Journal of Computational and Applied Mathematics 452 (2024) 116131

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Empirical density estimation based on spline quasi-interpolation with applications to copulas clustering modeling

Cristiano Tamborrino *, Antonella Falini, Francesca Mazzia

Department of Computer Science, University of Bari, Via Edoardo Orabona, 4, 70125 Bari BA, Italy

ARTICLE INFO

Keywords:
Spline quasi-interpolation
Copulas
Empirical density estimation
Clustering

ABSTRACT

Density estimation is a fundamental technique employed in various fields to model and to understand the underlying distribution of data. The primary objective of density estimation is to estimate the probability density function of a random variable. This process is particularly valuable when dealing with univariate or multivariate data and is essential for tasks such as clustering, anomaly detection, and generative modeling. In this paper we propose the monovariate approximation of the density using spline quasi interpolation and we apply it in the context of clustering modeling. The used clustering technique is based on the construction of suitable multivariate distributions which rely on the estimation of the monovariate empirical densities (marginals). Such an approximation is achieved by using the proposed spline quasi-interpolation, while the joint distributions to model the sought clustering partition is constructed with the use of copulas functions. In particular, since copulas can capture the dependence between the features of the data independently from the marginal distributions, a finite mixture copula model is proposed. The presented algorithm is validated on artificial and real datasets.

1. Introduction

Density estimation is a fundamental technique employed in various fields to model and to understand the underlying distribution of data. It plays a pivotal role in capturing the inherent patterns and structures within a dataset, making it a crucial component in statistical modeling, machine learning, and data analysis. The primary objective of density estimation is to estimate the probability density function (PDF) of a random variable. This process is particularly valuable when dealing with univariate or multivariate data and is essential for tasks such as clustering, anomaly detection, and generative modeling. Several methods have been developed to address the challenge of estimating the underlying density function. Classical approaches include histogram-based methods or kernel density estimation (KDE). These methods vary in complexity, assumptions, and computational efficiency, offering a range of options to suit different types of data and analytical requirements [1,2]. Other approaches in literature utilize splines [3–5]. Understanding the strengths and limitations of each method is crucial for selecting an appropriate density estimation technique based on the characteristics of the data at hand. In this work, we aim to investigate a novel method for estimating the probability density through the utilization of a technique known as B-spline Hermite quasi-interpolant (BSHQI) [6]. We will propose its application for the development of a new Copula-Based clustering algorithm, where density estimation plays a crucial role. Clustering is considered an effective method for organizing data into groups based on the similarities in features and characteristics among data points [7]. In recent years, various algorithms have been developed for this purpose [7–14]. For a comprehensive overview of the diverse approaches in the literature, e.g., [15–18]. One of the most well-known and widely used approaches involves finite mixture modeling,

* Corresponding author.
E-mail addresses: cristiano.tamborrino@uniba.it (C. Tamborrino), antonella.falini@uniba.it (A. Falini), francesca.mazzia@uniba.it (F. Mazzia).

https://doi.org/10.1016/j.cam.2024.116131
Received 15 December 2023; Received in revised form 13 May 2024
Available online 14 July 2024

which serves as a flexible and robust probabilistic tool for both univariate and multivariate data. However, it is important to note that Gaussian distributions, commonly used in finite mixtures, may not always accurately represent real-world data aggregation. To overcome this drawback, the use of alternative distributions, based on copulas, has gained significant interest in recent times for their potential to enhance the accuracy and applicability of clustering algorithms in a wider range of scenarios.

Copulas offer an alternative and increasingly popular approach to modeling data dependencies and aggregations [19–21]. They provide a more versatile framework that does not rely on specific distributional assumptions like Gaussian distributions. Copulas can capture complex and non-linear relationships between variables, making them particularly useful for situations where data aggregation behaviors deviate from traditional distributions. The copulas have already been used as a possible solution to the clustering problem, for example, the authors in [12] address the problem of clusterization by using different families of copulas. In [22,23], the clusterization process is done by the assumption that the multivariate dependencies are modeled by a new family of copula called Gaussian Mixture Copula, in [13] the authors analyze models of finite mixtures with different families of copulas, assuming that the marginals follow known distributions. The widely used method to fit the finite mixture model to observed data is the expectation–maximization (EM) algorithm [24,25], to estimate the maximum likelihood. Variations and/or adaptations to special situations of the EM algorithm exist, as the stochastic EM (SEM) algorithm (e.g., [26–28]), the classification EM (CEM) algorithm (e.g.[11]), the Monte CarloEM (MCEM) algorithm (e.g., [29,30]) and those developed by [31,32]. An important point to take into account is that the choice of the copula model-based clustering imposes distributional assumptions on the marginals, along each dimension, and these marginal distributions are assumed or forced to be identical (e.g. a multivariate normal imposes univariate normal distribution on each marginal); such assumptions restrict the modeling flexibility. These restrictive assumptions could lead to erroneous modeling. To address this issue, a semiparametric approach is employed, in which marginal distributions are empirically estimated using kernel density estimation (KDE) [33,34]. In this work, we propose a new strategy that enables the estimation of marginal densities through the use of the introduced BSHQI, a quasi interpolant operator that has been deeply studied in the context of ODEs and time series analysis [6,35]. In place of a general interpolation technique, the chosen QI is computationally less expensive as it relies on a pre-computation of the expression of the needed coefficients and a Python and Matlab implementation is freely available [36]. The chosen QI, being of Hermite type, ensures to construct a continuous model which is “shape-preserving”, a fundamental requirement in order to approximate a probability density function that should reflect the underlying distribution of the given observations. Moreover, we describe a mixture model for density estimation based on Copulas that allows us to automatically choose a different Copula for each cluster. The use of the QI enables clustering based on copulas to be rather competitive also in the multi-dimensional case. As such, usually copulas-based clustering techniques tend to be rather computationally expensive and therefore this limits the range of their applicability to 2D–3D cases. The paper is organized as follows, in Section 2, the BSHQI density estimation is described together with its theoretical consistency properties and some statistical tests are conducted to validate the proved theoretical results. In Section 3 we revise some preliminary concepts related to copulas. In Section 4 the used EM algorithm for Copula Mixture models is detailed. In Section 5 artificial and real datasets are analyzed and finally some conclusive remarks are presented in Section 6.

2. BSHQI density estimation

Density estimation is a fundamental task in statistical analysis, involving the determination of the underlying probability distribution for a set of observed data.

Let $X_1, \dots, X_n$ be independent and identically distributed (i.i.d) random variables with an unknown Cumulative Distribution Function (CDF) $F(x)$. A non parametric estimator for $F(x)$ is provided by the Empirical Cumulative Distribution Function (ECDF) $F_n(x) = \sum_{i=1}^n I(X_i \leq x)/n$, where $I(X_i \leq x)$ is the indicator function, equal to 1 if $X_i \leq x$, and equal to 0 otherwise. However, the ECDF is discontinuous as it jumps with size $1/n$ when $x = X_i, i = 1 \dots, n$ and this is inconvenient since, often, the CDF itself is a continuous function. The information given by the ECDF is the starting point to estimate the PDF. Widely used methods to obtain an estimator of the PDF are based on Kernel Density Estimation (KDE) [37–39]. KDE employs a kernel function, which serves as a “base shape”, to estimate the density of the data distribution. There are various methods and kernels available for KDE, each with its own characteristics. The choice of kernel and method can significantly influence the accuracy of the density estimation. Some common kernels include the Gaussian kernel, the rectangular kernel, the Epanechnikov kernel, and others [2]. The most used kernel function is the uniform kernel:

$$K(x) = \begin{cases} 1 & \text{if } x \in [-1/2, 1/2], \\ 0 & \text{otherwise.} \end{cases}$$

and the resulting estimation of the density is called *naive* kernel $\hat{f}_K(x)$, [1]:

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

where h is the bandwidth parameter and its value affects the width of the KDE curves, and consequently the accuracy of the estimation.

We propose to estimate the CDF by applying the B-spline Hermite quasi-interpolant [6] to compute $\hat{f}(x)$, approximation of the probability density $f(x) = F'(x)$, so that the final approximation $\hat{F}(x)$ of the CDF will be given by integrating $\hat{f}(x)$.

Quasi-interpolation is a technique that allows to construct a local approximant by keeping low the computational cost, see e.g., [40,41] and references therein. Generally, the common way to express a univariate spline quasi-interpolant (QI) of d-degree reads as,

$$Q_d\ f(\cdot) = \sum_{j=-d}^{N-1} \lambda_j(f) B_{j,d}(\cdot), \tag{2}$$

where $B_{j,d}$ are d-degree B-splines assumed to be defined on an extended knot vector $\tau := \{\tau_{-d}, \dots, \tau_{N+d}\}$, $\tau_j \le \tau_{j+1}$, and spanning the space, $\mathbb{S}_d^{\pi} := \langle B_{-d,d}, \dots, B_{N-1,d}\rangle$. The local linear functionals λ_j in (2) can be computed by using several methodologies, such as differential, integral methods, and discrete approaches see,e.g., [41–44]. The main advantage of QI is that it has a direct construction without solving big linear systems. Moreover, it is local, in the sense that the value of $Q_d f(x)$ depends only on values of f in a neighborhood of x.

Given an interval $[a,b]$ such that $X_i \in [a,b]$ for $i = 1, \dots, n$ and a uniform mesh $\pi = \{a = x_0, x_1, \dots, x_N = b\}$ defined by a constant stepsize $h = (b-a)/N$. Note that the choice of h is important and depends on n as it plays the bandwidth role for the kernel density. For the estimation approximation of the PDF we use the B-spline Hermite quasi-interpolant BSHQI defined in [6]. BSHQI computes the $\lambda_j(f)$ as a linear combination of the function f and its derivatives evaluated at the mesh points.

We define the BSHQI with uniform knot vector π, coincident auxiliary knots and $d = 2$. Hence, in the following to ease the notation, $B_{j,d} = B_j$. A discrete approximation of the sought CDF is expressed as,

$$F_h(x_j) := \frac{1}{n}\sum_{i=1}^{n} I(X_i \le x_j), \qquad j = 0, \dots, N. \tag{3}$$

Starting from $F_h(x)$ it is possible to approximate $f(x)$ at the mesh points by computing the first derivative using finite differences:

$$\begin{aligned} &f(x_j) = F'(x_j) \approx F'_{h,j} = \frac{F_h(x_{j+1}) - F_h(x_{j-1})}{2h}, \qquad j = 1, \dots, N-1, \\ &f(x_0) = F'(x_0) \approx F'_{h,0} = \frac{F_h(x_1) - F_h(x_0)}{h}, \qquad f(x_N) = F'(x_N) \approx F'_{h,N} = \frac{F_h(x_N) - F_h(x_{N-1})}{h}. \end{aligned} \tag{4}$$

Since the used quasi-interpolant is of Hermite type we also need an approximation of the first derivative at the same mesh points:

$$\begin{aligned} &f'(x_j) = F''(x_j) \approx F''_{h,j} = \frac{F_h(x_{j+1}) - 2F_h(x_j) + F_h(x_{j-1})}{h^2}, \qquad j = 1, \dots, N-1, \\ &f'(x_0) = F''(x_0) \approx F''_{h,0} = 0, \qquad f'(x_N) = F''(x_N) \approx F''_{h,N} = 0. \end{aligned} \tag{5}$$

Note that the values attained by the density at x_0 and x_N has been set to zero, as this is the expected value of the sought CDF. From the definition of the coefficients of BSHQI for $d = 2$, we get

$$\begin{aligned} &\lambda_j = \tfrac{1}{2}\left(F'_{h,(j+1)} + F'_{h,(j+2)}\right) - \tfrac{1}{4}h\left(-F''_{h,(j+1)} + F''_{h,(j+2)}\right), \qquad j = -1, \dots, N-2, \\ &\lambda_{-2} = F'_{h,0}, \quad \lambda_{N-1} = F'_{h,N}. \end{aligned} \tag{6}$$

The following theorem proves that the constructed $\hat{f}$ is indeed a continuous density function having first derivative continuous as well.

Theorem 2.1. *The function $\hat{f}$, BSHQI estimation of f in a given interval $[a,b]$,*

$$\hat{f}(\cdot) = \sum_{j=-d}^{N-1} \lambda_j(f) B_j(\cdot), \tag{7}$$

with λ_j as defined in (6), is a density function. In particular:

(a) $\hat{f}(\cdot) \ge 0$,
(b) $\int_{-\infty}^{+\infty} \hat{f}(x)\,dx = 1$,
(c) $\hat{f} \in C^1[a,b]$.

Proof. To prove (a) it is sufficient to show that $\lambda_j \ge 0$, for $j = -d, \dots, N-1$. In particular, since h is constant, setting

$$F_j := F_h(x_j),$$

it can be shown that

$$\begin{aligned} &\lambda_j = \frac{F_{(j+2)} - F_{(j+1)}}{h}, \qquad j = -1, \dots, N-2, \\ &\lambda_{-2} = \lambda_{-1} \qquad \lambda_{N-1} = \lambda_{N-2}. \end{aligned} \tag{8}$$

by substituting (4) and (5) into Eq. (6). Therefore, it is straightforward to see that they are always positive.

To prove (b), we set $\hat{f}(x) = 0$ outside the interval $[a,b]$. Then, knowing that the integral of a B-spline is given by

$$\int_a^b B_{i,d}(x)\,dx = \int_{\tau_i}^{\tau_{i+d+1}} B_{i,d}(x)\,dx = \frac{\tau_{i+d+1}-\tau_i}{d+1},$$

recalling that $d=2$, we have

$$\tau_{i+d+1}-\tau_i = \begin{cases} 3h & \text{for} \quad i=0,\dots,N-3, \\ 2h & \text{for} \quad i=-2, N-2, \\ h & \text{for} \quad i=-1, N-1. \end{cases}$$

Therefore,

$$\begin{aligned} &\int_{-\infty}^{+\infty} \hat{f}(x)\,dx = \int_a^b \sum_{j=-d}^{N-1} \lambda_j(f) B_j(x)\,dx = \sum_{j=-d}^{N-1} \lambda_j(f) \int_{\tau_j}^{\tau_{j+d+1}} B_j(x)\,dx = \\ &= \frac{F_1-F_0}{h}\frac{h}{3} + \frac{F_1-F_0}{h}\frac{2h}{3} + \frac{F_2-F_1}{h}h + \cdots + \frac{F_{N-3}-F_{N-2}}{h}h + \frac{F_{N-1}-F_N}{h}\frac{2h}{3} + \frac{F_{N-1}-F_N}{h}\frac{h}{3} \\ &= -F_0 + F_N = 0+1 = 1. \end{aligned}$$

The point (c) descends from the properties of the B-spline functions of degree 2. □

In the following we investigate the consistency of the derived density function following the analysis proposed in [3]. In particular, the next Lemma will be useful as a preliminary result.

Lemma 2.2. *The coefficients in (8) can be obtained by evaluating $\hat{f}_K(x)$ at $c_j := x_{j+1} + h/2, j=-1,\dots,N-2$, i.e.:*

$$\lambda_j = \hat{f}_K(c_j) = \frac{1}{nh}\sum_{i=1}^n K\left(\frac{X_i-c_j}{h}\right), \qquad j=-1,\dots,N-2,$$

and

$$\lambda_{-2} = \lambda_{-1}, \qquad \lambda_{N-1} = \lambda_{N-2}.$$

Theorem 2.3. *Let $X_1,\dots,X_n$ denote i.i.d. observations having a PDF $f(x) \in C^1[a,b]$, f and f' bounded, and let $B_j(x)$ denote the jth, 2-nd degree B-spline basis. Let $x \in [a,b]$ and let $\hat{f}_n(x)$ be as in (7), with uniform mesh defined in $[a,b]$ choosing a constant h such that as $n\to\infty$, as $nh\to\infty$ and as $h\to 0$, then $\hat{f}_n$ is a uniformly consistent estimator of f.*

Proof. To prove the pointwise consistency of $\hat{f}_n$ it is necessary to show that the $MSE(\hat{f}_n(x)) \to 0$ as $n\to\infty$, as $nh\to\infty$ and as $h\to 0$. In the following we use the well-known "Bias-Variance" formulation:

$$MSE(\hat{f}_n(x)) \equiv E\left[\left|\hat{f}_n(x)-f(x)\right|^2\right] = \text{Var}(\hat{f}_n(x)) + \left[E(\hat{f}_n(x))-f(x)\right]^2 \equiv \text{Var}(\hat{f}_n(x)) + \text{Bias}^2(\hat{f}_n(x)).$$

We start considering the absolute value of the Bias:

$$\begin{aligned} \left|\text{Bias}(\hat{f}_n(x))\right| &= \left|E\left(\frac{1}{nh}\sum_j \lambda_j B_j(x)\right) - f(x)\right| \\ &= \left|E\left(\frac{1}{nh}\sum_j\sum_{i=1}^n K\left(\frac{X_i-c_j}{h}\right)B_j(x)\right) - f(x)\right| \\ &= \left|\frac{1}{h}\sum_j B_j(x) E\left(K\left(\frac{X-c_j}{h}\right)\right) - f(x)\right| \\ &= \left|\frac{1}{h}\sum_j B_j(x)\left(\int f(X)K\left(\frac{X-c_j}{h}\right)dX\right) - f(x)\left(\int \frac{1}{h}\sum_j K\left(\frac{X-c_j}{h}\right)B_j(x)dX\right)\right| \end{aligned}$$

since $\int \frac{1}{h}\sum_j K\left(\frac{X-c_j}{h}\right)B_j(x)dX = 1$ and denoting with $I_{x,h} = [x_{j+1}, x_{j+2}] = supp\{B_j(x)\} \cap supp\{K\left(\frac{X-c_j}{h}\right)\}$ we have,

$$\begin{aligned} &= \left|\frac{1}{h}\sum_j B_j(x)\left(\int f(X)K\left(\frac{X-c_j}{h}\right)dX\right) - f(x)\frac{1}{h}\sum_j B_j(x)\int K\left(\frac{X-c_j}{h}\right)dX\right| \\ &= \left|\frac{1}{h}\sum_j B_j(x)\left(\int (f(X)-f(x))K\left(\frac{X-c_j}{h}\right)dX\right)\right| \\ &\le \sup_{X\in I_{x,h}} |(f(X)-f(x))|\frac{1}{h}\sum_j B_j(x)\int K\left(\frac{X-c_j}{h}\right)dX = \sup_{X\in I_{x,h}} |f(X)-f(x)| \\ &\le \sup_{\xi\in I_{x,h}} |f'(\xi)|h \end{aligned}$$

therefore

$$\mathrm{Bias}^2\left(\hat{f}_n(x)\right) \leq h^2 \left(\sup_{\xi \in I_{x,h}} |f'(\xi)| \right)^2 .$$

Considering now the variance we deduce that:

$$\begin{aligned}
\mathrm{Var}(\hat{f}_n(x)) &= \mathrm{Var}\left(\sum_j \lambda_j B_j(x) \right) = \\
&= \mathrm{Var}\left(\frac{1}{nh} \sum_j \sum_{i=1}^{n} K\left(\frac{X_i - c_j}{h} \right) B_j(x) \right) = \\
&= \mathbf{E}\left[\left(\frac{1}{nh} \sum_j \sum_{i=1}^{n} K\left(\frac{X_i - c_j}{h} \right) B_j(x) \right)^2 \right] - \left(\mathbf{E}\left[\frac{1}{nh} \sum_j \sum_{i=1}^{n} K\left(\frac{X_i - c_j}{h} \right) B_j(x) \right] \right)^2 \\
&= \frac{1}{nh^2} \sum_j \sum_z B_j(x) B_z(x) \int K\left(\frac{X - c_j}{h} \right) K\left(\frac{X - c_z}{h} \right) f(X) dX - \\
&\frac{1}{nh^2} \sum_j \sum_z B_j(x) B_z(x) \int K\left(\frac{X - c_j}{h} \right) f(X) dX \int K\left(\frac{X - c_z}{h} \right) f(X) dX \\
&= \frac{1}{nh^2} \sum_j \sum_z B_j(x) B_z(x) \\
&\left(\int K\left(\frac{X - c_j}{h} \right) K\left(\frac{X - c_z}{h} \right) f(X) dX - \int K\left(\frac{X - c_j}{h} \right) f(X) dX \int K\left(\frac{X - c_z}{h} \right) f(X) dX \right)
\end{aligned}$$

since the kernel K is evaluated at the mesh points it is easy to prove that this quantity is bounded and

$$h = \int K\left(\frac{X - c_j}{h} \right) K\left(\frac{X - c_z}{h} \right) dX,$$

hence,

$$\begin{aligned}
\mathrm{Var}(\hat{f}_n(x)) = &\leq \frac{1}{nh^2} \sum_j \sum_z B_j(x) B_z(x) \left(\sup_{X \in \mathbb{R}} |f(X)| h + \sup_{X \in \mathbb{R}} (f(X))^2 \left(\int K\left(\frac{X - c_j}{h} \right) dX \right) \left(\int K\left(\frac{X - c_z}{h} \right) dX \right) \right) \leq \\
&\leq \frac{1}{nh^2} \sum_j \sum_z B_j(x) B_z(x) \left(\sup_{X \in \mathbb{R}} |f(X)| h + \sup_{X \in \mathbb{R}} (f(X))^2 h^2 \right) \qquad \square \\
&= \frac{1}{nh} \left(\sup_{X \in \mathbb{R}} |f(X)| + \sup_{X \in \mathbb{R}} (f(X))^2 h \right) \left(\sum_j B_j(x) \right) \left(\sum_z B_z(x) \right) \\
&= \frac{1}{nh} \sup_{X \in \mathbb{R}} |f(X)| + \frac{1}{n} \sup_{X \in \mathbb{R}} (f(X))^2 .
\end{aligned}$$

Then we have

$$\mathrm{MSE}(\hat{f}_n(x)) = \mathrm{Bias}^2(\hat{f}_n(x)) + \mathrm{Var}(\hat{f}_n(x)) \leq h^2 \left(\sup_{\xi \in I_{x,h}} |f'(\xi)| \right)^2 + \frac{1}{nh} \sup_{X \in \mathbb{R}} |f(X)| + \frac{1}{n} \sup_{X \in \mathbb{R}} (f(X))^2 . \tag{9}$$

This quantity will tend to zero as $n \to \infty$, as $nh \to \infty$, and as $h \to 0$.

Since $\sup_{\xi \in I_{x,h}} |f'(\xi)| \leq S_1 := \sup_{\xi \in [a,b]} |f'(\xi)|$, denoting by $S_0 := \sup_{X \in [a,b]} |f(X)|$, then, the upper bound for the MSE can be written as:

$$MSE(\hat{f}_n(x)) \leq \mathrm{Bias}^2(\hat{f}_n(x)) + \mathrm{Var}(\hat{f}_n(x)) \leq h^2 (S_1)^2 + \frac{1}{nh} S_0 + \frac{1}{n} S_0^2 .$$

The above upper bound does not depend on x as so the uniform consistency is proved. □

In the BSHQI density estimation, the bandwidth h can be freely chosen as long as the assumption of Theorem 2.3 are satisfied. Thus, the optimal bandwidth h can be chosen by minimizing the MSE neglecting the smallest term $\frac{1}{n} S_0^2$:

$$h_{\mathrm{opt}}(x) := \left(\frac{1}{2n} \frac{S_0}{(S_1)^2} \right)^{\frac{1}{3}} \sim n^{-1/3} . \tag{10}$$

This choice for the smoothing bandwidth leads to an MSE at the rate

$$\mathrm{MSE}_{\mathrm{opt}}(\hat{f}_n(x)) = O(n^{-\frac{2}{3}}) . \tag{11}$$

In the previous analysis, our focus was solely on a single point, x. However, in a broader context, our goal is to manage the overall MSE for every point. In such cases, a straightforward extension is the mean integrated square error (MISE) of $\hat{f}_n(x)$. We have the following corollary:

Table 1
Statistics ran on the results for Normal Density Estimation—Rice's Rule for bins.

	AMISE	RMSE	KS-Test		Cramér–von Mises	
			statistic	p-value	statistic	p-value
BSHQI	3.43e−06	1.13e−04	7.75e−03	2.77e−01	2.05e−01	2.58e−01
KDEpy	1.08e−05	3.53e−04	1.21e−02	1.58e−02	9.08e−01	4.05e−03

Table 2
Statistics ran on the results for Exponential Density Estimation—Rice's Rule for the bins.

	AMISE	RMSE	KS-Test		Cramér–von Mises	
			statistic	p-value	statistic	p-value
EMP_BSHQI	2.27e−06	2.96e−05	7.78e−03	2.73e−01	1.18e−01	5.02e−01
EMP_KDEpy	1.68e−04	2.19e−03	6.96e−02	0	5.79e+01	2.01e−08

Corollary 2.4. *Let f be a probability density function on $\mathbb{R}$, and let $A \subseteq \mathbb{R}$ be an open region with $A = \{x \mid f(x) \neq 0\}$, where $f \in C^1(\mathbb{R})$, and both f and f' are bounded. If A is contained in a closed and bounded region, then, with $\hat{f}_n$ as defined above, if $h \to 0$ and $nh \to \infty$ as $n \to \infty$, then*

$$MISE(\hat{f}_n) = \int MSE(\hat{f}_n(x))\,dx = \int \mathbf{E}\left(\hat{f}_n(x) - f(x)\right)^2 dx \to 0,$$

as $n \to \infty$, i.e., $\hat{f}_n$ is a consistent estimator of f in mean integrated squared error (MISE).

The above evidence concludes the investigation of the consistency results of the B-spline estimator with the proposed approach. However, in this context, how to choose h is an unsolved problem in statistics known as *bandwidth selection*. Most bandwidth selection approaches either suggest an estimate of AMISE and then aim to minimize the estimated AMISE. For more details, we refer to [45].

Observation 2.5. *Given the weighted CDF*

$$F_{h,w}(x_j) = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i I(X_i \leq x_j), \qquad j = 0, \dots, N, \tag{12}$$

with w_i a weight associated to the corresponding observation X_i, then the results of Theorems 2.1 and 2.3 continue to hold.

Observation 2.6. *Note that the continuous CDF $\hat{F}$ is then computed by integrating the density $\hat{f}$ in Eq. (7).*

2.1. Statistical tests for marginals fitting with BSHQI spline

In this subsection, we compare the BSHQI density estimation with the classical empirical approach constructed by using the Gaussian Kernel Density function. The careful selection of the number of bins is of crucial importance in approximating density, as it directly influences the accuracy and the visual representation of the data distribution pattern. For both procedures considered in this work, it is possible to select different criteria for choosing the number of bins, in particular, we consider the so called *Rice's Rule* [46], where the number of bins is equal to $2 \times \lceil n^{1/3} \rceil$.

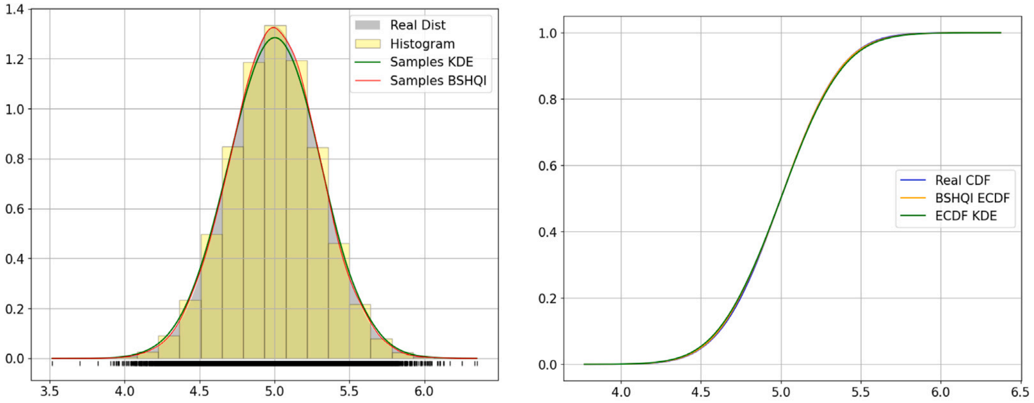
There is no single optimal criterion for selecting the most suitable bins. For the experiments conducted in this work, unless otherwise indicated, we will use the Rice rule. To assess the goodness of the produced model, we conduct two statistical tests: the Kolmogorov–Smirnov (KS) Test [47] and the Cramér–von Mises (CvM) Test [48]. Additionally, we show the error in terms of Average Mean Integrated Squared Error (AMISE) and Root Mean Square Error (RMSE) for the computed probability density functions.

We perform the tests on three different distributions: a normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$ with $\mu = 5$ and $\sigma^2 = 0.3$, an exponential distribution $X \sim \text{Exp}(\lambda)$ with $\lambda = 1$ and a third distribution consisting of a mixture of Gaussians with different means and variances.

All the numerical experiments are performed using Python 3.10 on a computing system equipped with Windows 11 operating system, 16 GB of RAM, and powered by an Intel(R) Core(TM) i7-9750H CPU @ 2.60 GHz with a base clock speed of 2.59 GHz. The Python package KDEpy[1] has been chosen for the comparison since, in our opinion, it is the most efficient among the available Python routines for this task; the Kernel density estimation is constructed following the theory in [49].

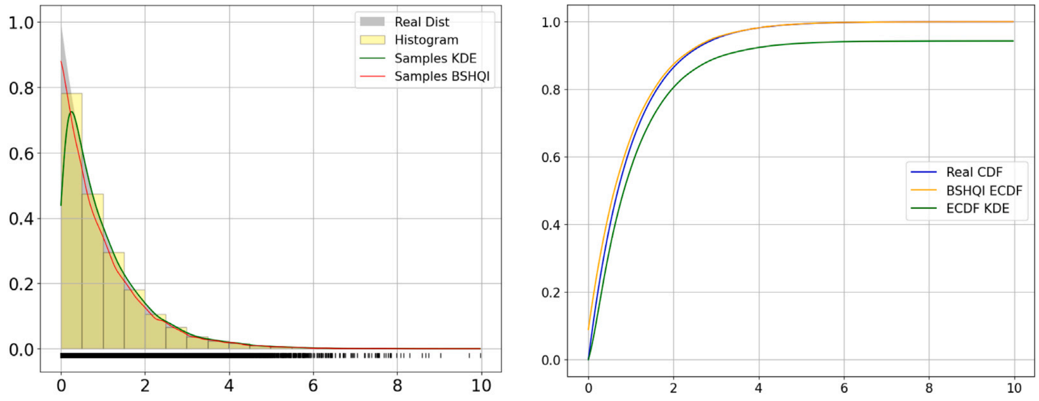
For all the considered distributions, we generated a group of $n = 2^{15}$ samplings for 20 iterations. This iterative process allows us to calculate the AMISE, the RMSE, the values of both statistics and relative p-values, and we derive their average values as

[1] https://kdepy.readthedocs.io/en/latest/introduction.html



(a) Pdf (b) Cdf

Fig. 1. Comparison of samples generated from $X \sim \mathcal{N}(5, 0.3)$ with the KDEpy and BSHQI method for probability density (a) and for the cumulative distribution (b).



(a) Pdf (b) Cdf

Fig. 2. Comparison of samples generated with the KDE and BSHQI method for probability density (a) and for the cumulative distribution (b).

Table 3
Statistics ran on the results for Mixture Gaussian Density Estimation—Rice's Rule for the bins.

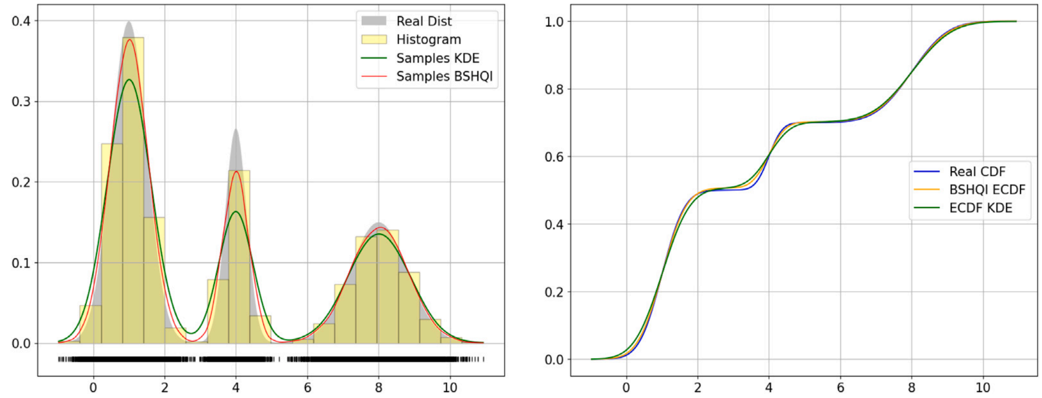
	AMISE	RMSE	KS-Test		Cramér–von Mises	
			statistic	p-value	statistic	p-value
EMP_BSHQI	1.84e−06	2.16e−05	4.91e−03	8.22e−01	8.09e−02	6.87e−01
EMP_KDEpy	9.35e−06	1.10e−04	1.06e−02	5.04e−02	5.18e−01	3.58e−02

comprehensive measures of performance. Furthermore, we evaluated efficiency in terms of computational time by calculating the mean and standard deviation after the set number of iterations. Regarding the statistical tests, the null hypothesis states that the true underlying distribution and the empirical one are identical; the alternative hypothesis suggests that they are not. The used statistics is the maximum absolute difference between the exact values and the ones computed by the empirical distribution functions at the same samples. If the KS or CvM statistics are large, then the *p*-value will be small, and this may be taken as evidence against the null hypothesis in favor of the alternative.

By observing the Tables 1, 2, 3 it can be seen that the density approximation with BSHQI is preferable to the classical empirical evaluation of the distribution taken into account.

Indeed, comparing the p-values of both tests in each of the Tables 1, 2, 3, allows us to accept the hypothesis when using the BSHQI, contrary to what can be concluded when referring to KDEpy.

Moreover, for all three experiments, the AMISE and RMSE obtained with the proposed approach are lower compared to the AMISE and RMSE obtained using KDEpy. This can be observed in Figs. 1, 2, and 3, where the estimates of the PDF and the CDF for



(a) Pdf

(b) Cdf

Fig. 3. Comparison of samples generated with the KDE and BSHQI method for probability density (a) and for the cumulative distribution (b).

Table 4
Comparison of Execution Times for KDE and BSQH Algorithms.

Algorithm	Mean time (ms)	Standard deviation (ms)
KDEpy	0.0241	±0.0011
BSHQI	0.0124	±0.0009

both methods are compared to the real distribution. Note that different results could be obtained by choosing a different rule for the number of bins. In particular, for the normal distribution in Fig. 1 we can observe a good agreement for both the methodologies; for the exponential and the mixture ones in Figs. 2 and 3 the result produced by BSHQI seems to better align with the original distribution while KDE seems to be less accurate as it always produces an under-estimate of the original distribution. The visual output is also supported by the results shown in Tables 1,2,3: if on the one hand, the reached accuracy seems similar in terms of AMISE and RMSE, on the other hand, the statistics conducted to test the validity of the null hypothesis clears out any doubt in assessing a better goodness of fit, under the statistical point of view, of the BSHQI method. In addition, examining the results in Table 4, it can be seen as an advantage in terms of computational time when utilizing the BSHQI, compared to the alternative approach. While the difference is not substantial, it is sufficient to highlight the efficiency of our approach as the time is almost halved when using BSHQI.

3. Copulas mixture model

In this section, we leverage the density estimation strategy introduced in Section 2 in the context of copulas. Our objective is to introduce the concept of copula and emphasize its profound connection to marginal distributions, with a specific focus on its application in formulating a novel clustering algorithm based on copula mixture. Copulas functions are a useful tool employed to easily express multivariate distributions by specifying the marginals. In this Section the principal concepts are revised following the setting in [21].

Definition 3.1. A D-dimensional copula is a CDF with uniform marginals:

$$C : [0,1]^D \to [0,1] \text{ such that } C(u) = C(u_1, \dots, u_D).$$

Theorem 3.2 (*Sklar's Theorem*). *Consider a D-dimensional CDF, G, with marginals $F_1, \dots, F_D$. Then there exists a copula, C, such that*

$$G(x_1, \dots, x_D) = C(F_1(x_1), \dots, F_D(x_D)) \tag{13}$$

for all $x_i \in [-\infty, \infty]$ and $i = 1, \dots, D$. If F_i is continuous for all $i = 1, \dots, D$ then C is unique. In the opposite direction, given a copula C and univariate CDFs, $F_1, \dots, F_D$, then G as in (13) is a multivariate CDF with marginals $F_1, \dots, F_D$.

Given a CDF G with PDF g, and a copula C defined as in Definition 3.1, the density copula function c can be computed as,

$$c(u_1, \dots, u_D) = \frac{g(F_1^{-1}(u_1), \dots, F_D^{-1}(u_D))}{f_1(F_1^{-1}(u_1)) \cdots f_D(F_D^{-1}(u_D))}, \tag{14}$$

where $f_1, \dots, f_D$ are the PDFs of the marginals.

A novel algorithm, which takes advantage of the proposed construction for empirical cumulative distribution based on BSHQI, for estimating the marginals of a chosen copula is presented. As main application, we show the performance of such an algorithm in the clustering context. Therefore, the following formulation will be framed within the clustering setting.

Our goal is to implement a model-based algorithm capable of correctly identifying how the instances of the dataset can be grouped into different clusters. In this sense, using copulas provides a way to fit data that have different probability distributions, thus having an advantage in better discriminating the possible clusters of a dataset. We can assume an a priori model made with K clusters and the data in each single cluster are distributed as a multidimensional Copula belonging to the Elliptical family, in particular Gaussian Copula, and Archimedean family, in which we consider the Clayton, Gumbel and Frank copulas [19]. We therefore stress the fact that our mixture distribution is composed of a linear combination of Copulas. This linear combination is called "*Copula Mixture Model*". In the following, we describe in detail the derived formulation.

Definition 3.3 (*Semiparametric Approach*)**.** A Copula Mixture is a function consisting of several Copula density functions c_k, with $k \in \{1,\dots,K\}$ and K denoting the number of clusters of the considered dataset. Each Copula c_k in the mixture is characterized by a vector $\boldsymbol{\omega}$ that defines the parameters of the specific copula chosen for the mixture, and by the methods chosen for the approximation of the marginals. Moreover, for each Copula density function c_k is defined a mixing probability π_k, referred to as mixing coefficient, such that:

$$\sum_{k=1}^{K} \pi_k = 1.$$

Let us assume a dataset $\mathbf{X} = (X_1, X_2, \dots, X_D)$ where each X_i consists of n i.i.d. observations, and where the ith observation is $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})^T$ for $i = 1 \dots, n$.

We express the probabilistic model of mixture copulas for all observations in the following form:

$$p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{x}_i) = \prod_{i=1}^{n}\sum_{k=1}^{K} \pi_k g_k(\mathbf{x}_i|\boldsymbol{\omega}_k) \tag{15}$$

where p is the probability, $\boldsymbol{\theta} := \{\pi_k, \boldsymbol{\omega}_k\}$ indicates the parameters of the model: π_k is the mixing probability and $\boldsymbol{\omega_k}$ represents the vector of parameters with respect to the chosen copula, while $g_k(\mathbf{x}_i|\boldsymbol{\omega_k})$ represents the multivariate distribution constructed through the copula density c_k, explicitly:

$$\begin{aligned} g_k(\mathbf{x}_i|\boldsymbol{\omega}_k) &= c_k(F_1(x_{i,1}), \dots, F_D(x_{i,D})|\boldsymbol{\omega}_k)\Big(f_1(x_{i,1}) \times \cdots \times f_D(x_{i,D})\Big) \\ &= c_k(F_1(x_{i,1}), \dots, F_D(x_{i,D})|\boldsymbol{\omega}_k)\prod_{j=1}^{D} f_j(x_{i,j}). \end{aligned}$$

Since there are different families of copulas, the parameter $\boldsymbol{\omega}_k$ is the one related to the specific copula that is chosen to model the cluster k. The goal of the mixture model is to find the optimal parameters in $\boldsymbol{\theta}$, in Eq. (15), that maximize the log-likelihood $\mathcal{L}(\mathbf{X}|\boldsymbol{\theta})$:

$$\arg\max_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) = \arg\max_{\pi_k, \boldsymbol{\omega}_k} \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k g_k(\mathbf{x}_i|\boldsymbol{\omega}_k). \tag{16}$$

Usually, in order to solve the optimization problem in (16), the Expectation–Maximization (EM) [24] algorithm is employed together with mixture models. To derive the probability that an observation $\mathbf{x}_i$, is drawn from g_k, we introduce the latent variable $\mathbf{z}_i = (z_{i1}, \dots, z_{iK})^T$ such that $z_{ik} \in \{0,1\}$ with $k \in \{1,\dots,K\}$. In this context, the introduction of latent variables is a common practice [50] that enhances both the theoretical framework and the rationale behind employing the expectation–maximization algorithm. We define the joint distribution $p(\mathbf{x}_i, \mathbf{z}_i)$ in terms of a marginal distribution $p(\mathbf{z}_i)$ and a conditional distribution $p(\mathbf{x}_i|\mathbf{z}_i)$,

$$p(\mathbf{x}_i, \mathbf{z}_i) := p(\mathbf{x}_i|\mathbf{z}_i)p(\mathbf{z}_i). \tag{17}$$

The marginal distribution for $\mathbf{z}_i$ is characterized by the mixing coefficients, with the specification that $p(z_{ik} = 1) = \pi_k$.

We know beforehand that each z_{ik} occurs independently and that it can only take the value one when k is equal to the cluster from which the observation comes, then the overall probability is:

$$p(\mathbf{z}_i) = p(z_{i1} = 1)^{z_{i1}} p(z_{i2} = 1)^{z_{i2}} \cdots p(z_{iK} = 1)^{z_{iK}} = \prod_{k=1}^{K} \pi_k^{z_{ik}},$$

while the conditional distribution $p(\mathbf{x}_i|\mathbf{z}_i)$ can be written as,

$$p(\mathbf{x}_i|\mathbf{z}_i) = \prod_{k=1}^{K} g_k(\mathbf{x}_i|\boldsymbol{\omega}_k)^{z_{ik}}. \tag{18}$$

Let us introduce $\mathbf{Z}$ as the matrix whose ith row is the vector of the latent variables $\mathbf{z}_i$, then we have the overall joint distribution,

$$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = \prod_{i=1}^{n}\prod_{k=1}^{K} \pi_k^{z_{i_k}} g_k(\mathbf{x}_i|\boldsymbol{\omega}_k)^{z_{i_k}}$$

and hence the log-likelihood is,

$$\log(p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{i_k}\left(\log\pi_k + \log g_k(\mathbf{x}_i|\boldsymbol{\omega}_k)\right). \tag{19}$$

The formulated expression for the joint distribution $p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})$ leads to significant simplifications in the EM algorithm. However this cannot be computed, since $\mathbf{z}_i$ is unknown, then we evaluate the conditional probability of $\mathbf{z}_i$ given $\mathbf{x}_i$, and its value can be determined using Bayes' theorem,

$$p(z_{ik}=1|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|z_{ik}=1)\,p(z_{ik}=1)}{p(\mathbf{x}_i)} = \frac{p(\mathbf{x}_i|z_{ik}=1)\,p(z_{ik}=1)}{\sum_{k'=1}^{K}\pi_{k'}\,p(\mathbf{x}_i|z_{ik'}=1)p(z_{ik'}=1)},$$

where the quantity $p(\mathbf{x}_i)$ has been computed via marginalization. Knowing that $p(z_{ik}=1)=\pi_k$ and $p(\mathbf{x}_i|z_{ik}=1)=g_k(\mathbf{x}_i|\boldsymbol{\omega}_k)$, then, the above equation becomes:

$$p(z_{ik}=1|\mathbf{x}_i) = \frac{\pi_k g_k(\mathbf{x}_i|\boldsymbol{\omega}_k)}{\sum_{k'=1}^{K}\pi_{k'}\,g_{k'}(\mathbf{x}_i|\boldsymbol{\omega}_{k'})}. \tag{20}$$

The quantity in (20) is called *responsibility of kth cluster to observation i* and from now on will be denoted with the symbol γ_{ik}. Then for every cluster, we have an array of responsabilities. This quantity is crucial in the Expectation step of the EM algorithm for the maximization of the complete log-likelihood.

4. Expectation–maximization for copula mixture model

We formalize the expectations maximization algorithm for the copula mixture model in the general form, this can be implemented in different ways:

- a completely parametric way in which the parameters of the copula and the parameters of the marginal probability densities are estimated. In this case, there are two approaches: Inference For Marginal (IFM) [19], and Expectation/Conditional Maximization ECM [28]. This last one, although it may work well for small-sized data, requires high computational costs when there is a lack of a-priori knowledge about the marginals. In such cases, one needs to search for the distribution that fits the data well within a set of distributions;
- a semiparametric way in which, the estimation of marginals is approached empirically. The semiparametric nature of this approach strikes a balance between flexibility and computational efficiency.

In this work, we adopt the semiparametric approach, leveraging the approximation properties of densities with the previously introduced BSHQI. Below, we describe the details of the proposed algorithm.

In the Expectation step, we employ the existing parameter values $\boldsymbol{\theta} := \boldsymbol{\theta}^{(t)}$ to determine the posterior distribution of the latent variables at the t-step of the algorithm, denoted as $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)})$. Subsequently, we utilize this posterior distribution to calculate the expectation of the complete-data log-likelihood, evaluated for a general parameter vector $\boldsymbol{\theta}$. This expectation, is called *auxiliary function* represented as $Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)})$, is given by

$$Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)})}\left(\log p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})\right).$$

To simplify the notation, we shall consider $\mathbb{E}_{(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)})}\left(\log p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})\right) = \mathbb{E}\left(\log p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})\right)$ and so we have:

$$\begin{aligned} Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)}) &= \mathbb{E}[\log p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta}^{(t)})] = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{(t)})\log p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(t)}\left(\log\pi_k + \log g_k(x_i|\boldsymbol{\omega}_k)\right) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(t)}\left(\log\pi_k + \log\Big(c_k(\hat{F}_1(x_{i,1}),\dots,\hat{F}_D(x_{i,D})|\boldsymbol{\omega}_k)\Big) + \sum_{j=1}^{D}\log\hat{f}_j(x_{ij})\right) \end{aligned} \tag{21}$$

in which $\gamma_{ik}^{(t)}$ is the *Responsibility* introduced in Eq. (20).

Maximization step:. In the maximization step we update the parameters in $\boldsymbol{\theta}^{(t+1)}$ by computing:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)}).$$

This is the most complex step of the algorithm and in the following, we address separately the computation of the optimal parameters $\pi_k^{(t+1)}$ and $\boldsymbol{\omega}_k^{(t+1)}$.

For $\pi_k^{(t+1)}$, a closed form can be derived. Note that the maximization of the function $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)})$ should take into account the restriction that $\sum_{k=1}^{K}\pi_k = 1$. Hence, we can add a Lagrange multiplier to (21),

$$Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(t)}\left(\log\pi_k + \log g_k(x_i|\boldsymbol{\omega}_k)\right) - \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right).$$

Taking the derivative of Q with respect to π_k and setting it equal to zero, leads to

$$\frac{\partial Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(t)})}{\partial \pi_k} = \sum_{i=1}^{n} \frac{\gamma_{ik}^{(t)}}{\pi_k} - \lambda = 0. \tag{22}$$

Then, by rearranging the terms and applying a summation over k to both sides of the equation, we obtain:

$$\sum_{i=1}^{n} \gamma_{ik}^{(t)} = \pi_k \lambda \rightarrow \sum_{k=1}^{K}\sum_{i=1}^{n} \gamma_{ik}^{(t)} = \sum_{k=1}^{K} \pi_k \lambda.$$

We know that the summation of all mixing coefficients equals one. In addition, we know that summing up the responsabilities γ over k will also give us 1. Thus we get $\lambda = n$. Using this result, we can solve (22) for π_k:

$$\pi_k^{(t+1)} := \frac{\sum_{i=1}^{n} \gamma_{ik}^{(t)}}{n}.$$

Regarding the optimization with respect to $\boldsymbol{\omega}_k$, systematic decomposition into distinct maximization steps is carried out in order to get the optimization of the complete data log-likelihood with respect to the parameters of the model.

One notable feature of our algorithm lies in its inherent flexibility in the selection of copulas from the outset. When a single copula is chosen initially, the subsequent maximization step, following the updating of marginals, exclusively targets the parameters of the selected copula until the log-likelihood function converges. Conversely, when the initial choice encompasses various copulas, a comprehensive fitting process is initiated after updating the marginals. The copula that best aligns with the updated data is then chosen based on the maximum likelihood, enhancing the adaptability and performance of our algorithm in diverse scenarios, formally:

- **Maximization first step:** For each cluster, use the data $\mathbf{X}_j$, $j \in \{1, 2, \dots, D\}$ to update the CDFs $\hat{F}_j$ according to Eq. (12) in which the weights are the *responsabilities* and compute the PDFs $\hat{f}_j$, $j \in \{1, 2, \dots, D\}$ with the BSHQI strategy described in Section 2.
- **Maximization second step (one copula):** By looking at (21), for each cluster, we need to maximize only the following,

$$\sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \left(\log\Big(c_k(\hat{F}_1(x_{i,1}), \dots, \hat{F}_D(x_{i,D}) | \boldsymbol{\omega}_k) \Big) \right), \tag{23}$$

  with respect to $\boldsymbol{\omega}_k$ for the optimal copula parameters.
- **Maximization second step (two or more copulas):** If the choice of different copulas is enabled, then, the optimization of (23) is carried out also for each copula.
  Note that, in this case for the maximization of the log-likelihood we use the Limited-memory Broyden, Fletcher, Goldfarb, Shanno (L-BFGS-B) method,[2] that is an optimization algorithm in the family of quasi-Newton methods that approximates the Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS) using a limited amount of computer memory [51,52].

To start the algorithm we use the following initialization procedure:

Initialization.

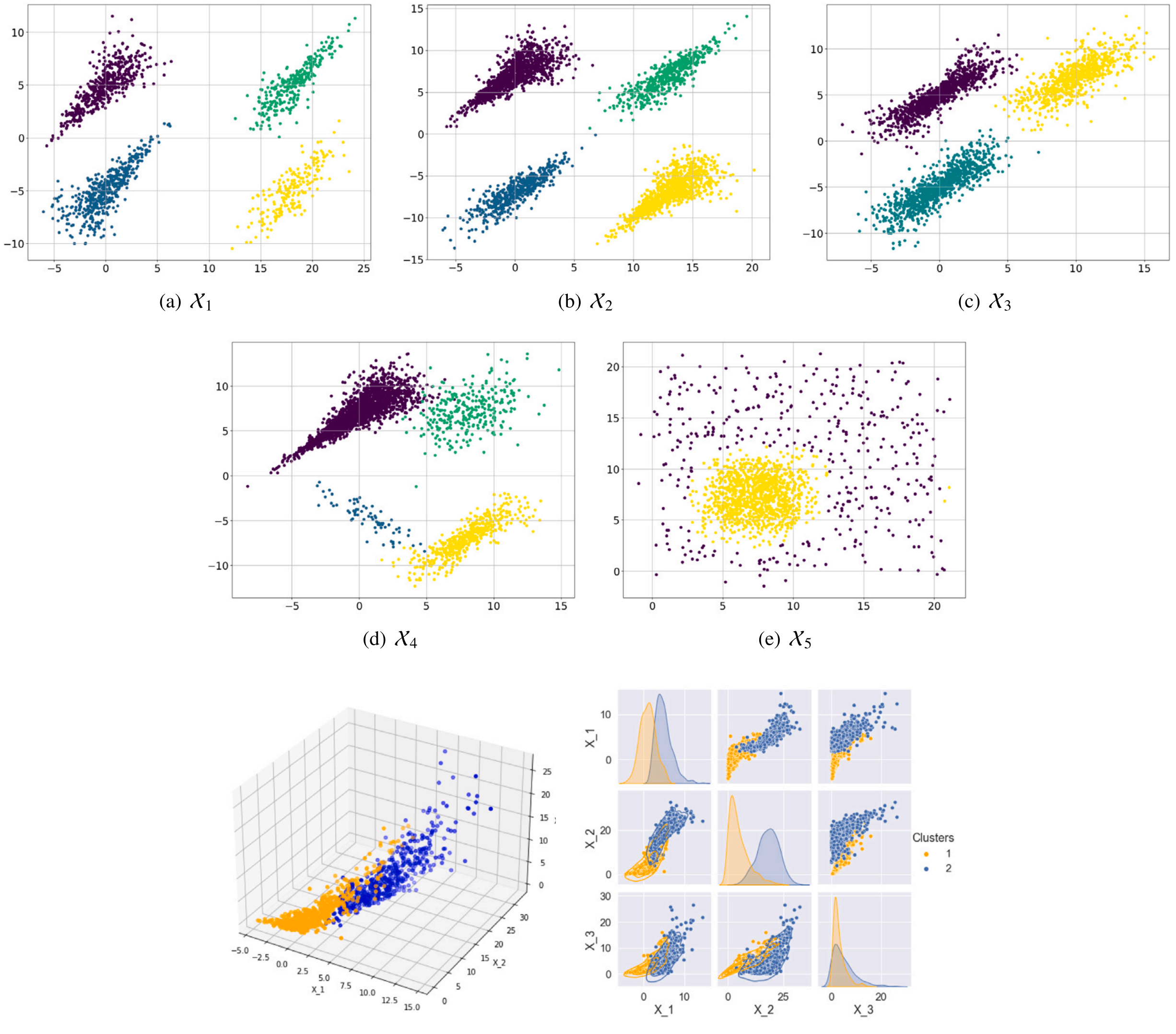
0. Choose a set S of copulas function among Gaussian, Clayton, Gumbel, Frank.
1. Create a random clustering partition of the given data. We call $X_{j,r}$ the subset of the jth column given by the produced partition.
2. For every cluster, compute the marginals $\hat{F}_j(X_{j,r})$ for $j = 1, \dots, D$, either with the KDE methodology or with the BSHQI (the choice is at the user's discretion).
3. For every cluster, find the best copula in S according to the maximum likelihood achieved and keep its parameters $\boldsymbol{\omega}_k$ fitted by the produced data $u_1 = \hat{F}_1(X_{1,r}), u_2 = \hat{F}_2(X_{2,r}), \dots, u_D = \hat{F}_D(X_{D,r})$, and compute the associated π_k for $k = 1, \dots, K$.
4. Repeat steps 1–3 for 5 times and select the copulas partition with the maximum likelihood so that $\theta^{(0)} := \{\pi_k^{(0)}, \boldsymbol{\omega}_k^{(0)}\} = \{\pi_k^{\text{best}}, \boldsymbol{\omega}_k^{\text{best}}\}$ for $k = 1, \dots, K$.

Expectation. In this step we evaluate the responsabilities.

Maximization. The algorithm as just described monotonically approaches a local minimum of the cost function. After computing the new estimates, we set $\theta^{(t)} = (\pi_k^{(t)}, \omega_k^{(t)})$ for $k = 1 \dots, K$, and go to the next Expectation step. We set the tolerance $tol \leq 10^{-4}$ then, the best parameters are obtained when the convergence of the log-likelihood is reached, i.e.

$$\frac{|\mathcal{L}(\mathbf{X}|\theta^{(t+1)}) - \mathcal{L}(\mathbf{X}|\theta^{(t)})|}{1 + |\mathcal{L}(\mathbf{X}|\theta^{(t+1)})|} < tol \quad \text{for} \quad t = 1, \dots, (iter - 1). \tag{24}$$

[2] https://docs.scipy.org/doc/scipy/reference/optimize.minimize-lbfgsb.html



(a) $\mathcal{X}_1$ (b) $\mathcal{X}_2$ (c) $\mathcal{X}_3$

(d) $\mathcal{X}_4$ (e) $\mathcal{X}_5$

(f) $\mathcal{X}_6$ (g) Pairwise $\mathcal{X}_6$

Fig. 4. Synthetic dataset: (a) Ground truth $\mathcal{X}_1$, (b) Ground truth $\mathcal{X}_2$, (c) Ground truth $\mathcal{X}_3$, (d) Ground truth $\mathcal{X}_4$, (e) Ground truth $\mathcal{X}_5$, (f) Ground truth $\mathcal{X}_6$, (g) Pairwise ground truth $\mathcal{X}_4$.

5. Experiments

In this section, we describe the experiments conducted to validate the accuracy of our algorithm. Specifically, we start by examining synthetic datasets constructed with diverse families of copulas. Subsequently, we delve into the assessment of real-world datasets. The clustering strategy using copulas and density estimation with the proposed BSHQI technique, called CopMixM_BSHQI, will be compared by using marginals estimation with kernel density estimation using the Python package KDEpy, referred to as CopMixM_KDEpy here, which is found to be the fastest among various Python packages for empirical density estimators as it utilizes the convolution Fast Fourier Transform for calculations. Additionally, we will compare the results obtained with clustering algorithms available from `scikit-learn`: Gaussian Mixture Model (GMM), where we set the initialization to '`random`', the tolerance equal to 10^{-4} (as in our algorithm) and default parameters setting otherwise, K-Means where we set the initialization to `random` and algorithm Density-Based Spatial Clustering of Applications with Noise (DBSCAN). While DBSCAN exhibits high performance when clusters are well-separated, it encounters a challenge in determining the optimal parameters for its operation. To address this issue, we utilize the DBSCAN implementation provided by `scikit-learn` and further perform a grid search to identify the optimal value for the parameters `eps` and `min_samples` searching, respectively in the range of $[0.1, 1]$ for `eps` and in the range $[2, 20]$ for `min_sample`. To measure the achieved performance we use some classical metrics, available from `scikit-learn`, such as:

Table 5
Clustering metrics for Synthetic Dataset $\mathcal{X}_1$. In bold the best values.

	K-Means	GMM	DBSCAN	CopMixM_BSHQI	CopMixM_KDEpy
Silhouette Score	0.660	0.512	0.64	**0.659**	0.628
Calinski–Harabasz Index	5977	1570	3430	**5890**	3940
Davies–Bouldin Score	0.482	0.529	1.36	**0.486**	0521
Adjusted Rand Score	0.974	0.437	0.968	**1**	0.963
Homogeneity Score	0.963	0.455	0.973	**1**	0.952
Rand Score	0.989	0.707	0.987	**1**	0.985
Completeness Score	0.964	0.797	0.986	**1**	0.957

Table 6
Results of CopMixM_BSHQI for the Synthetic Dataset $\mathcal{X}_1$ obtained with different copulas and the Gaussian copula.

Clusters	# Points	Different Copulas	# Points	One Copula
1	500	Gaussian	505	Gaussian
2	500	Gaussian	498	Gaussian
3	300	Gaussian	299	Gaussian
4	200	Clayton	198	Gaussian
Log-likelihood	−7535		−8125	

- Silhouette Score: a value between -1 and 1, with 1 being the best;
- Calinski–Harabasz Index: highest values indicate better performance.
- Davies–Bouldin Score: the minimum is 0, lower values indicate better output.

In those tests where ground truth labels are accessible, the evaluation is also done using the Adjusted Rand Score, Homogeneity Score, Rand Score, and Completeness Score which are permutation invariant and the highest value 1 indicates optimal clustering.

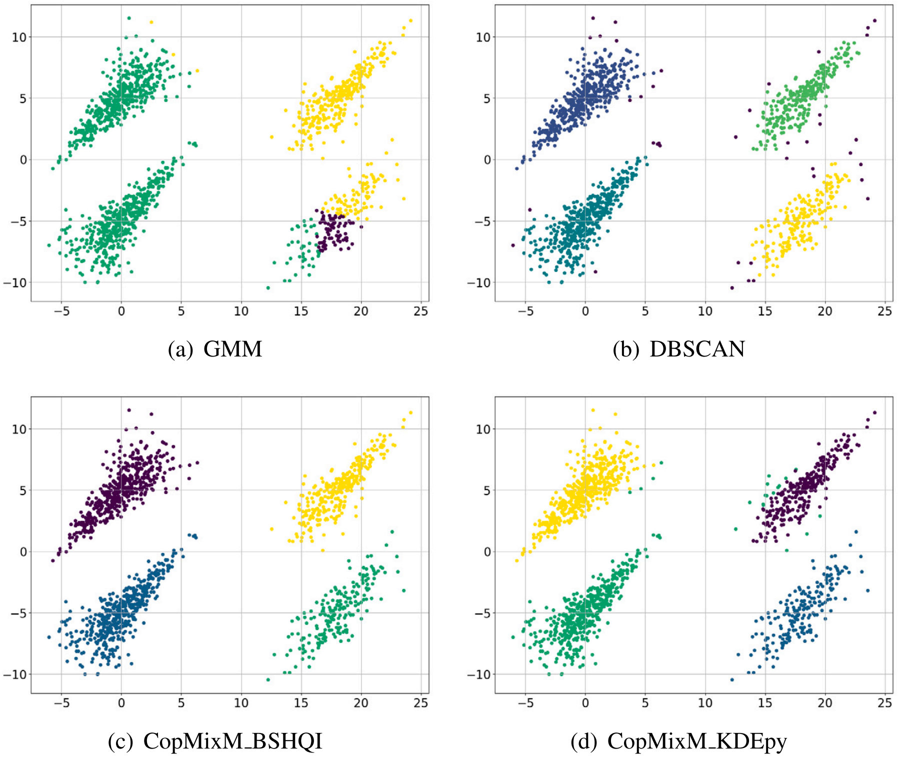
5.1. Synthetic dataset

We examine six synthetic datasets $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$, $\mathcal{X}_4$, $\mathcal{X}_5$, $\mathcal{X}_6$ characterized as follows:

- the samples in $\mathcal{X}_1$ are drawn from two Clayton copulas, one Frank and one Gumbel, see Fig. 4(a);
- the samples in $\mathcal{X}_2$ constitute 4 clusters and drawn from two Clayton copulas and two Gumbel copulas, see Fig. 4(b);
- the samples in $\mathcal{X}_3$ constitute 3 clusters drawn from the Frank copula, see Fig. 4(c);
- the samples in $\mathcal{X}_4$ constitute 4 clusters drawn from different copulas that are not well separated, see Fig. 4(d);
- the samples in $\mathcal{X}_5$ constitute 2 clusters with high density points and noisy points, see Fig. 4(e);
- the samples in $\mathcal{X}_6$ constitute 2 clusters of 3D scattered points, see Fig. 4(f)–(g).

All the presented results are obtained with random initialization and with the selection of different copulas. Regarding dataset $\mathcal{X}_1$, in Fig. 5, the outcomes from GMM, DBSCAN, CopMixM_BSHQI and clustering with copulas fitting the marginals using the KDEpy approach are shown. Visually, the best results are obtained with DBSCAN and CopMixM_BSHQI, see Fig. 5(b)–(c), as they closely align with the ground truth in Fig. 4(a). The metrics detailed in Table 5 show a more quantitative comparison for the five used clustering algorithms. Notably, CopMixM_BSHQI outperforms GMM and DBSCAN and gives better results compared to the other methods with respect to various metrics. The highest Silhouette Score, Calinski–Harabasz Index, and Homogeneity Score indicate a superior cluster quality and better separation. For the Davies–Bouldin Score, where a lower value is desirable in presence of well-defined and compact clusters, CopMixM_BSHQI again excels. The Adjusted Rand Score, Rand Score, and Completeness Score further support the dominance of CopMixM_BSHQI, as it achieves perfect score= 1.0 in these metrics, signifying strong agreement with the ground truth. In contrast, K-Means, GMM and CopMixM_KDEpy exhibit lower scores, reflecting a lower level of agreement with the true cluster assignments. Regarding DBSCAN, we anticipated favorable results since the synthetic example consists of well-separated clusters with high point density, despite the satisfactory performance of this algorithm, our approach outperforms it in terms of different metrics. Moreover, in Table 6, we present the results of CopMixM_BSHQI on the synthetic dataset $\mathcal{X}_1$ under various copula configurations. Two primary scenarios were considered: one utilizing diverse copulas for specific clusters, and the other employing a single Gaussian copula, which is commonly used in practice and is often considered the default choice in mixture models. The evaluation was based on the maximum likelihood for the choice of the cluster's copula and the Log-likelihood for the overall mixture model performance. The results suggest to adopt diverse copulas tailored to specific clusters rather than using a single one. Indeed, the higher log-likelihood highlights the superior model adaptability, to the considered synthetic dataset $\mathcal{X}_1$, compared to the simplistic use of a single Gaussian copula.

Moreover, with respect to datasets $\mathcal{X}_2$ and $\mathcal{X}_3$, the findings depicted in Figs. 6 and 7, along with the corresponding metrics presented in Tables 7 for $\mathcal{X}_2$ and in Tables 9 for $\mathcal{X}_3$, confirm the superiority of the CopMixM_BSHQI methodology over the other four methodologies. This validate that CopMixM_BSHQI works better not only in effectively distinguishing distinct clusters but also in appropriately assigning the suitable copula for each of them. This is further illustrated in Tables 8 and 10, which show again



(a) GMM

(b) DBSCAN

(c) CopMixM_BSHQI

(d) CopMixM_KDEpy

Fig. 5. Synthetic dataset $\mathcal{X}_1$: (a) GMM, (b) DBSCAN, (c) CopMixM_BSHQI, (d) CopMixM_KDEpy.

Table 7
Clustering metrics for Synthetic Dataset $\mathcal{X}_2$. In bold the best values.

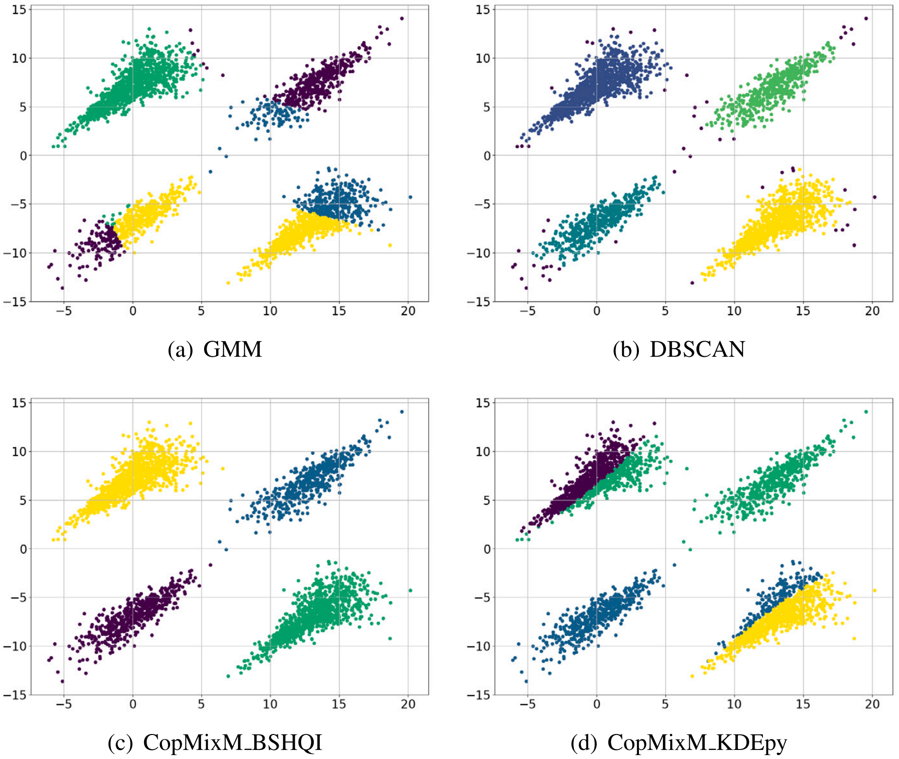
	K-Means	GMM	DBSCAN	CopMixM_BSHQI	CopMixM_KDEpy
Silhouette Score	**0.726**	0.269	0.711	**0.726**	0.370
Calinski–Harabasz Index	11 495	2640	6620	**11500**	2990
Davies–Bouldin Score	**0.365**	1.48	1.55	**0.365**	0.964
Adjusted Rand Score	0.998	0.382	0.979	**1**	0.605
Homogeneity Score	0.994	0.488	0.982	**0.999**	0.738
Rand Score	0.999	0.758	0.991	**1**	0.845
Completeness Score	0.994	0.512	0.938	**0.998**	0.720

Table 8
Results of CopMixM_BSHQI for the Synthetic Dataset $\mathcal{X}_2$ obtained with different copulas and the Gaussian copula.

Clusters	# Points	Different Copulas	# Points	One_Copula
1	1013	Clayton	1152	Gaussian
2	1014	Clayton	980	Gaussian
3	985	Gumbel	858	Gaussian
4	988	Gumbel	1010	Gaussian
Log-likelihood	**−13324**		−14873	

the better performance in the use of diverse copulas with respect to a single copula. In these instances as well, the log-likelihood supports this conclusion.

The two datasets $\mathcal{X}_4$ and $\mathcal{X}_5$ pose more critical challenges. Particularly, dataset $\mathcal{X}_4$ comprises four unbalanced clusters, while dataset $\mathcal{X}_5$ consists of additional noisy points into a high-density point distribution. These two datasets are introduced to assess the robustness to the presence of noise and the ability to handle unbalanced datasets. These experiments were introduced to further demonstrate the efficacy of our approach using copulas compared to DBSCAN, which could fail in presence of uneven distribution of data. Unlike traditional methods such as K-Means and DBSCAN, our approach enhances interpretability by uncovering the probability structure of the data. This capability holds significant implications for real-world applications where understanding the data's probability distribution is crucial. Additionally, it is worth noting that DBSCAN faces challenges in parameter tuning,



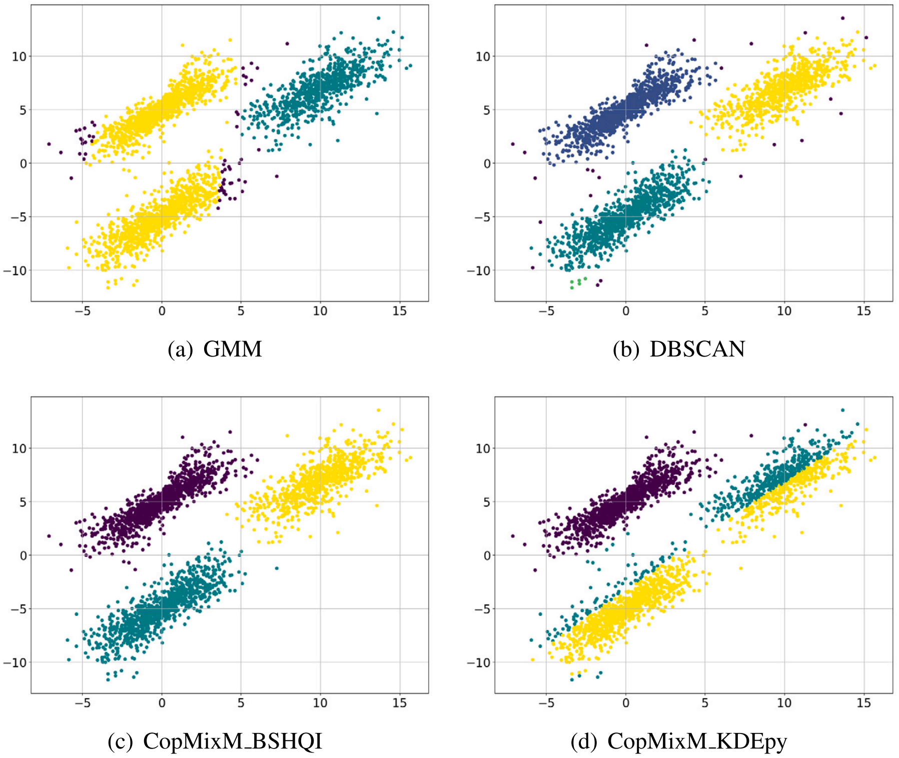
(a) GMM

(b) DBSCAN

(c) CopMixM_BSHQI

(d) CopMixM_KDEpy

Fig. 6. Synthetic dataset $\mathcal{X}_2$: (a) GMM, (b) DBSCAN, (c) CopMixM_BSHQI, (d) CopMixM_KDEpy.



(a) GMM

(b) DBSCAN

(c) CopMixM_BSHQI

(d) CopMixM_KDEpy

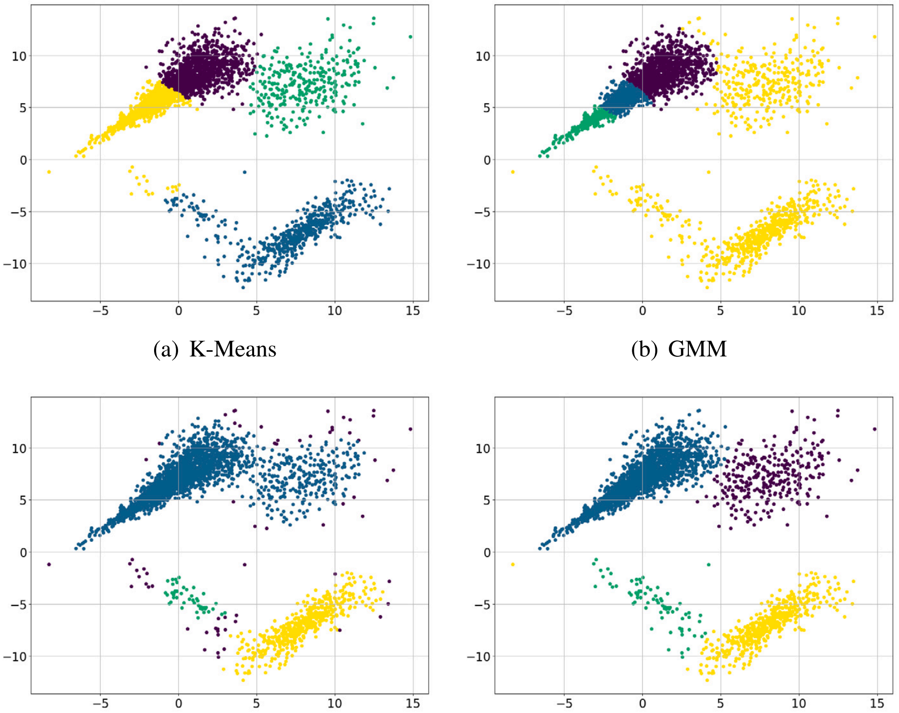
Fig. 7. Synthetic dataset $\mathcal{X}_3$: (a) GMM, (b) DBSCAN, (c) CopMixM_BSHQI, (d) CopMixM_KDEpy.

Table 9
Clustering metrics for Synthetic Dataset $\mathcal{X}_3$. In bold the best values.

	K-Means	DBSCAN	GMM	CopMixM_BSHQI	CopMixM_KDEpy
Silhouette Score	**0.638**	0.253	0.544	**0.638**	0.384
Calinski–Harabasz Index	**7874**	1420	3600	7710	702
Davies–Bouldin Score	**0.474**	3.8	1.86	0.476	1.47
Adjusted Rand Score	0.960	0.476	0.984	**0.999**	0.615
Homogeneity Score	0.932	0.518	0.991	**0.997**	0.598
Rand Score	0.982	0.724	0.993	**1**	0.804
Completeness Score	0.931	0.841	0.945	**0.997**	0.771

Table 10
Results of CopMixM_BSHQI for the Synthetic Dataset $\mathcal{X}_3$ obtained with different copulas and the Gaussian copula.

Clusters	# Points	Different Copulas	# Points	One Copula
1	702	Frank	698	Gaussian
2	1001	Frank	935	Gaussian
3	998	Frank	1067	Gaussian
Log-likelihood	**−12644**		−13835	



(a) K-Means

(b) GMM

(c) DBSCAN

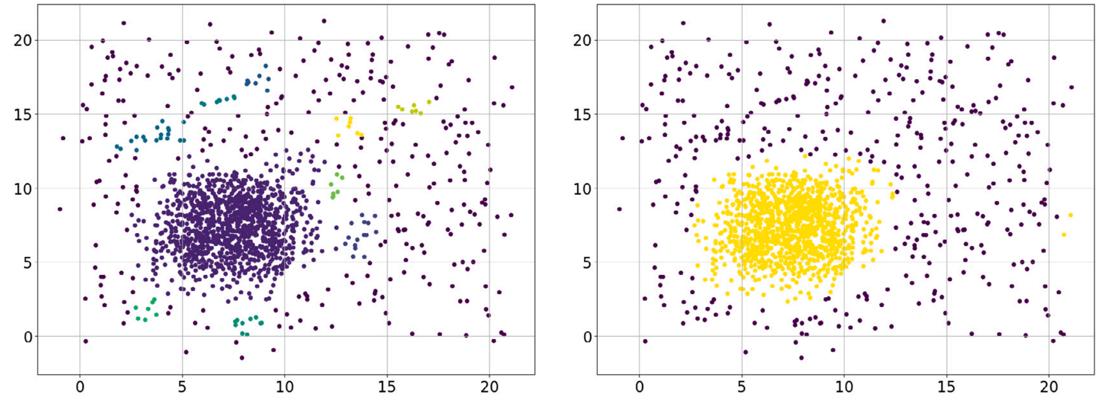
(d) CopMixM_BSHQI

Fig. 8. Synthetic dataset $\mathcal{X}_4$: (a) GMM, (b) K-Means, (c) DBSCAN, (d) CopMixM_BSHQI.

adding another layer of complexity to its application. We provide images and results for both experiments in Figs. 8 and 9, and in Tables 11 and 12. Particularly for dataset $\mathcal{X}_5$, we only measure the miss-classification rate instead of all other metrics, which could be misleading, especially in situations where the dataset is not well separated. In the latest experiment, we show the effectiveness of the algorithm using the synthetic dataset $\mathcal{X}_6$, which is in 3D and comprises of two clusters. The results are presented in Fig. 10 where only GMM, DBSCAN and CopMixM_BSHQI are shown. Moreover in Table 13 the achieved metrics are reported. It can be seen that the worst performance is given by GMM and DBSCAN, while the best sometimes is achieved by CopMixM_KDEpy, altought the discrepancy with CopMixM_BSHQI is minimal. For this test, since there are only two classes, we have also included accuracy in terms of miss-classification rate. As can be observed in this scenario the CopMixM_BSHQI performance surpasses that of the conventional K-Means, GMM and KDEpy based approach.

Table 11
Clustering metrics for Synthetic Dataset $\mathcal{X}_4$. In bold the best results.

	K-Means	GMM	DBSCAN	CopMixM_BSHQI
Silhouette Score	0.532	0.501	0.561	**0.574**
Calinski–Harabasz Score	**7680**	6670	2280	4880
Davies–Bouldin Score	0.592	0.664	1.64	**0.57**
Adjusted Rand Score	0.503	0.455	0.715	**0.974**
Homogeneity Score	0.856	0.801	0.631	**0.929**
Rand Score	0.746	0.721	0.86	**0.987**
Completeness Score	0.589	0.537	0.866	**0.931**



(a) DBSCAN (b) CopMixM_BSHQI

Fig. 9. Synthetic dataset $\mathcal{X}_5$: (a) DBSCAN, (b) CopMixM_BSHQI.

Table 12
Clustering metrics for Synthetic Dataset $\mathcal{X}_5$. In bold the best values.

	K-Means	GMM	DBSCAN	CopMixM_BSHQI
Miss-classification Rate:	0.32	0.41 (±0.02)	0.87	**0.06 (±0.01)**

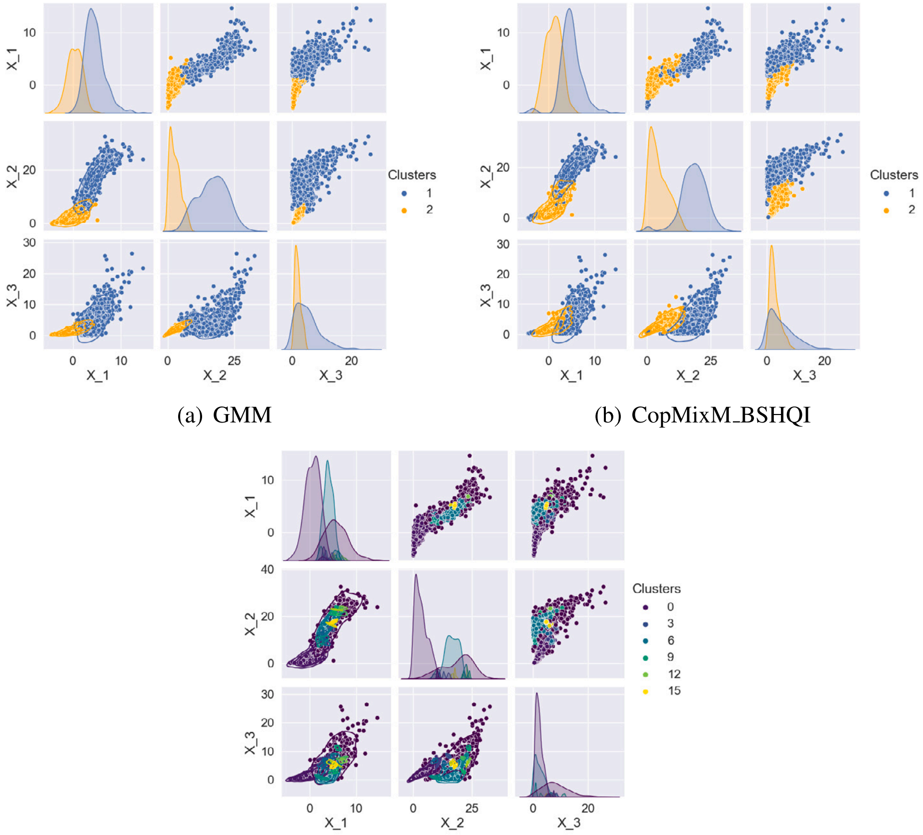
Table 13
Clustering metrics for Synthetic Dataset $\mathcal{X}_6$. In bold the best values.

	K-Means	GMM	DBSCAN	CopMixM_BSHQI	CopMixM_KDEpy
Silhouette Score	**0.576**	0.504	−0.0127	0.561	0.536
Calinski–Harabasz Score	**2030**	1420	164	1850	1630
Davies–Bouldin Score	0.601	**0.592**	1.6	0.618	0.661
Adjusted Rand Score	0.595	0.544	0.465	**0.695**	0.685
Homogeneity Score	0.488	0.528	0.601	0.589	**0.659**
Rand Score	0.797	0.772	0.632	**0.848**	0.843
Completeness Score	0.489	0.555	0.317	0.594	**0.595**
Miss-classification Rate:	0.12	0.11 (±0.02)	0.43	**0.08 (±0.01)**	0.09 (±0.01)

In summary, these results highlight the algorithm's effectiveness in capturing complex structures within synthetic datasets, positioning it as a robust choice for clustering activities in similar contexts. The findings suggest that the choice of copula significantly influences goodness of fit, with specific copula types being favored by certain clusters. Therefore, the model has an overall robust performance, particularly in contexts where the option to choose from multiple copulas is available, compared to relying on a single copula.

5.2. Real datasets

We conduct experiments to validate the effectiveness of the CopMixM_BSHQI algorithm on real-world datasets. Our analysis considers three datasets: the Australian Institute of Sport (AIS), Breast Cancer Wisconsin, and a case study involving the clustering



(a) GMM

(b) CopMixM_BSHQI

(c) DBSCAN

Fig. 10. Synthetic dataset $\mathcal{X}_6$: (a) GMM, (b) CopMixM_BSHQI, (c) DBSCAN.

of textual data using the 20newsgroups dataset.[3] For the AIS[4] and Breast Cancer Wisconsin datasets,[5] we compare our approach against a specific method referenced in [10], where our initialization settings are considered the same as in [10] wherever possible. In the case of the 20newsgroups dataset, we compare our results with the conventional GMM because as observed in synthetic datasets $\mathcal{X}_4$ and $\mathcal{X}_5$, DBSCAN is not an optimal choice in the presence of poorly separated clusters.

5.2.1. AIS

The AIS dataset, comprising 13 measurements collected from 102 male athletes and 100 female athletes. The primary objective of clustering this dataset is to assess the effectiveness of our methodologies in producing results aligned with the given ground-truth. We consider the same settings as in [10], focusing on a subset of five variables: lean body mass (LBM), weight (Wt), body mass index (BMI), white blood cell count (WBC), and percentage of body fat (PBF).

This particular example exhibits a non-Gaussian distribution, as the contours are non-elliptical and display asymmetric dependency patterns. Therefore, the application of copulas seems to be the optimal one for capturing this more complex form of dependency. Our algorithm was run with both K-Means and random initialization. Furthermore, we explored the use of different copula choices. We utilized the miss-classification rate metric to assess the performance of the clustering algorithm. The results obtained with our approach, employing both types of initialization, are presented in Table 14 alongside with those reported in [10].

The method called VCMM (K-Means) used in [10] for this dataset, initialized with K-Means reaches the best miss-classification value if in our model we set the number of bins equal to $\lceil n^{1/3} \rceil$, however the CopMixM_BSHQI with random initialization, shows a competitive result and in particular when the Rice Rule is adopted, it achieves the best results. This discrepancy in performance between the two initialization methods indicates that random initialization produces more favorable results for CopMixM_BSHQI in the context of the AIS dataset.

[3] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

[4] https://rdrr.io/cran/GLMsData/man/AIS.html

[5] https://archive.ics.uci.edu/dataset/14/breast+cancer

Table 14
Results for AIS dataset. The different initializations are reported in round brackets under the name of the algorithm. In bold the best values.

	K-Means	VCMM (init: K-Means)	CopMixM_BSHQI (init: K-Means)	CopMixM_BSHQI (init: Random)
Miss-classification rate (Bins=$\lceil n^{1/3} \rceil$)	0.21	**0.040**	0.090	0.045
Miss-classification rate (Bins=Rice)	0.21	0.040	0.040	**0.035**

Table 15
Results for Breast Cancer Dataset. The different initializations are reported in round brackets under the name of the algorithm. In bold the best values.

	K-Means	VCMM (init: K-Means)	VCMM (C-vine)	Multivariate normal	Multivariate skew normal	Multivariate t	Multivariate skew t	CopMixM_BSHQI (init: K-Means)	CopMixM_BSHQI (init: Random)
Miss-classification rate (Bins=$\lceil n^{1/3} \rceil$)	0.14	0.10	0.18	0.12	0.15	0.11	0.15	**0.09**	0.10
Miss-classification rate (Bins=Rice)	0.14	0.10	0.18	0.12	0.15	0.11	0.15	**0.08**	**0.08**

These results generally highlight the sensitivity of the mixture model algorithm to the choice of initialization method. While the algorithm CopMixM_BSHQI generally works well, it is critical to consider the impact of initialization on its effectiveness. The higher miss-classification rate observed with K-Means initialization should be taken into account when implementing CopMixM_BSHQI in scenarios similar to the AIS dataset and its further investigation will be the object of future work especially as it seems to improve by setting a specific number of bins. Indeed, by choosing the Rice rule, the CopMixM_BSHQI results are very competitive with respect to both initializations.

5.2.2. Breast Cancer Wisconsin (Diagnostic)

The Breast Cancer Wisconsin (Diagnostic) dataset from the UCI Machine Learning Repository [53] consists of digitized images of fine needle aspirates from breast masses from 569 patients. For each of the considered ten features, the mean value, extreme value (mean of the three largest values), and standard error are computed, resulting in 30 total features. The dataset comprises benign (352 patients) and malignant (212 patients) diagnoses, enabling the measurement of miss-classification rates for binary classification algorithms. In this study, the same features as in [10] were considered, specifically: perimeter standard error (PSE), extreme smoothness (ES), extreme concavity (EC), and extreme concave points (ECP).

The copMxM_BSHQI algorithm was applied with both random initialization and K-Means and by setting two different numbers for the bins. Our results are reported in Table 15, and a comparison is made with the outcomes presented in the study in [10]. The results indicate that CopMixM_BSHQI with K-Means and with the two choices of bins outperforms the other approaches, achieving a low miss-classification rate of 0.09 and 0.084, respectively. In this case, with the random initialization, we can observe a miss-classification rate of 0.082 only by choosing the Rice bins. This indicates a high degree of accuracy in grouping data points, demonstrating the capability of our methodologies to discern underlying patterns within the dataset. In contrast, the other methods from [10], such as VCMM, VCMM (C-vine), Multivariate normal, Multivariate skew normal, Multivariate t, and Multivariate skew t, exhibit comparatively higher miss-classification rates ranging from 0.10 to 0.18. The results show the effectiveness of the CopMixM_BSHQI approach when combined with either K-Means or random initialization.

5.2.3. Text clustering

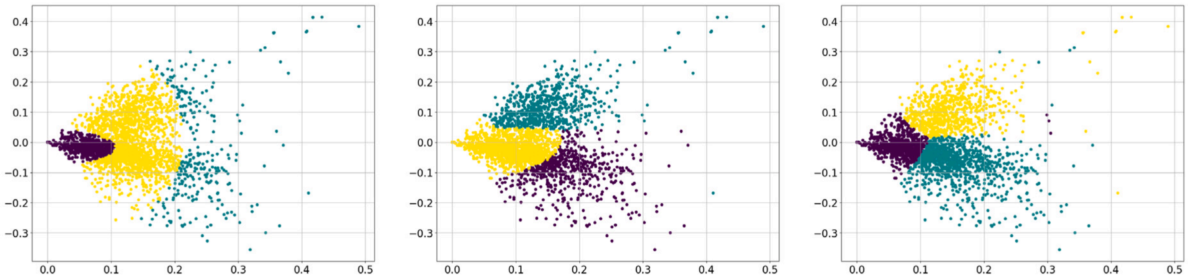
For the last experiment, we focus on the well-known 20newsgroups dataset, accessible through Scikit-Learn's API and designed for linguistic analysis. This dataset contains online discussions, or newsgroups, categorized into different thematic groups. For our clustering analysis, we specifically selected texts related to technology, religion, and sports.

To convert the textual data into a format suitable for quantitative analysis, we employ the `TfidfVectorizer` (implemented in `scikit-learn` library in Python[6]). This essential tool facilitates the transformation of text into numerical vectors by calculating Term Frequency-Inverse Document Frequency (TF-IDF) values for each term. TF-IDF reflects the importance of each term within individual documents relative to the entire dataset, allowing us to capture semantic information.

It is crucial to note that the `TfidfVectorizer` is also pivotal for the clustering process. We transform textual data into numerical representations, creating a dataset with 18,846 rows and 24,471 features. Considering the large size of the dataset, a preliminary step in our clustering analysis involves the application of a dimensionality reduction technique. Specifically, we employ Truncated Singular Value Decomposition (T-SVD)[7] with a predefined number of components set to 2. This preliminary step is fundamental to mitigate the challenges posed by high-dimensional data, allowing for a more efficient and manageable representation of the underlying structure. We present scatter plots for GMM, CopMixM_BSHQI and CopMixM_KDEpy in Fig. 11. Additionally, Table 16 provides a comparative overview of clustering performance between K-Means, GMM, CopMixM_BSHQI and CopMixM_KDEpy algorithms applied to the 20newsgroup dataset. Again the proposed procedure achieves the best results with respect to Silhouette score and Davies–Bouldin Score, while is the runner up for the Calinski–Harabasz score, but we observe a small discrepancy with the best result provided by K-Means.

[6] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html



(a) GMM (b) CopMixM_BSHQI (c) CopMixM_KDEpy

Fig. 11. Text clustering results.

Table 16
Clustering metrics for 20newsgroup Dataset. In bold the best values.

	K-Means	GMM	CopMixM_BSHQI	CopMixM_KDEpy
Silhouette Score	0.423	0.154	**0.425**	0.373
Calinski–Harabasz Score	**2326**	386	2200	1900
Davies–Bouldin Score	0.896	1.89	**0.849**	0.856

The highest Silhouette Score emphasizes a better ability to create well-defined clusters similarly, the highest Davis-Bouldin score assesses a better compactness and separation between clusters. The Calinski–Harabasz Score reflects the effectiveness in achieving a favorable ratio of between-cluster to within-cluster variance. In summary, across multiple clustering metrics, CopMixM_BSHQI outperforms GMM, with random initialization, as highlighted by the values in bold and provides a correct interpretation of the underlying statistical distribution.

6. Conclusions

In this paper we presented a novel algorithm for empirical density estimation and we used it for cluster modeling based on the use of Copulas. In particular, the multivariate copulas distribution rely on the estimation of the marginal distributions based on the Hermite quasi-interpolant in [6]. The proposed construction is superior in terms of statistical significance with respect to classical approaches based on empirical kernel density estimation and provides consistent cumulative distribution functions as outlined in the detailed analysis of Section 2. The novel clustering algorithm allows for the automatic selection of the best copula among a certain set of copulas families and provides a robust strategy with respect to a random seed as initialization. Moreover, the obtained results show a rather good agreement with the ground-truth (when provided), and mostly we are able to correctly identify the underlying statistical distribution from where the given points where drawn. The obtained clusters exhibit good shape parameters, in terms of Silhouette Score, Calinski–Harabasz Score and Davis-Bouldin Score, and achieve good accuracy in terms of permutation invariant metrics. Future work will be devoted to investigate the choice of the optimal bandwidth and further to deeply analyze how to deal with overlapping clusters, like in the text mining example.

Data availability

Data will be made available on request.

Acknowledgments

Cristiano Tamborrino and Francesca Mazzia acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU. The research of Antonella Falini is founded by PON Ricerca e Innovazione 2014–2020 FSE REACT-EU, Azione IV.4 “Dottorati e contratti di ricerca su tematiche dell’innovazione” CUP H95F21001230006. The authors thank the GNCS for its valuable support under the INdAM-GNCS project CUP E55F22000270001.

References

[1] Q. Li, J.S. Racine, Nonparametric econometrics: Theory and practice, Econ. Finance 23 (7) (1994) 2059–2078, http://dx.doi.org/10.1080/03610929408831371, URL https://press.princeton.edu/books/paperback/9780691248080/nonparametric-econometrics.

[2] M. Rosenblatt, Remarks on some nonparametric estimates of a density function, Ann. Math. Stat. 27 (3) (1956) 832–837, http://dx.doi.org/10.1214/aoms/1177728190.

[3] K.R. Gehringer, R.A. Redner, Nonparametric probability density estimation using normalized b–splines, Comm. Statist. Simulation Comput. 21 (3) (1992) 849–878, http://dx.doi.org/10.1080/03610919208813053.

[4] R.A. Redner, K. Gehringer, Function estimation using partitions of unity, Comm. Statist. Theory Methods 23 (7) (1994) 2059–2078, http://dx.doi.org/10.1080/03610929408831371.

[5] J. Kirkby, A. Leitao Rodriguez, D. Nguyen, Spline local basis methods for nonparametric density estimation, Stat. Surv. 17 (2023) http://dx.doi.org/10.1214/23-SS142.

[6] F. Mazzia, A. Sestini, The BS class of Hermite spline quasi-interpolants on nonuniform knot distributions, BIT Numer. Math. 49 (2009) 611–628, http://dx.doi.org/10.1007/s10543-009-0229-9.

[7] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666, http://dx.doi.org/10.1016/j.patrec.2009.09.011, URL https://www.sciencedirect.com/science/article/pii/S0167865509002323.

[8] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, J. Comput. Sci. 25 (2018) 456–466, http://dx.doi.org/10.1016/j.jocs.2017.07.018, URL https://www.sciencedirect.com/science/article/pii/S1877750316305002.

[9] Y. Zhou, H. Wu, Q. Luo, M. Abdel-Baset, Automatic data clustering using nature-inspired symbiotic organism search algorithm, Knowl.-Based Syst. 163 (2019) 546–557, http://dx.doi.org/10.1016/j.knosys.2018.09.013, URL https://www.sciencedirect.com/science/article/pii/S0950705118304647.

[10] Ö. Sahin, C. Czado, Vine copula mixture models and clustering for non-Gaussian data, Econometr. Stat. 22 (2022) 136–158, http://dx.doi.org/10.1016/j.ecosta.2021.08.011, URL https://www.sciencedirect.com/science/article/pii/S2452306221001052. The 2nd Special issue on Mixture Models.

[11] G. Celeux, G. Govaert, A classification EM algorithm for clustering and two stochastic versions, Comput. Statist. Data Anal. 14 (3) (1992) 315–332, http://dx.doi.org/10.1016/0167-9473(92)90042-E, URL https://www.sciencedirect.com/science/article/pii/016794739290042E.

[12] F.M.L. Di Lascio, F. Durante, R. Pappadà, Copula–based clustering methods, in: M. Úbeda Flores, E. de Amo Artero, F. Durante, J. Fernández Sánchez (Eds.), Copulas and Dependence Models with Applications, Springer International Publishing, Cham, 2017, pp. 49–67.

[13] I. Kosmidis, D. Karlis, Model-based clustering using copulas with applications, Stat. Comput. 26 (2014) http://dx.doi.org/10.1007/s11222-015-9590-5.

[14] S.R. Kasa, S. Bhattacharya, V. Rajan, Gaussian mixture copulas for high-dimensional clustering and dependency-based subtyping, Bioinformatics 36 (2) (2019) 621–628, http://dx.doi.org/10.1093/bioinformatics/btz599, arXiv:https://academic.oup.com/bioinformatics/article-pdf/36/2/621/31962871/btz599.pdf.

[15] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678, http://dx.doi.org/10.1109/TNN.2005.845141.

[16] A. Ezugwu, A. Ikotun, O. Olaide, L. Abualigah, O. Agushaka, C. Eke, A. Akinyelu, A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, Eng. Appl. Artif. Intell. 110 (2022) 104743, http://dx.doi.org/10.1016/j.engappai.2022.104743.

[17] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A.Y. Zomaya, S. Foufou, A. Bouras, A survey of clustering algorithms for big data: Taxonomy and empirical analysis, IEEE Trans. Emerg. Top. Comput. 2 (3) (2014) 267–279, http://dx.doi.org/10.1109/TETC.2014.2330519.

[18] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O.P. Patel, A. Tiwari, M.J. Er, W. Ding, C.-T. Lin, A review of clustering techniques and developments, Neurocomputing 267 (2017) 664–681, http://dx.doi.org/10.1016/j.neucom.2017.06.053, URL https://www.sciencedirect.com/science/article/pii/S0925231217311815.

[19] H. Joe, J.J. Xu, The Estimation Method of Inference Functions for Margins for Multivariate Models, Faculty Research and Publications, 1996, http://dx.doi.org/10.14288/1.0225985, URL https://open.library.ubc.ca/collections/facultyresearchandpublications/52383/items/1.0225985.

[20] R.B. Nelsen, An Introduction to Copulas (Springer Series in Statistics), Springer-Verlag, Berlin, Heidelberg, 2006.

[21] F. Durante, C. Sempi, Principles of Copula Theory, CRC Press, 2015.

[22] A. Tewari, M. Giering, A. Raghunathan, Parametric characterization of multimodal distributions with non-Gaussian modes, ICDM, in: Proceedings - IEEE International Conference on Data Mining, vol. 11, 2011, pp. 286–292, http://dx.doi.org/10.1109/ICDMW.2011.135.

[23] V. Rajan, S. Bhattacharya, Dependency clustering of mixed data with Gaussian mixture copulas, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI '16, AAAI Press, 2016, pp. 1967–1973.

[24] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B Stat. Methodol. 39 (1) (1977) 1–38.

[25] G. Mclachlan, T. Krishnan, The EM Algorithm and Extensions (Wiley Series in Probability and Statistics), 2007, http://dx.doi.org/10.1002/9780470191613.

[26] D.C. Gilles. Celeux, J. Diebolt, Stochastic versions of the em algorithm: An experimental study in the mixture case, J. Stat. Comput. Simul. 55 (4) (1996) 287–314, http://dx.doi.org/10.1080/00949659608811772.

[27] I. Marschner, On stochastic versions of the EM algorithm, Biometrika 88 (2001) http://dx.doi.org/10.1093/biomet/88.1.281.

[28] X.-L. Meng, D.B. Rubin, Maximum likelihood estimation via the ECM algorithm: A general framework, Biometrika 80 (2) (1993) 267–278, http://dx.doi.org/10.1093/biomet/80.2.267.

[29] M.A. Tanner, W.H. Wong, The calculation of posterior distributions by data augmentation, J. Amer. Statist. Assoc. 82 (398) (1987) 528–540, http://dx.doi.org/10.1080/01621459.1987.10478458, URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1987.10478458.

[30] G.C.G. Wei, M.A. Tanner, A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms, J. Amer. Statist. Assoc. 85 (411) (1990) 699–704, URL http://www.jstor.org/stable/2290005.

[31] R.A. Redner, H.F. Walker, Mixture densities, maximum likelihood and the Em algorithm, SIAM Rev. 26 (2) (1984) 195–239, URL http://www.jstor.org/stable/2030064.

[32] G.J. McLachlan, D. Peel, Finite mixture models, in: Probability and Statistics – Applied Probability and Statistics Section, vol. 299, Wiley, New York, 2000, URL https://www.bibsonomy.org/bibtex/2eb61da66e6f9b3fb503758fc9661122d/neongod.

[33] C. Genest, K. Ghoudi, L.-P. Rivest, A semiparametric estimation procedure of dependence parameters in multivariate families of distributions, Biometrika 82 (1995) http://dx.doi.org/10.1093/biomet/82.3.543.

[34] H. Joe, Dependence Modeling with Copulas, CRC Press, 2014.

[35] A. Falini, F. Mazzia, C. Tamborrino, Spline based Hermite quasi-interpolation for univariate time series, Discrete Contin. Dyn. Syst. - S 15 (12) (2022) 3667–3688, http://dx.doi.org/10.3934/dcdss.2022039, URL https://www.aimsciences.org/article/id/621c8f822d80b7479e4357ab.

[36] E. Bertolazzi, A. Falini, F. Mazzia, The object oriented C++ library QIBSH++ for Hermite spline quasi interpolation, 2022, arXiv preprint arXiv:2208.03260.

[37] E.A. Nadaraya, Some new estimates for distribution functions, Theory Probab. Appl. 9 (3) (1964) 497–500, http://dx.doi.org/10.1137/1109069.

[38] H. Yamato, et al., Uniform convergence of an estimator of a distribution function, Bull. Math. Stat. 15 (1973) http://dx.doi.org/10.5109/13073.

[39] A. Azzalini, A note on the estimation of a distribution function and quantiles by a kernel method, Biometrika 68 (1981) http://dx.doi.org/10.1093/biomet/68.1.326.

[40] C. Dagnino, P. Lamberti, S. Remogna, On spline quasi-interpolation through dimensions, Ann dell'Univ di Ferrara 68 (2) (2022) 397–415.

[41] T. Lyche, L.L. Schumaker, Local spline approximation methods, J. Approx. Theory 15 (4) (1975) 294–325.

[42] C. De Boor, G. Fix, Spline approximation by quasiinterpolants, J. Approx. Theory 8 (1) (1973) 19–45.

[43] B.-G. Lee, T. Lyche, K. Mørken, Some examples of quasi-interpolants constructed from local spline projectors, in: Mathematical Methods for Curves and Surfaces, vol. 2000, Oslo, 2000, pp. 243–252.

[44] P. Sablonniere, Recent progress on univariate and multivariate polynomial and spline quasi-interpolants, Trends Appl. Construct. Approxim. (2005) 229–245.

[45] R. Redner, Convergence rates for uniform B-spline density estimators part I: One dimension, SIAM J. Sci. Comput. 20 (1999) http://dx.doi.org/10.1137/S1064827595291996.

[46] M.P. Wand, M.C. Jones, Kernel smoothing, in: Chapman & Hall/CRC Monographs on Statistics & Applied Probability, (no. 60) Chapman & Hall, Boca Raton, FL, U.S., 1994, URL https://oro.open.ac.uk/28198/.

[47] F.J. Massey, The Kolmogorov-Smirnov test for goodness of fit, J. Amer. Statist. Assoc. 46 (253) (1951) 68–78, URL http://www.jstor.org/stable/2280095.

[48] T.W. Anderson, On the distribution of the two-sample cramer-von mises criterion, Ann. Math. Stat. 33 (3) (1962) 1148–1159, http://dx.doi.org/10.1214/aoms/1177704477.

[49] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Routledge, 2018.

[50] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, Berlin, Heidelberg, 2006.

[51] G. Andrew, J. Gao, Scalable training of regularized log-linear models, in: Proceedings of the 24th International Conference on Machine Learning, Association for Computing Machinery, New York, NY, USA, 2007, pp. 33–40, http://dx.doi.org/10.1145/1273496.1273501.

[52] R. Malouf, A comparison of algorithms for maximum entropy parameter estimation, in: Proceedings of the 6th Conference on Natural Language Learning, in: COLING-02, Vol. 20, Association for Computational Linguistics, USA, 2002, pp. 1–7, http://dx.doi.org/10.3115/1118853.1118871.

[53] W. Wolberg, W. Street, O. Mangasarian, Breast Cancer Wisconsin (Diagnostic), UCI Machine Learning Repository, 1995, http://dx.doi.org/10.24432/C5DW2B.