where the dependence between the marginal densities is modeled using the copula device. Pseudo EM (Expectation-Maximization) stochastic algorithms were recently proposed to estimate all of the components of this model under a location-scale constraint on the marginals. Here, we introduce a deterministic algorithm that seeks to maximize a smoothed semiparametric likelihood. No location-scale assumption is made about the marginals. The algorithm is monotonic in one special case, and, in another, leads to “approximate monotonicity”—whereby the difference between successive values of the objective function becomes non-negative up to an additive term that becomes negligible after a sufficiently large number of iterations. The behavior of this algorithm is illustrated on several simulated and real datasets. The results suggest that, under suitable conditions, the proposed algorithm may indeed be monotonic in general. A discussion of the results and some possible future research directions round out our presentation.

**Keywords:** Nonparametric finite density mixture, copula, pseudo-EM algorithm

# 1 Introduction

Let

$$g(\mathbf{x}) = g(x_1, \dots, x_d) = \sum_{k=1}^K \pi_k f_k(x_1, \dots, x_d) \quad (1)$$

be a multivariate mixture model with  $K$  components (or clusters—we shall use these two words interchangeably). We view the model (1) as a nonparametric mixture model where individual components  $f_k$  are not defined as belonging to any specific parametric family. The research on selecting the number of components for non- and semiparametric density mixtures is currently at a very early stage; some developments in this area can be found in e.g. [Kasahara and Shimotsu \(2014\)](#) and [Kwon and Mbakop \(2021\)](#). Due to this, we assume that the number of components  $K$  is fixed and known in our model. In general, most of the work on nonparametric mixture modeling so far assumed that the marginal distributions  $f_{k1}, \dots, f_{kd}$  of each component are conditionally independent. Such an assumption implies that, conditional on knowing which component a particular observation has been generated from, its distribution is equal to the product of its marginals. More formally, this means that

$$g(\mathbf{x}) = \sum_{k=1}^K \pi_k \prod_{j=1}^d f_{kj}(x_j).$$

This model has been introduced for the first time in [Hall and Zhou \(2003\)](#). The conditions sufficient to ensure identifiability for the conditionally independent model are known [Allman et al \(2009\)](#). There are also a number of approaches to estimating their parameters [Xiang et al \(2019\)](#), both iterative [Benaglia et al \(2009\)](#); [Levine et al \(2011\)](#) and closed form solutions [Bonhomme et al \(2016\)](#). However, the assumption of conditional independence is not always a realistic one. For example, it is unlikely to

be true when dealing with RNA-seq data [Rau et al \(2015\)](#). Thus, it seems desirable to relax this assumption while retaining the generality of the nonparametric approach.

To the best of our knowledge, the only known results on estimation of nonparametric mixture models with conditionally non-independent components are [Mazo \(2017\)](#); [Mazo and Averyanov \(2019\)](#). A somewhat related model was also considered in [Vrac et al \(2012\)](#). There, however, the authors model not the distribution of observations, but rather the distribution of a number of *cumulative distribution curves* assumed to represent observations. Thus, they assume that, at any moment in time, one can observe an entire cumulative distribution curve  $F(\mathbf{x})$  for any of the  $K$  possible distributions comprising the overall mixture. This is not at all possible in our setting.

[Mazo \(2017\)](#) and [Mazo and Averyanov \(2019\)](#) consider a special case of the general nonparametric mixture model, allowing for a non-trivial dependence structure where the marginals are assumed to belong to a location-scale family. Stochastic algorithms were proposed to estimate the copula parameter and the nonparametric marginals. The estimation algorithms, while performing well in practice, do not optimize any particular objective function. Because of this, their convergence analysis will necessarily be a difficult one. In this manuscript, our goal is to suggest a deterministic algorithm capable of estimating the components of a nonparametric mixture model with conditionally non-independent components without a location-scale assumption for the marginals, since such an assumption is far from commonly satisfied in applications.

In order to continue, we are going to fix the notation first. It is well-known that, due to Sklar's theorem [Nelsen \(2007\)](#) p. 18, every  $d$ - dimensional multivariate cumulative distribution function can be represented as a copula of the corresponding marginal cumulative distribution functions. Indeed, let  $F_{k1}(x_1), \dots, F_{kd}(x_d)$  be the marginal cumulative distribution functions of the cumulative distribution function  $F_k(x_1, \dots, x_d)$  that corresponds to the density  $f_k(x_1, \dots, x_d)$ . Then, there exists a  $d$ -

copula  $C_k$ , which is a function  $C_k : [0, 1]^d \rightarrow [0, 1]$ , such that

$$F_k(x_1, \dots, x_d) = C_k(F_{k1}(x_1), \dots, F_{kd}(x_d)),$$

see [Nelsen \(2007\)](#) pp. 46. If the marginal cumulative distribution functions are continuous, then the copula is unique. The copula  $C_k$  can be viewed as a  $d$ -dimensional cumulative distribution function with uniform marginal distributions. Taking the derivative of order  $d$ , one immediately obtains the representation

$$f_k(x_1, \dots, x_d) = c_k(F_{k1}(x_1), \dots, F_{kd}(x_d)) \prod_{j=1}^d f_{kj}(x_j)$$

where  $c_k$  is the density of the copula  $C_k$ . We assume that each copula density  $c_k$  belongs to some parametric family of copula densities indexed by a parameter  $\theta_k$ . Due to this, from now on we will use the index  $k$  as a subscript for  $\theta_k$  only but will drop this subscript for  $c_k$ . Denoting by  $\varphi$  the set of all marginal densities  $\{f_{kj}\}$ , and denoting by  $\pi = (\pi_1, \dots, \pi_K)'$  and  $\theta = (\theta_1, \dots, \theta_K)'$  the vectors of all weights and copula parameters, respectively, we have

$$f_k(\mathbf{x}; \theta, \varphi) = f_k(x_1, \dots, x_d; \theta, \varphi) = c(F_{k1}(x_1), \dots, F_{kd}(x_d); \theta_k) \prod_{j=1}^d f_{kj}(x_j), \quad (2)$$

so that (1) and (2) define a class of mixture densities that can be stated as  $g(\cdot; \pi, \theta, \varphi)$ . To the best of our knowledge, no identifiability results are available concerning this model.

The rest of this manuscript is structured as follows. Section 2 introduces a general algorithm that can be used to estimate finite mixtures of multivariate densities with a dependence structure defined through the use of copulas. Section 3 provides some results about the monotonicity property of two simplified versions of this algorithm.

Section 4 details the implementation of the algorithm. Section 5 analyses the performance of our algorithm with several simulation studies. Section 6 presents applications to two real datasets. Finally, the conclusion section discusses the results obtained and suggests possible directions for future research.

## 2 Algorithm

The goal of our manuscript is to estimate the components and weights of the model (1)-(2). The definition of such an algorithm starts with an objective function that we are going to introduce next. First, let  $K(\cdot)$  be a proper univariate density function that can be used for kernel density estimation and  $K_h(\cdot) := \frac{1}{h}K\left(\frac{\cdot}{h}\right)$  its rescaled version where  $h > 0$  is a bandwidth. Next, for a generic function  $f$ , we define

$$\mathcal{N}_h f(x) := \exp\left(\int K_h(x-u) \log f(u) du\right) \quad (3)$$

which is a nonlinear smoother of the function  $f$ . Note that, even if  $f$  is a density,  $\mathcal{N}f$  is not, in general a density due to Jensen's inequality. Now, we define the operator  $\mathcal{O}$  by  $\mathcal{O}f_k(\mathbf{x}; \theta, \varphi) = c(F_{k1}(x_1), \dots, F_{kd}(x_d); \theta_k) \prod_{j=1}^d \mathcal{N}f_{kj}(x_j)$ . This definition allows different bandwidths for different dimensions and clusters, if needed. Finally, let us denote  $\check{g}(\mathbf{x}; \pi, \theta, \varphi) = \sum_{k=1}^K \pi_k \mathcal{O}f_k(\mathbf{x}; \theta, \varphi)$ .

The objective function we seek to maximize is the population version of the smoothed semiparametric log-likelihood, given by

$$\ell(\pi, \theta, \varphi) = \int g(\mathbf{x}) \log \frac{\check{g}(\mathbf{x}; \pi, \theta, \varphi)}{g(\mathbf{x})} d\mathbf{x}, \quad (4)$$

over all  $(\pi, \theta, \varphi)$ ; here  $g(\mathbf{x})$  is the target density. If the marginal distributions are conditionally independent then  $c(u_1, \dots, u_d; \theta_k) \equiv 1$  for every  $\theta_k$  and  $k$ , and hence (4)

reduces to the smoothed semiparametric log-likelihood considered in [Levine et al \(2011\)](#).

**Lemma 1.** *For any choice of parameters  $\tilde{\pi}, \tilde{\theta}, \tilde{\varphi}$ , the smoothed loglikelihood difference is bounded as*

$$\begin{aligned} \ell(\pi, \theta, \varphi) - \ell(\tilde{\pi}, \tilde{\theta}, \tilde{\varphi}) &\leq \sum_{k=1}^K -\log \frac{\tilde{\pi}_k}{\pi_k} \int g(\mathbf{x}) w_k(\mathbf{x}; \pi, \theta, \varphi) d\mathbf{x} \\ &\quad - \int g(\mathbf{x}) \sum_{k=1}^K w_k(\mathbf{x}; \pi, \theta, \varphi) \log \frac{\prod_{j=1}^d \mathcal{N} \tilde{f}_{kj}(x_j)}{\prod_{j=1}^d \mathcal{N} f_{kj}(x_j)} d\mathbf{x} \\ &\quad - \int g(\mathbf{x}) \sum_{k=1}^K w_k(\mathbf{x}; \pi, \theta, \varphi) \log \frac{c(\tilde{F}_{k1}(x_1), \dots, \tilde{F}_{kd}(x_d); \tilde{\theta}_k)}{c(F_{k1}(x_1), \dots, F_{kd}(x_d); \theta_k)} d\mathbf{x} \\ &:= \Psi_1(\tilde{\pi} | \pi, \theta, \varphi) + \Psi_2(\tilde{\varphi} | \pi, \theta, \varphi) + \Psi_3(\tilde{\theta}, \tilde{\varphi} | \pi, \theta, \varphi), \end{aligned}$$

where the cumulative distribution functions  $\tilde{F}_{kj}$  are those associated with  $\{\tilde{f}_{kj}\} = \tilde{\varphi}$  and

$$w_k(\mathbf{x}; \pi, \theta, \varphi) = \pi_k \mathcal{O} f_k(\mathbf{x}; \theta, \varphi) / \check{g}(\mathbf{x}; \pi, \theta, \varphi), \quad (5)$$

$k = 1, \dots, K$ .

*Proof of Lemma 1.* By definition, the difference of smoothed log-likelihoods can be written down as

$$\begin{aligned} \ell(\pi, \theta, \varphi) - \ell(\tilde{\pi}, \tilde{\theta}, \tilde{\varphi}) &= - \int g(\mathbf{x}) \log \frac{\sum_{k=1}^K \tilde{\pi}_k \mathcal{O} f_k(\mathbf{x}; \tilde{\theta}, \tilde{\varphi})}{\sum_{k=1}^K \pi_k \mathcal{O} f_k(\mathbf{x}; \theta, \varphi)} d\mathbf{x} \\ &= - \int g(\mathbf{x}) \log \sum_{k=1}^K w_k(\mathbf{x}; \pi, \theta, \varphi) \frac{\tilde{\pi}_k \mathcal{O} f_k(\mathbf{x}; \tilde{\theta}, \tilde{\varphi})}{\pi_k \mathcal{O} f_k(\mathbf{x}; \theta, \varphi)} d\mathbf{x} \end{aligned}$$

At this point, it remains only to apply Jensen's inequality to a convex combination on the right-hand side whose coefficients are  $w_k(\mathbf{x}; \theta, \varphi)$ .  $\square$

Instead of minimizing  $\ell(\pi, \theta, \varphi) - \ell(\tilde{\pi}, \tilde{\theta}, \tilde{\varphi})$  with respect to  $(\tilde{\pi}, \tilde{\theta}, \tilde{\varphi})$  directly, we seek to minimize the upper bound proposed by Lemma 1. This approach is in the

spirit of MM (Minimization-Majorization) algorithms; see e.g. [Wu and Lange \(2010\)](#) for the detailed discussion. To do this, our heuristic is to minimize each of the three terms  $\Psi_1(\tilde{\pi}|\pi, \theta, \varphi)$ ,  $\Psi_2(\tilde{\varphi}|\pi, \theta, \varphi)$ ,  $\Psi_3(\tilde{\theta}, \tilde{\varphi}|\pi, \theta, \varphi)$  separately. This is sometimes called “minimization by part”. To minimize the first term  $\Psi_1(\tilde{\pi}|\pi, \theta, \varphi)$ , we have to choose  $\hat{\pi} = \hat{\pi}$  where  $\hat{\pi}_k = \int g(\mathbf{x})w_k(\mathbf{x}; \pi, \theta, \varphi) d\mathbf{x}$ ,  $k = 1, \dots, K$ . This is the result that can be obtained using standard constrained optimization techniques. Note that the resulting minimum must be non-positive since the first term can be made zero by choosing  $\tilde{\pi} = \pi$ . To minimize the second term  $\Psi_2(\tilde{\varphi}|\pi, \theta, \varphi)$ , define, as a first step,

$$\hat{f}_{kj}(u_j) = \alpha_{kj} \int g(\mathbf{x})w_k(\mathbf{x}; \pi, \theta, \varphi)K_{h_{kj}}(x_j - u_j) d\mathbf{x},$$

for any  $k = 1, \dots, K$  and  $j = 1, \dots, d$ , where  $\alpha_{kj}$  is the normalizing constant ensuring that the newly defined  $\hat{f}_{kj}$  is, indeed, a proper density function. Then, we have

$$\begin{aligned} & - \int g(\mathbf{x})w_k(\mathbf{x}; \pi, \theta, \varphi) \log \mathcal{N}\tilde{f}_{kj}(x_j) d\mathbf{x} \\ &= - \int g(\mathbf{x})w_k(\mathbf{x}; \pi, \theta, \varphi) \left( \int K_{h_{kj}}(x_j - u_j) \log \tilde{f}_{kj}(u_j) du_j \right) d\mathbf{x} \\ &= - \int \log \tilde{f}_{kj}(u_j) \hat{f}_{kj}(u_j) du_j. \end{aligned}$$

The same argument as in [Levine et al \(2011\)](#) applies: the quantity above is minimized if we select  $\tilde{f}_{kj}(u) = \hat{f}_{kj}(u)$ . The resulting minimum will also be less than or equal to zero because  $\Psi_2(\tilde{\varphi}|\pi, \theta, \varphi) = 0$  when  $\tilde{\varphi} = \varphi$ .

Now, we can propose the following general algorithm for estimation of  $(\pi, \theta, \varphi)$ .

A1 Choose initial values  $\pi^0, \varphi^0, \theta^0$

A2 Compute the initial set of weights

$$w_k(\mathbf{x}; \pi^0, \theta^0, \varphi^0) = \pi_k^0 \mathcal{O}f_k(\mathbf{x}; \theta^0, \varphi^0) / \check{g}(\mathbf{x}; \pi^0, \theta^0, \varphi^0).$$



A3 At any step of iteration  $t = 1, 2, \dots$  select

$$\pi_k^t = \int g(\mathbf{x}) w_k(\mathbf{x}; \pi^{t-1}, \theta^{t-1}, \varphi^{t-1}) d\mathbf{x},$$

$k = 1, \dots, K$ .

A4 Select as the next value of the density function vector  $\varphi^t = \{f_{kj}^t\}$  where

$$f_{kj}^t(u_j) = \alpha_{kj} \int g(\mathbf{x}) w_k(\mathbf{x}; \pi^{t-1}, \theta^{t-1}, \varphi^{t-1}) K_{h_{kj}}(x_j - u_j) d\mathbf{x}$$

where  $\alpha_{kj}$  is the normalizing constant ensuring that the newly defined function is, indeed, a density function. As a part of this step, also compute updated cumulative distribution functions  $F_{kj}^t(u_j) = \int_{-\infty}^{u_j} f_{kj}^t(y) dy$ .

A5 Choose the value

$$\theta^t = \arg \min_{\theta} \Psi_3(\theta, \varphi^t | \pi^{t-1}, \theta^{t-1}, \varphi^{t-1}).$$

A6 Redefine weights

$$w_k(\mathbf{x}; \pi^t, \theta^t, \varphi^t) = \pi_k^t \mathcal{O} f_k(\mathbf{x}; \theta^t, \varphi^t) / \check{g}(\mathbf{x}; \pi^t, \theta^t, \varphi^t).$$

and return to step A3.

At each step of the algorithm defined above, the marginals are updated first and independently of the copula parameter. This strategy was used in [Mazo \(2017\)](#); [Mazo and Averyanov \(2019\)](#).

**Remark 1.** *In practice, one implements the empirical version of the algorithm. Every integral of the form  $\int g(\mathbf{x}) \zeta(\mathbf{x}) d\mathbf{x}$ , where  $\zeta$  is some arbitrary function, is replaced by  $\frac{1}{n} \sum_{i=1}^n \zeta(\mathbf{X}_i)$ , where  $\mathbf{X}_i = (X_{i1}, \dots, X_{id})$ ,  $i = 1, \dots, n$ , are observations from the target density  $g$ . The objective function to be maximized is then the empirical version of the smoothed log-likelihood, given by  $\frac{1}{n} \sum_{i=1}^n \log \check{g}(\mathbf{X}_i; \pi, \theta, \varphi)$  (up to an additive*

constant). Here the bandwidths of the nonlinear smoothers are allowed to depend on the data.

### 3 Studying the algorithm

Whether the algorithm proposed in Section 2 is monotonic with respect to the objective functional (4) is an open question. In some special cases, the answer is positive. One such case that we identified is when probabilities  $\pi_k$  and the marginal densities  $f_{kj}$  are known beforehand. In such a case, the simplified algorithm is as follows.

B1 Choose initial value of the copula parameter  $\theta^0$ .

B2 Compute the initial set of weights

$$w_k(\mathbf{x}; \pi, \theta^0, \varphi) = \pi_k \mathcal{O} f_k(\mathbf{x}; \theta^0, \varphi) / \check{g}(\mathbf{x}; \pi, \theta^0, \varphi).$$

B3 For any  $t = 1, 2, \dots$  choose the value

$$\theta^t = \arg \min_{\theta} \Psi_3(\theta, \varphi | \pi, \theta^{t-1}, \varphi).$$

B4 Redefine weights

$$w_k(\mathbf{x}; \pi, \theta^t, \varphi) = \pi_k \mathcal{O} f_k(\mathbf{x}; \theta^t, \varphi) / \check{g}(\mathbf{x}; \pi, \theta^t, \varphi).$$

and return to step B3.

**Proposition 1.** *The algorithm defined in B1–B4 is monotonic with respect to  $\theta$ , that is,  $\ell(\pi, \theta^{t-1}, \varphi) - \ell(\pi, \theta^t, \varphi) \leq 0$  for every  $t = 1, 2, \dots$*

*Proof.* The smoothed likelihood difference is bounded from above as

$$\begin{aligned} \ell(\pi, \theta, \varphi) - \ell(\pi, \tilde{\theta}, \varphi) &\leq \Psi_3(\tilde{\theta}, \varphi | \pi, \theta, \varphi) \\ &= - \int g(\mathbf{x}) \sum_{k=1}^K w_k(\mathbf{x}; \pi, \theta, \varphi) \log \frac{c(F_{k1}(x_1), \dots, F_{kd}(x_d); \tilde{\theta}_k)}{c(F_{k1}(x_1), \dots, F_{kd}(x_d); \theta_k)} d\mathbf{x}. \end{aligned}$$

Choosing  $\theta^* = \arg \min_{\tilde{\theta}} \Psi_3(\tilde{\theta}, \varphi | \pi, \theta, \varphi)$  produces

$$\ell(\pi, \theta, \varphi) - \ell(\pi, \theta^*, \varphi) \leq \Psi_3(\theta^*, \varphi | \pi, \theta, \varphi) = \min_{\tilde{\theta}} \Psi_3(\tilde{\theta}, \varphi | \pi, \theta, \varphi);$$

since there exists a value  $\tilde{\theta} = \theta$  such that  $\Psi_3(\theta, \varphi | \pi, \theta, \varphi) \equiv 0$ , the minimal value of  $\Psi_3(\tilde{\theta}, \varphi | \pi, \theta, \varphi)$  will be less than or equal to zero.  $\square$

Another interesting special case results when one assumes that both component weights  $\pi_k$  and copula parameters  $\theta_k$  are known while the marginal densities  $f_{kj}$  are unknown. In this case, the simplified algorithm will be as follows.

C1 Choose initial values  $\varphi^0$

C2 Compute the initial set of weights

$$w_k(\mathbf{x}; \pi, \theta, \varphi^0) = \pi_k \mathcal{O} f_k(\mathbf{x}; \theta, \varphi^0) / \check{g}(\mathbf{x}; \pi, \theta, \varphi^0).$$

C3 For  $t = 1, 2, \dots$  select as the next value of the density function vector  $\varphi^t = \{f_{kj}^t\}$

where  $f_{kj}^t(u_j) = \alpha_{kj} \int g(\mathbf{x}) w_k(\mathbf{x}; \pi, \theta, \varphi^{t-1}) K_{h_{kj}}(x_j - u_j) d\mathbf{x}$ . Here,  $\alpha_{kj}$  is a normalizing constant, ensuring that the newly defined function is, indeed, a density function. As a part of this step, also compute updated cumulative distribution functions  $F_{kj}^t(u_j) = \int_{-\infty}^{u_j} f_{kj}^t(y) dy$ .

C4 Redefine weights

$$w_k(\mathbf{x}; \pi, \theta, \varphi^t) = \pi_k \mathcal{O}f_k(\mathbf{x}; \theta, \varphi^t) / \hat{g}(\mathbf{x}; \pi, \theta, \varphi^t).$$

and return to step C3.

The special case of the general algorithm defined above possesses an “approximate monotonicity” property in the following sense.

**Proposition 2.** *We assume that the target density  $g(\mathbf{x})$  has a compact support  $\Omega$ . We also assume that none of the known weights  $\pi_k$  is equal to zero. Suppose that the kernel function  $K(\cdot)$  is a proper density function defined on  $[-1, 1]$ , bounded away from zero by  $K_* > 0$ , and Lipschitz continuous with a positive Lipschitz constant  $L$ . We assume that the copula density function  $c(u_1, \dots, u_d; \theta)$  is also Lipschitz continuous on  $[0, 1]^d$  and bounded away from zero. Then, there exists a subsequence  $\varphi^{t_l} = (f_{kj}^{t_l}, k = 1, \dots, K, j = 1, \dots, d), l = 1, 2, \dots$ , such that the the algorithm C1–C4 is “approximately monotonically ascending” along this subsequence:*

$$\ell(\pi, \theta, \varphi^{t_{l-1}}) - \ell(\pi, \theta, \varphi^{t_l}) \leq o(1)$$

as  $l \rightarrow \infty$ .

**Remark 2.** *It follows directly from the definition that  $K_* \leq K(\cdot) \leq K^*$  where both  $K_*$  and  $K^*$  are positive. The assumptions of Lipschitz continuity and boundedness away from zero for the kernel function  $K(\cdot)$  do not represent a practical problem since they are not concerned with the actual data—rather,  $K(\cdot)$  is a tool used to analyze the data. Our simulation results suggest that they also may not be necessary.*

**Remark 3.** *The assumption of compact support for the target density  $g(\mathbf{x})$  and, by extension, for all of the marginal densities  $f_{kj}$  does not represent a problem from the practical viewpoint. From the theoretical viewpoint, a result analogous to Proposition 2*

can be proved if one assumes that all of the marginal densities decay to zero sufficiently fast at infinity and using the Fréchet-Kolmogorov theorem instead of the Arzelà-Ascoli theorem [Brezis \(2011\)](#) p. 126.

**Remark 4.** As an example of copulas satisfying conditions of [Proposition 2](#) we can point out Farlie-Gumbel-Morgenstern (FGM) copulas as well as so-called copulas with cubic sections (that are direct generalizations of FGM copulas) [Nelsen \(2007\)](#) pp. 77 – 84.

*Proof.* The difference in log-likelihoods can be bounded as

$$\begin{aligned} \ell(\pi, \theta, \varphi^{t_{l-1}}) - \ell(\pi, \theta, \varphi^{t_l}) &\leq \Psi_2(\varphi^{t_l} | \pi, \theta, \varphi^{t_{l-1}}) + \Psi_3(\theta, \varphi^{t_l} | \pi, \theta, \varphi^{t_{l-1}}) \\ &= - \int g(\mathbf{x}) \sum_{k=1}^K w_k(\mathbf{x}; \pi, \theta, \varphi^{t_{l-1}}) \log \frac{\prod_{j=1}^d \mathcal{N}^{f_{kj}^{t_l}}(x_j)}{\prod_{j=1}^d \mathcal{N}^{f_{kj}^{t_{l-1}}}(x_j)} d\mathbf{x} \\ &\quad - \int g(\mathbf{x}) \sum_{k=1}^K w_k(\mathbf{x}; \pi, \theta, \varphi^{t_{l-1}}) \log \frac{c(F_{k1}^{t_l}(x_1), \dots, F_{kd}^{t_l}(x_d); \theta_k)}{c(F_{k1}^{t_{l-1}}(x_1), \dots, F_{kd}^{t_{l-1}}(x_d); \theta_k)} d\mathbf{x}. \end{aligned}$$

Recall that minimization of  $\Psi_2(\varphi^{t_l} | \pi, \theta, \varphi^{t_{l-1}})$  always results in  $\Psi_2(\varphi^{t_l} | \pi, \theta, \varphi^{t_{l-1}}) \leq 0$  since the choice  $f_{kj}^{t_l} = f_{kj}^{t_{l-1}}$  for all  $k = 1, \dots, K$  and  $j = 1, \dots, d$  makes this term equal to zero. Therefore, it remains to show that  $\Psi_3(\theta, \varphi^{t_l} | \pi, \theta, \varphi^{t_{l-1}}) \rightarrow 0$  as  $l \rightarrow \infty$ . To do this, let us introduce a lemma.

**Lemma 2.** For each  $k = 1, \dots, K$  and  $j = 1, \dots, d$ , the sequence  $f_{kj}^t$ ,  $t = 1, 2, \dots$  has a uniformly converging subsequence  $f_{kj}^{t_l}$ ,  $l = 1, 2, \dots$ .

The proof of [Lemma 2](#) is similar to the proof of [Lemma A2](#) in [Levine et al \(2011\)](#) and is not given. Denote by  $f_{kj}^*$  the limit of  $f_{kj}^{t_l}$  as  $l \rightarrow \infty$ . Denote by  $\varphi^*$  the collection of all such limits. Since  $\Omega$  is compact, it follows in a straightforward manner from [Lemma 2](#) that each subsequence  $F_{kj}^{t_l}(u)$  converges uniformly to  $F_{kj}^*(u) := \int_{-\infty}^u f_{kj}^*(x) dx$ . To show that  $\Psi_3(\theta, \varphi^{t_l} | \pi, \theta, \varphi^{t_{l-1}})$  goes to zero as  $l$  goes to infinity, we

proceed as follows. We have

$$\begin{aligned} & |\Psi_3(\theta, \varphi^{t_l} | \pi, \theta, \varphi^{t_l-1})| \\ & \leq \sum_{k=1}^K \left| \int g(\mathbf{x}) w_k(\mathbf{x}; \pi, \theta, \varphi^{t_l-1}) \log \frac{c(F_{k1}^{t_l}(x_1), \dots, F_{kd}^{t_l}(x_d); \theta_k)}{c(F_{k1}^{t_l-1}(x_1), \dots, F_{kd}^{t_l-1}(x_d); \theta_k)} d\mathbf{x} \right|. \end{aligned}$$

Each summand is bounded as

$$\begin{aligned} & \left| \int g(\mathbf{x}) w_k(\mathbf{x}; \pi, \theta, \varphi^*) \log \frac{c(F_{k1}^{t_l}(x_1), \dots, F_{kd}^{t_l}(x_d); \theta_k)}{c(F_{k1}^{t_l-1}(x_1), \dots, F_{kd}^{t_l-1}(x_d); \theta_k)} d\mathbf{x} \right| + \\ & \left| \int g(\mathbf{x}) (w_k(\mathbf{x}; \pi, \theta, \varphi^{t_l-1}) - w_k(\mathbf{x}; \pi, \theta, \varphi^{t_l})) \log \frac{c(F_{k1}^{t_l}(x_1), \dots, F_{kd}^{t_l}(x_d); \theta_k)}{c(F_{k1}^{t_l-1}(x_1), \dots, F_{kd}^{t_l-1}(x_d); \theta_k)} d\mathbf{x} \right|. \end{aligned} \quad (6)$$

Since the copula density is bounded from above and below, the second term is less than or equal to a constant times  $\int g(\mathbf{x}) |w_k(\mathbf{x}; \pi, \theta, \varphi^{t_l-1}) - w_k(\mathbf{x}; \pi, \theta, \varphi^{t_l})| d\mathbf{x}$ . But, by the dominated convergence theorem, this integral vanishes because the kernel  $K$  and the copula density are bounded from above and below, the copula density is Lipschitz continuous and, from [Levine et al \(2011\)](#),  $\mathcal{N}f_{kj}^{t_l}$  converges uniformly to  $\mathcal{N}f_{kj}^*$  as  $l \rightarrow \infty$ .

The first term in (6) is bounded by

$$\begin{aligned} & \left| \int g(\mathbf{x}) w_k(\mathbf{x}; \pi, \theta, \varphi^*) \log \frac{c(F_{k1}^{t_l}(x_1), \dots, F_{kd}^{t_l}(x_d); \theta_k)}{c(F_{k1}^*(x_1), \dots, F_{kd}^*(x_d); \theta_k)} d\mathbf{x} \right| \\ & + \left| \int g(\mathbf{x}) w_k(\mathbf{x}; \pi, \theta, \varphi^*) \log \frac{c(F_{k1}^*(x_1), \dots, F_{kd}^*(x_d); \theta_k)}{c(F_{k1}^{t_l-1}(x_1), \dots, F_{kd}^{t_l-1}(x_d); \theta_k)} d\mathbf{x} \right|. \end{aligned}$$

But again this bound goes to zero by similar arguments. This finishes the proof.  $\square$

## 4 Implementation

Details about the implementation of the algorithm of Section 2 are given below.

*Initialization.* For initialization, the data are partitioned into  $K$  groups by a  $k$ -means algorithm. The initial weights  $\pi^0$  are set equal to proportions of observations belonging to each group. The marginal densities  $\varphi^0$  are initialized by standard kernel density estimation methods. The marginal density of the  $k$ th group in the  $j$ th dimension is set to its kernel density estimate calculated from the projection of the data belonging to the  $k$ th group into the  $j$ th dimension. The bandwidths are specified by standard bandwidth selection methods (Silverman, 1998, p. 47–48). A value for the bandwidth of the marginal of the  $k$ th group in the  $j$ th dimension  $h_{kj}$  is selected by applying a bandwidth selection method to the projection of the data belonging to the  $k$ th group into the  $j$ th dimension. The bandwidth selection method used consists of taking  $h_{kj} = 1.06A_{kj}n_{kj}^{-1/5}$ , where  $A_{kj}$  is the minimum between the standard deviation of the data and, the interquartile range divided by 1.34 (Scott, 2015). The initial copula parameters  $\theta^0$  are set to the value corresponding to the independence copula.

*Choice of the kernel.* It is well known in kernel density estimation that the choice of the kernel has little impact on the estimates (Silverman, 1998). Therefore, the Gaussian kernel was chosen for convenience.

*Bandwidth selection.* Once the bandwidths have been initialized, they can be kept fixed or be updated from one iteration to another. In the latter, each observation  $\mathbf{x}_i$  is assigned the cluster that maximizes the current value of  $w_k(\mathbf{x}_i; \pi, \theta, \varphi)$  over  $k = 1, \dots, K$  and the same bandwidth selection method as in the initialization step is applied.

*Numerical evaluation of the integral (3).* A bottleneck of the algorithm is the numerical evaluation of the integral in (3). Indeed, the quantity  $\log f(u)$  might be close or even equal to  $-\infty$  in some regions of the integration domain. Moreover, if the values of  $K_h(x - u)$  are zero or close to zero, this may create numerical issues of the kind “ $0 \times \infty$ ”. To avoid those issues, two remedies are implemented. First, we substitute  $\max\{f(u), \varepsilon\}$  for  $f(u)$  where  $\varepsilon$  is some tolerance threshold. We arbitrarily set  $\varepsilon = 10^{-5}$ .

Second, we truncate the domain of integration. After thresholding and a change of variables, the integral to evaluate becomes  $\int_{-\infty}^{\infty} K_h(u) \log[\max\{f(x-u), \varepsilon\}] du$ . We evaluate the integral on  $(0 \pm 1.96h)$  instead of the whole real line, retaining about 95% of the mass of the kernel.

*Stopping criterion.* To terminate the algorithm, we may let the algorithm run an arbitrary number of steps and, in retrospect, visually check the convergence of the sequence of the objective function values, or we may stop the algorithm once some criterion has been reached. One possible stopping criterion is the relative increase of the objective function. That is, if  $\bar{\ell}^t = \int g(\mathbf{x}) \log \check{g}(\mathbf{x}; \pi^t, \theta^t, \varphi^t) d\mathbf{x}$  denotes the objective function to be maximized at step  $t$  of the algorithm, then the algorithm may be stopped as soon as the inequality  $|\bar{\ell}^{t+1} - \bar{\ell}^t| < \varepsilon |\bar{\ell}^t|$  occurs  $k$  times in a row. In practice, we arbitrarily set  $\varepsilon = 10^{-2}$  and  $k = 3$ .

*Choice of the number of clusters.* To estimate the number of mixture components  $K$  in the mixture model (1–2), the algorithm of Section 2 is run with several values of  $K$ . To select the “best” model, we use the pseudo-AIC criterion introduced in Mazo (2017), namely “maximum smoothed semiparametric log-likelihood times sample size minus number of copula parameters”. Note that in the definition above we need to multiply by the sample size because the smoothed semiparametric log-likelihood in (4) is defined as an expectation and hence, contrarily to Mazo (2017), the sample version is a sample average.

## 5 Simulation studies

### 5.1 A first study

Five hundred replications of four independent artificial datasets of sizes  $n = 300, 500, 700, 900$  were generated from the mixture model (1)–(2) with  $K = 3$  clusters of equal proportions, FGM copulas with parameters  $-0.5, 0.5, 0$  and marginals as in Table 1, where  $N(\mu, \sigma^2)$  and  $L(\mu, \sigma^2)$  refer to the normal and Laplace distributions



with mean  $\mu$  and standard deviation  $\sigma$ , respectively. (The density of a  $L(\mu, \sigma^2)$  distribution is then given by  $f(x) = e^{-\sqrt{2}|x-\mu|/\sigma}/(\sqrt{2}\sigma)$  for any real  $x$ .) The algorithm of Section 2 was run with  $K = 3$  to estimate the cluster proportions, the copula parameters and the marginal densities. Initialization was carried out as described in Section 4. The bandwidths were kept fixed after initialization. The algorithm was stopped after 50 iterations.

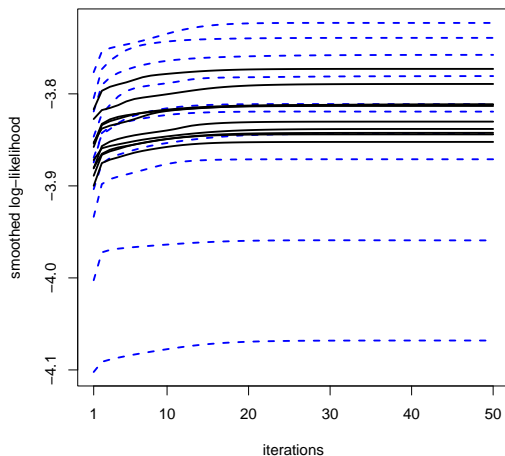
	cluster 1	cluster 2	cluster 3
dim 1	$N(-3, 2^2)$	$N(0, 0.7^2)$	$N(3, 1.4^2)$
dim 2	$L(0, 0.7^2)$	$L(3, 1.4^2)$	$L(0, 2.8^2)$

**Table 1:** Marginals used for the numerical experiment of Section 5.1.

Figure 1 shows the values of the empirical smoothed log-likelihood (4) at each step of the algorithm for the first ten replications in the case  $n = 300$  and  $n = 900$ . All of the trajectories look monotonic. It was numerically calculated that, out of the  $N = 500$  trajectories, only 17 were non-monotonic for  $n = 300$  at the  $10^{-5}$  precision. This number goes down to 1 for  $n = 500$ , and zero for  $n = 700$  and  $n = 900$ . This suggests that the algorithm of Section 2 may indeed be monotonic for the copula and marginal families chosen above.

Figure 2 shows the sum of the estimated squared biases and variances for the copula parameter vector. The variance is at least 10 times higher than the squared bias for all values of  $n = 300$ . The variance decreases with  $n$  at a rate about that of the “parametric” rate  $1/n$ : the variance at  $n = 900$  is between 2.18 and 3.02 times smaller than the variance at  $n = 300$ .

Figure 3 shows the marginal density estimates at the last step of the algorithm for  $n = 900$ , for the last replication. The estimates agree well with the true marginal densities. We noticed, however, that they were similar to the initial estimates.

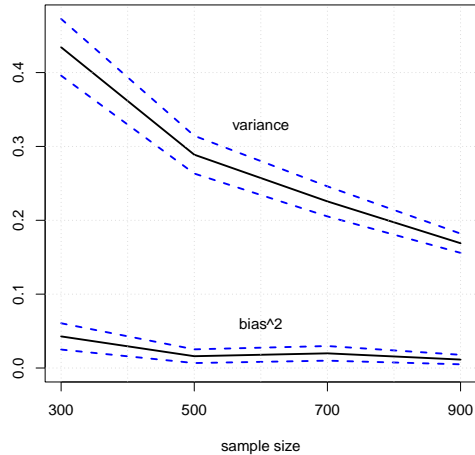


**Fig. 1:** Values of the empirical smoothed log-likelihood at each step of the algorithm, for the first ten replications. Black plain lines:  $n = 900$ . Blue dotted lines:  $n = 300$ .

## 5.2 Sensitivity to initialization

To assess the impact of initialization on the results of the algorithm of Section 2, the last dataset generated in the simulation experiment of Section 5.1 with  $n = 900$  was reused. Initialization of the algorithm was changed to a fit of a Gaussian mixture model with independent components in lieu of the  $k$ -means algorithm. In other words, in step A1 of the algorithm in Section 2, the marginals  $\varphi^0$  were set to Gaussian marginals with means and variances estimated by the Gaussian mixture model. The values of  $\pi^0$  were also obtained from the Gaussian mixture model. The other tuning and initialization parameters of the algorithm were left unchanged. The number of iterations was arbitrarily set to 30.

The estimated marginal densities at initialization and at the last iteration are shown in Figure 4. On the top row, we see that estimates at initialization are as expected: they correctly capture the salient features of the true marginals, although they are not able to reproduce non-Gaussian shapes (top row, right, compare with Figure 3). Intriguingly, the estimates have deteriorated at the last iteration of the



**Fig. 2:** Estimated squared bias and variance of the copula parameter vector estimator for various sample sizes at the last step of the algorithm. Dashed blue lines represent 95% confidence bands (aka simultaneous confidence intervals) obtained from an application of the multivariate central limit theorem to the five hundred replications.

algorithm—compare the bottom row of Figure 4 with the top row of Figure 3. This is in sharp contrast with the bottom row of Figure 3, where the estimates were good. It seems that initialization plays a key role in the final performance of the algorithm. This is confirmed by comparing the values of the three components of the estimated copula parameter vector across iterations, depicted in Figure 5. We see in Figure 5(a) that one of the sequence of estimates seems to have not converged, while the others have their values stuck at  $-1$  and  $1$ , which is in general not an indication that estimation was performed correctly. By contrast, Figure 5(b) depicts stable and reasonable estimates. In sum, a Gaussian mixture modeling step during initialization of the algorithm produced poor estimates.

### 5.3 Estimation of the number of mixture components $K$

A numerical experiment was carried out to see whether the pseudo-AIC criterion described in Section 4 is able to select the correct number of components. A number

of 500 synthetic datasets of size 300 were generated from a mixture model with three components of equal weights. The components are bivariate normal distributions with means  $(0,3)$ ,  $(3,0)$ ,  $(-3,0)$ , standard deviations  $(\sqrt{2}, 1/\sqrt{2})$ ,  $(\sqrt{2}, 1/\sqrt{2})$ ,  $(\sqrt{2}, 1/\sqrt{2})$  and correlations 0.5, 0.5, 0.5. Gaussian copulas were assumed for all components. For each dataset, the mixture model (1-2) was fitted with the algorithm of Section 2 for  $K = 2, \dots, 5$ , where  $K$  denotes the number of components of the mixture model, and the pseudo-AIC criterion of Section 4 was computed to select the number of clusters. Initialization was carried out as described in Section 4. The bandwidths were updated at each step of the algorithm. The stopping criterion described in Section 4 was used to terminate the algorithm.

The results are reported in Figure 6. Among the 500 estimates, 402 (standard error 9) were correct, and 98 were incorrect (standard errors 7 and 6 for  $K = 4$  and  $K = 5$ , respectively). The chart suggests that the pseudo-AIC criterion is reasonable. This is consistent with the findings in Mazo (2017).

## 6 Real data analysis

### 6.1 The iris dataset

The iris dataset has  $n = 150$  observations of  $d = 4$  variables (sepal and petal length and width) belonging to three groups (“setosa”, “versicolor”, “virginica”). For simplicity and illustrative purposes, only two variables were considered (sepal and petal length). The algorithm of Section 2 was run with Gaussian copulas. Initialization was carried out as described in Section 4. The stopping criterion was used to terminate the algorithm. For bandwidth selection, the two strategies described in Section 4 were tested: the first consists of keeping the bandwidth fixed after initialization and the second consists of updating the bandwidth at each step of the algorithm. For the first

strategy, the algorithm was successfully run with  $K = 2, \dots, 6$  clusters. With 7 clusters, one of the clusters became empty and the algorithm was stopped. For the second strategy, the algorithm was successfully run with  $K = 2, 3, 4$  clusters.

The values of the pseudo-AIC criterion are depicted in Figure 7. In Figure 7(a), the values increase and stabilize at  $K = 4$ . In Figure 7(b), the presence of a plateau is less clear. The pseudo-AIC increases as  $K$  increases, and hence a reasonable choice would also be  $K = 4$ . The obtained classification results are reported in Figure 8. We see a clear difference between the two bandwidth selection strategies. The classification results for the case  $K = 3$  are reported in Figure 9. Here, the results of the two bandwidth selection strategies are similar (top row) and better reflect the true partitioning of the data than the results of the Gaussian mixture model fitted with 3 clusters (bottom row). For the Gaussian mixture model, the optimal number of clusters according to the BIC criterion is two.

## 6.2 The wine dataset

To illustrate the practical performance of our method, we will apply it to the analysis of the wine dataset that has been analyzed earlier in Bouveyron et al (2019) pp. 60–65. This dataset contains 27 physical and chemical measurements on 178 wine samples of three types—Barolo, Grignolino, and Barbera. The dataset is publicly available as a part of the pgmm R package (McNicholas, 2016). Bouveyron et al (2019) conducted a preliminary principal components analysis and selected 5 variables with the highest loadings on each of the first five principal components. Moreover, they noted that just two variables—Flavonoids and Color Intensity—seem to give a strong visual suggestion of clustering, based on the pairs plot. Their analysis is based on the use of multivariate Gaussian density mixtures with various covariance matrix structures.

We conduct our analysis of this dataset based on these two variables. As a tool for selection of the number of clusters, we use the pseudo-AIC criterion introduced earlier

in Section 4. To do so, we select a range of the possible number of clusters from 2 to 8 and compute the value of pseudo-AIC for each of these choices. Gaussian copula was used to model the dependence between the two variables. Initialization was carried out as described in Section 4. The bandwidths were updated at each iteration of the algorithm. The stopping criterion described in Section 4 was used to terminate the algorithm. The result is illustrated in Figure 10.

The result suggests the choice of either  $K = 5$  or  $K = 8$  as a possible number of clusters. Since the choice of five clusters produces an obviously more parsimonious model, we proceed with it. (Note that, when using BIC as a model selection criterion, Bouveyron et al (2019) also comes up with two possible models based on either 3 or 7 clusters with different respective covariance matrix structure.) At first sight, the choice of 5 clusters does not seem to be a very reasonable one since there are only three types of wine described by this dataset. However, we will see later that, nevertheless, this solution describes the true classification quite well. The resulting classification is illustrated in Figure 11.

Note that the red group (Grignolino) has about 9 observations separated from the main cluster. These observations, that were the source of confusion for Gaussian density mixture based solutions of Bouveyron et al (2019), also present some difficulties for our approach as well. Most of these observations have been separated into a separate (green) cluster. Moreover, observations with large values of the Flavonoids indicator have become, apparently, a reason for creation of yet another (blue) cluster.

At the same time, the confusion matrix of our classification that compares it with the partition into three wine types suggests that our classification is not widely off the mark. Indeed, consider the confusion matrix given in Table 2. Note that Barbero samples are split between Clusters 1 and 4 while Barolo samples are split (with the exception of just one sample) between Clusters 3 and 5.

	Barolo	Grignolino	Barbero
Cluster 1	0	0	28
Cluster 2	1	60	0
Cluster 3	29	7	0
Cluster 4	0	2	20
Cluster 5	29	2	0

**Table 2:** Confusion matrix of the 5-clusters solution

If Clusters 1 and 4, on one hand, and Clusters 3 and 5, on the other hand, are merged, one ends up with a 3-clusters solution whose misclassification rate is only  $\frac{12}{178}$ . For comparison purposes, when [Bouveyron et al \(2019\)](#) merge the necessary clusters of their 7-clusters solution, the resulting misclassification rate is  $\frac{11}{178}$ . Even if such a merger is not contemplated, the misclassification rate of our 5-clusters solution is  $\frac{61}{178}$  which is less than 52% misclassification rate of the 7-clusters solution of [Bouveyron et al \(2019\)](#). Thus, we believe that our approach provides an adequate clustering and classification analysis of the wine dataset.

## 7 Conclusion

An algorithm was designed and implemented to estimate the parameters of copula-based semiparametric mixture models. The model considered is a very general one since it does not impose any specific structure (such as the location-scale assumption) on marginal densities. The algorithm is deterministic, and hence always returns the same result if fed with the same initial point. Good performance was obtained in illustrative numerical examples, which suggests that the algorithm may indeed be monotonic under appropriate conditions.

However, its theoretical analysis proved to be challenging and only partial results were obtained for versions of the algorithm where either the copula parameter or the marginals were fixed. A future avenue of research may consist of rejecting those updates where the smoothed log-likelihood does not increase and investigate whether convergence results of [Meyer \(1976\)](#); [Zangwill \(1969\)](#) could be applied. To simplify, the

full parametric case may first be considered. To improve the numerical implementation of the algorithm, the integral (3) may be computed using other methods, such as [Qiang \(2010\)](#).

## Acknowledgments

Michael Levine's research has been partially funded by the NSF-DMS grant # 2311103. We thank two anonymous reviewers for helpful comments that led to an improved version of the manuscript.

## Declaration

The authors have no relevant financial or non-financial interests to disclose.

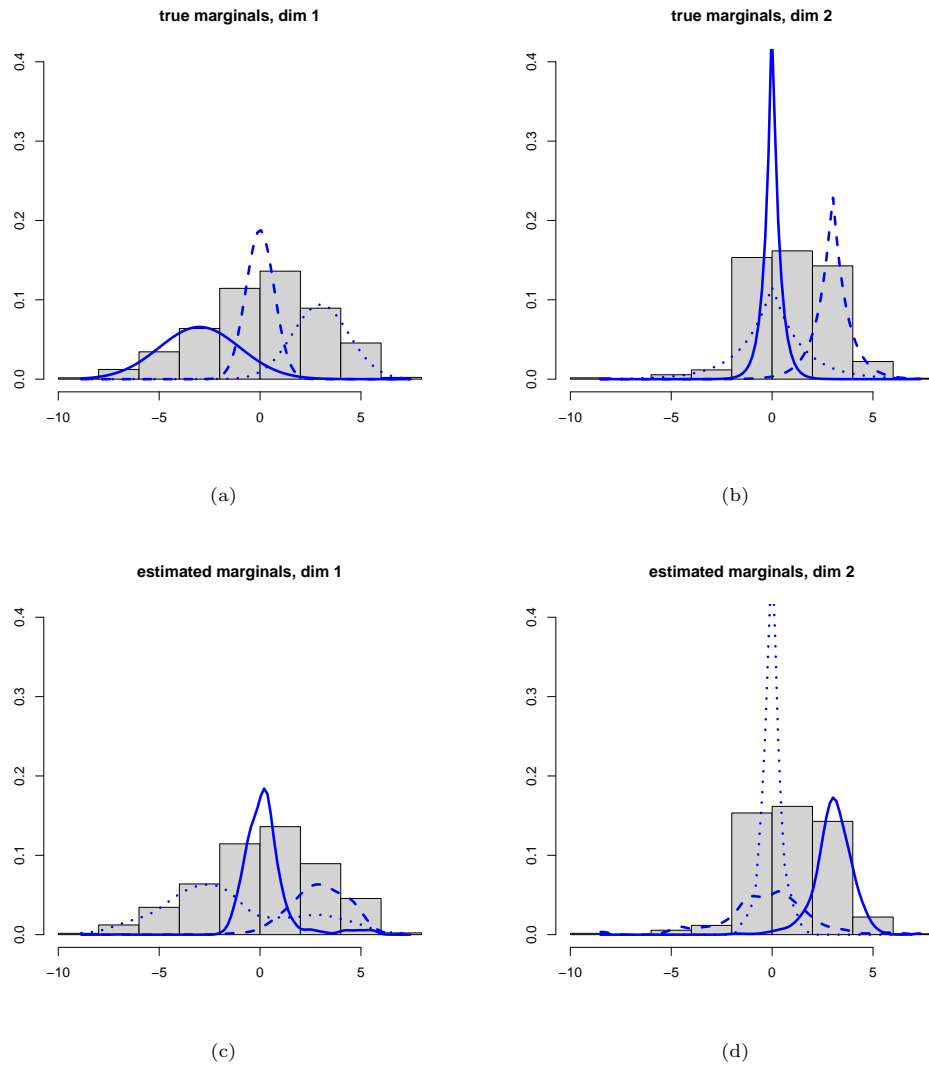
## References

- Allman ES, Matias C, Rhodes JA (2009) Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics* 37(6A):3099–3132
- Benaglia T, Chauveau D, Hunter DR (2009) An EM-like algorithm for semi-and nonparametric estimation in multivariate mixtures. *Journal of Computational and Graphical Statistics* 18(2):505–526
- Bonhomme S, Jochmans K, Robin JM (2016) Non-parametric estimation of finite mixtures from repeated measurements. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78(1):211–229
- Bouveyron C, Celeux G, Murphy TB, et al (2019) *Model-based clustering and classification for data science: with applications in R*, vol 50. Cambridge University Press

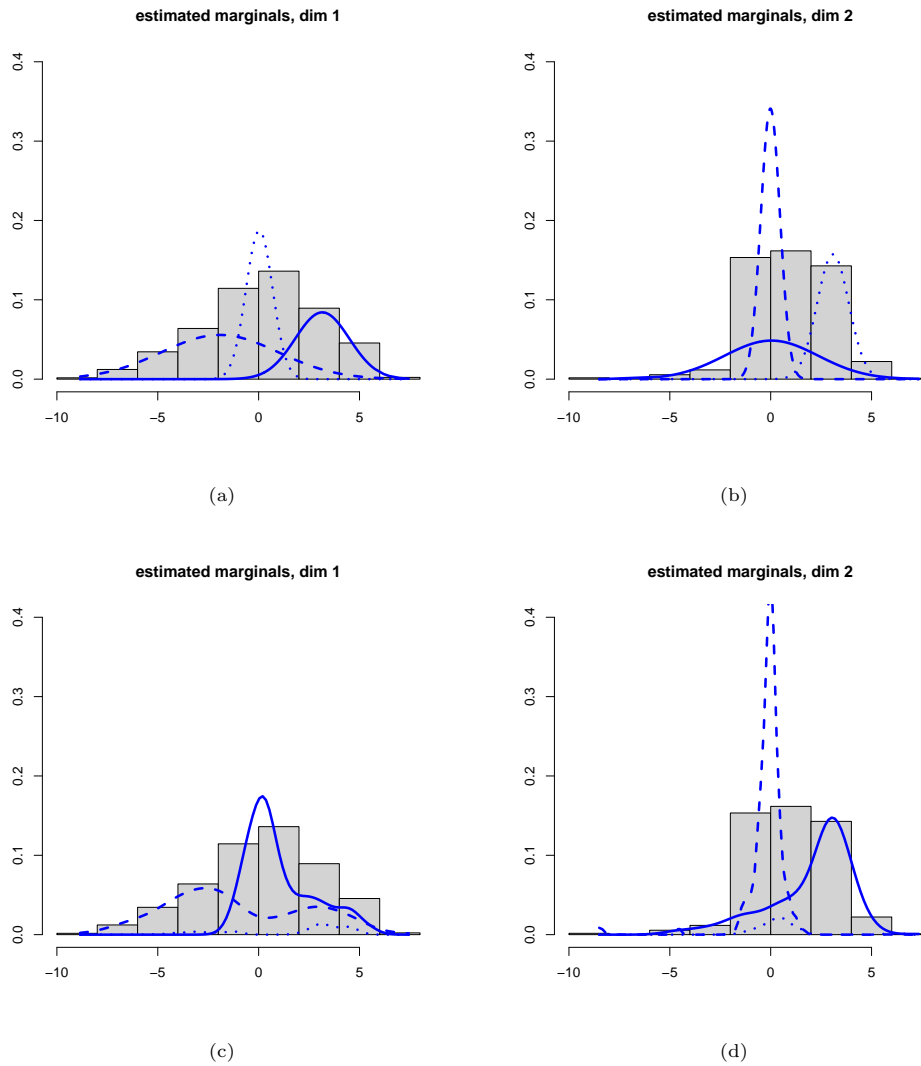


- Brezis H (2011) Functional analysis, Sobolev spaces and partial differential equations. Springer
- Hall P, Zhou XH (2003) Nonparametric estimation of component distributions in a multivariate mixture. *The Annals of statistics* 31(1):201–224
- Kasahara H, Shimotsu K (2014) Non-parametric identification and estimation of the number of components in multivariate mixtures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76(1):97–111
- Kwon C, Mbakop E (2021) Estimation of the number of components of nonparametric multivariate finite mixture models. *The Annals of Statistics* 49(4):2178–2205
- Levine M, Hunter DR, Chauveau D (2011) Maximum smoothed likelihood for multivariate mixtures. *Biometrika* 98(2):403–416
- Mazo G (2017) A semiparametric and location-shift copula-based mixture model. *Journal of Classification* 34(3):444–464
- Mazo G, Averyanov Y (2019) Constraining kernel estimators in semiparametric copula mixture models. *Computational Statistics & Data Analysis* 138:170–189
- McNicholas PD (2016) Mixture model-based classification. CRC press
- Meyer RR (1976) Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *Journal of Computer and System Sciences* 12:108–121
- Nelsen RB (2007) An introduction to copulas. Springer Science & Business Media
- Qiang J (2010) A high-order fast method for computing convolution integral with smooth kernel. *Computer Physics Communications* 181(2):313–316

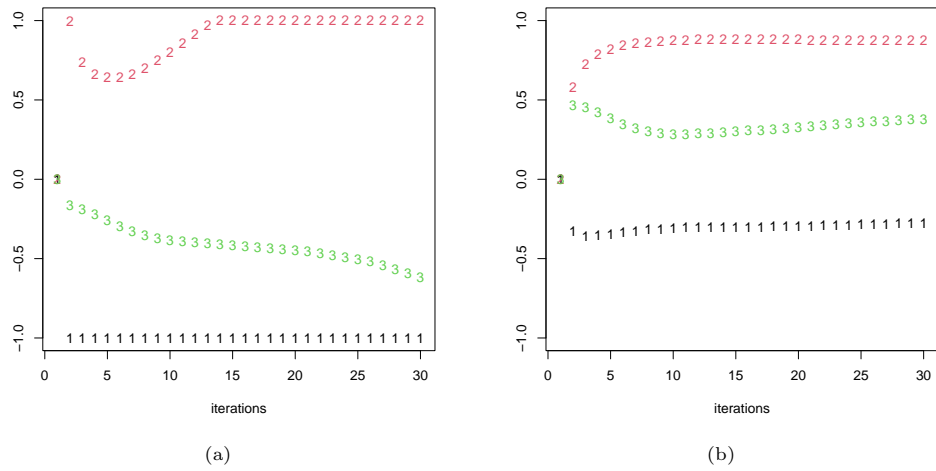
- Rau A, Maugis-Rabusseau C, Martin-Magniette ML, et al (2015) Co-expression analysis of high-throughput transcriptome sequencing data with Poisson mixture models. *Bioinformatics* 31(9):1420–1427
- Scott DW (2015) *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons
- Silverman BW (1998) *Density estimation for statistics and data analysis*. Chapman & Hall
- Vrac M, Billard L, Diday E, et al (2012) Copula analysis of mixture models. *Computational Statistics* 27:427–457
- Wu TT, Lange K (2010) The MM alternative to EM. *Statistical Science* 25(4):492–505
- Xiang S, Yao W, Yang G (2019) An Overview of Semiparametric Extensions of Finite Mixture Models. *Statistical Science* 34(3):391–404. Publisher: Institute of Mathematical Statistics
- Zangwill WI (1969) *Nonlinear Programming—A Unified Approach*. Prentice-Hall



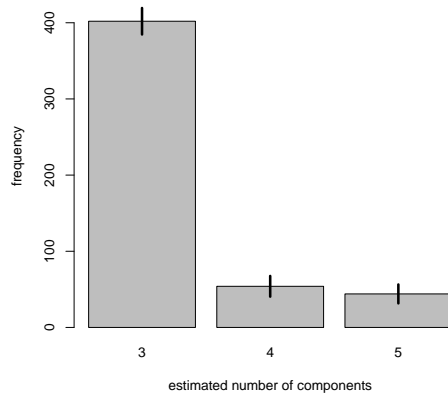
**Fig. 3:** True and estimated marginal densities of the three clusters and the two dimensions for  $n = 900$  (last replication). The top row contains the true marginals and the column on the left contains the first dimension. The marginal estimates are those found at the last step of the algorithm.



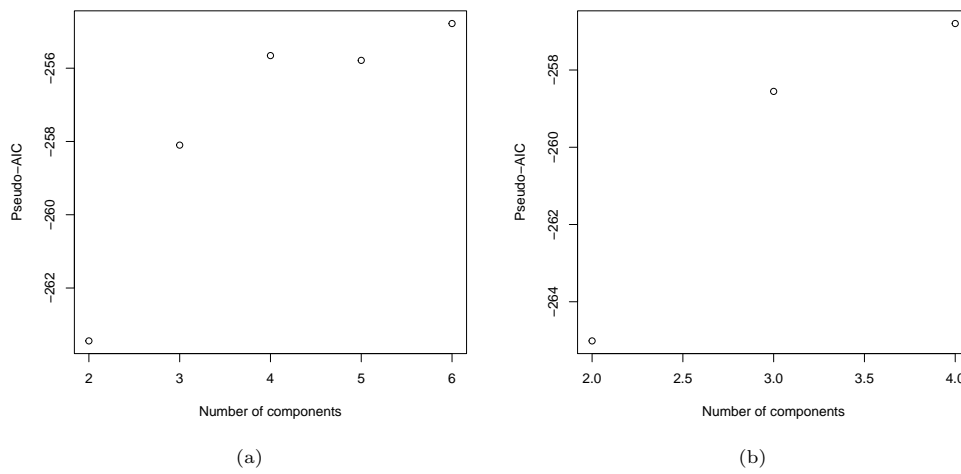
**Fig. 4:** Estimated marginal densities of the three clusters and the two dimensions for the last dataset generated in Section 5.1 with  $n = 900$  and initialization by fitting a Gaussian mixture model. The top and bottom rows contain the results after the initialization step and at the last iteration of the algorithm, respectively. The column on the left and on the right contain the first and the second dimensions, respectively.



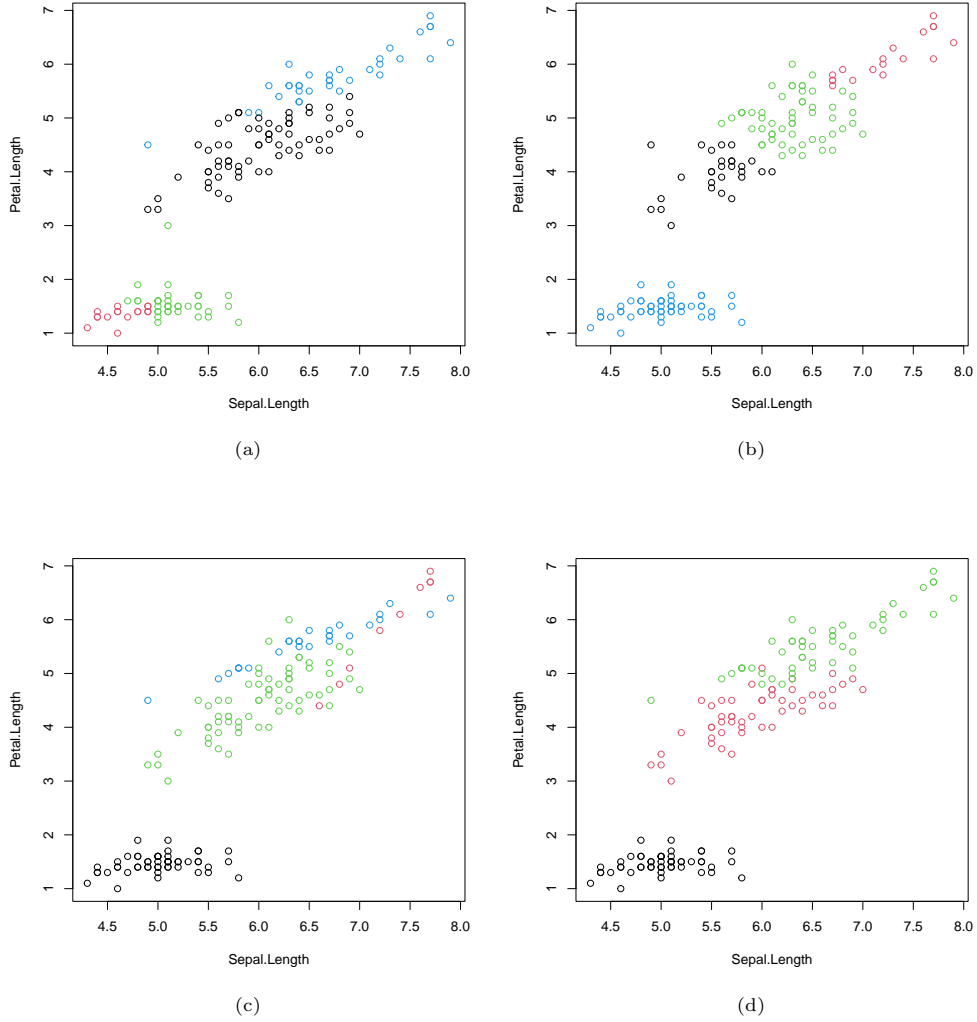
**Fig. 5:** Values of the three components of the estimated copula parameter vector across iterations for the last dataset generated in Section 5.1 with  $n = 900$  and initialization by fitting (a) a Gaussian mixture model and (b) a  $k$ -means algorithm.



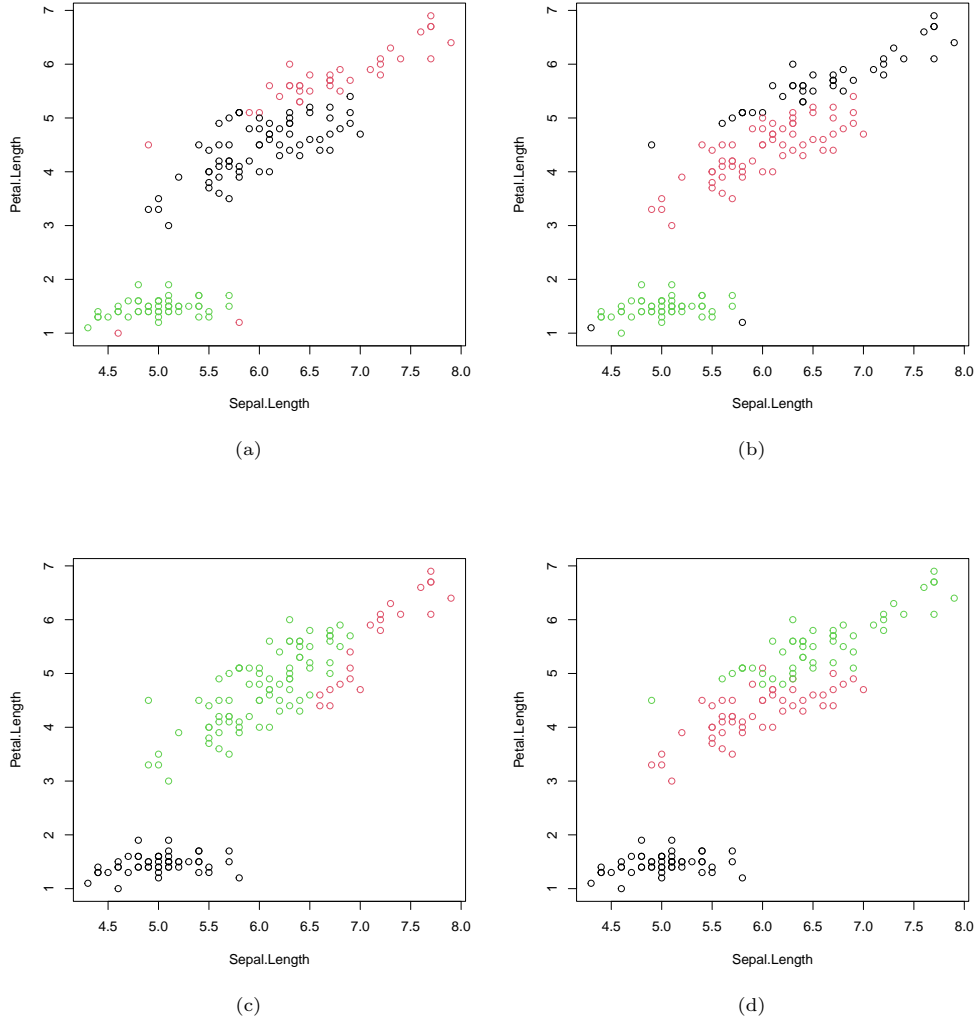
**Fig. 6:** Frequency of the various values for the estimated number of components. The frequency for  $K = 2$  is zero. The vertical bars correspond to the Monte Carlo asymptotic confidence intervals of level 95%.



**Fig. 7:** The value of pseudo-AIC model selection criterion for several values of the cluster number (iris dataset). Left: the bandwidth is kept fixed after initialization. Right: the bandwidth is updated at every step of the algorithm.

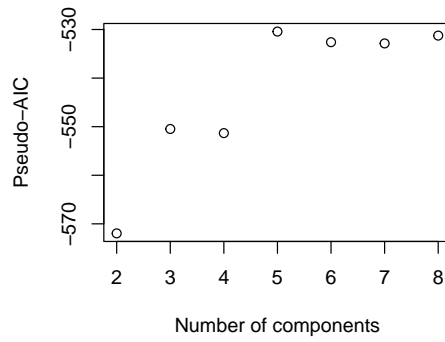


**Fig. 8:** Iris data: classification based on the choice of 4 clusters. Top left: algorithm of Section 2 with the bandwidths kept fixed after initialization. Top right: with the bandwidths updated at every step. Bottom left: results for the Gaussian mixture model with 4 clusters. Bottom right: true classification.

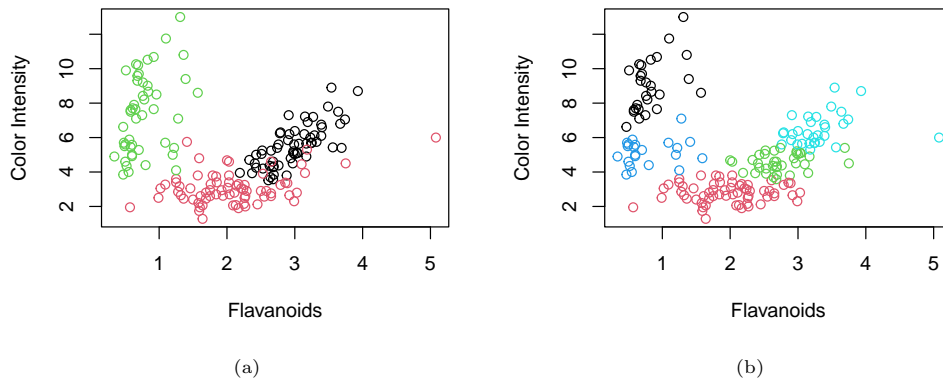


**Fig. 9:** Iris data: classification based on the choice of 3 clusters. Top left: algorithm of Section 2 with the the bandwidths kept fixed after initialization. Top right: with the bandwidths updated at every step. Bottom left: results for the Gaussian mixture model with 3 clusters. Bottom right: true classification.





**Fig. 10:** The value of pseudo-AIC model selection criterion for several values of the number of mixture components  $K$ .



**Fig. 11:** Wine data: true classification (left) and classification based on the choice of 5 clusters (right). The measurements shown are Flavanoids and Color Intensity.