# On the Estimation of Gaussian Mixture Copula Models

**Ashutosh Tewari** [*1]

## Abstract

This paper revisits Gaussian Mixture Copula Model (GMCM), a more expressive alternative to the widely used Gaussian Mixture Model (GMM), with the goal to make its parameter estimation tractable. Both the Expectation Maximization and the direct Likelihood Maximization frameworks for GMCM have to grapple with a likelihood function that lacks a closed-form. This has led to a few approximation schemes that alleviate the problem, nonetheless leaving the issue still unresolved. Additionally, past works have alluded to an additional challenge of parameter non-identifiability, but none has offered a rigorous treatment and a commensurate solution framework to overcome the same. This work offers solutions to each of these issues in an attempt to help GMCM realize its full potential. The source of non-identifiability is not only proven but also suitable priors are proposed that eliminate the problem. Additionally, an efficient numerical framework is proposed to evaluate the intractable likelihood function, while also providing its analytical derivatives. Finally, a view of GMCM as a series of bijective mappings from a base distribution is presented, which paves the way to synthesize GMCM using modern, probabilistic programming languages (PPLs). The main claims of this work are supported by empirical evidence gathered on synthetic and real-world datasets.

## 1. Introduction

Modeling multivariate data is of fundamental interest, in several domains, to solve myriad of practical problems. From a probabilistic viewpoint, it amounts to defining a generative process that best explains the observed data when seen as random variables. Copulas provide a unique framework to model multivariate data that allows for complete control on the marginal behaviors of the random variables while being able to separately capture the dependencies between them. See (Durante & Sempi, 2010) and references therein for a survey on this subject. This decoupling induced by a copula can be especially significant when the true data-generating process imposes strict constraints over the marginal distributions of some or all random variables. Ideally, any effort to model such data should adhere to these constraints. However, in the pursuit of finding a joint model of the random variables, one typically ends up with inconsistent marginal models. Given the ability of copulas to overcome such inconsistencies, they have been applied in many scientific fields though particularly in finance (Genest et al., 2009; Cherubini et al., 2004), reliability analysis (Rychlik, 2010) and molecular biology (Bilgrau et al., 2012; Li et al., 2011; Kim et al., 2008; Ma & Wang, 2012). There have also been attempts to find synergies between copula theory and machine learning to build high-fidelity data-driven models (see Elidan, 2013, for a survey on the applications of copulas in machine learning approaches).

The focus of this paper is on the multivariate modeling of continuous random variables, exhibiting multimodal behavior in their joint (and/or marginal) distribution. Gaussian Mixture Models (GMMs) (Bilmes, 1997) have been prolifically used to model such datasets, thanks to their simplicity and an efficient Expectation-Maximization (EM) algorithm for parameter estimation. However, the assumption of jointly normally distributed components is frequently violated in real-world applications, with unintended practical ramifications. The Gaussian Mixture Copula Model (GMCM) (Tewari et al., 2011; Bilgrau et al., 2016; Bhattacharya & Rajan, 2014; Kasa & Rajan, 2018) offers a more expressive alternative to GMM while keeping the same parameterization to capture multimodal dependence structure. Figure 1 illustrates the expressivity endowed by the GMCM via a synthetic two-dimensional dataset with 100 samples (Figure 1a) that appears to have a bimodal distribution with non-Gaussian modes. The best-fit GMCM and a GMM are obtained on this dataset, wherein the optimal number of components is determined via the widely used Bayesian Information Criterion (BIC). A quick look at the density contours (Figure 1b-1c) and the generated random samples (Figure 1d-1e), suggests that the GMCM is a more faithful model of the underlying data, with a noticeable tighter fit.

[1]Amazon Mechatronics, Seattle, USA. Correspondence to: Ashutosh Tewari <ashutosh80@gmail.com>.

*Proceedings of the $40^{th}$ International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

The GMM, on the other hand, can be seen to diffuse into regions with no data support, even with an extra mixing component.

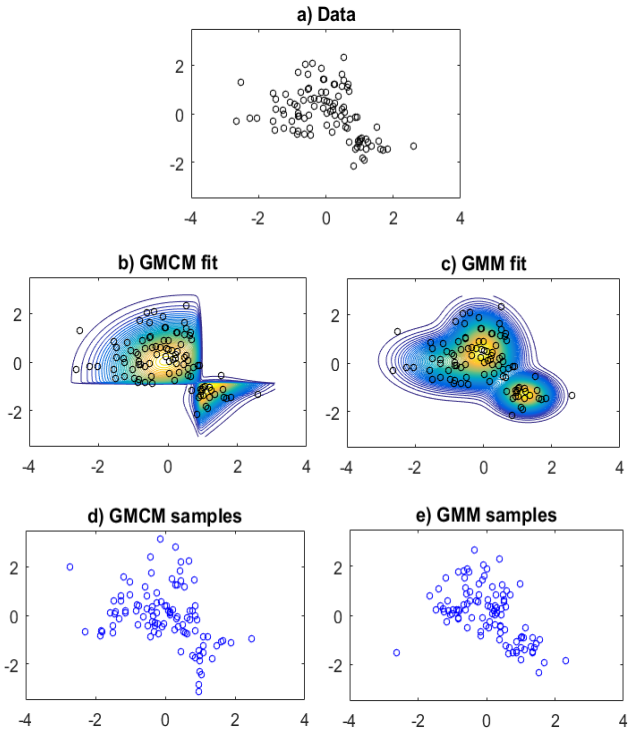


*Figure 1.* (a) A 2-dimensional dataset with 100 samples, (b-c) contours of best-fit GMCM (with 2 components) and GMM (with 3 components), (d-e) 100 random samples generated from the two fitted distributions. A tighter fit and a closer resemblance of the random samples to the training dataset suggest the GMCM to be a superior generative model of the data than the GMM.

Despite its superior expressivity, the estimation of GMCM parameters remains a challenge primarily due to three reasons that we briefly mention here and later explained. First, GMCMs suffer from an inherent issue of *parameter non-identifiability*. Second, its likelihood function does not admit a closed analytical form, and thus not amenable to the EM framework for parameter estimation. Third, concomitant to the second reason, even direct likelihood maximization (via gradient-based methods) becomes hard due to the lack of analytical gradients (numerical gradients are computationally expensive). The main contributions of this paper address each of these issues (not in the same order) i.e. 1) additional conditions are specified that provably mitigate the non-identifiability of GMCM, 2) a correct formulation of EM algorithm for GMCM is presented, 3) a numerical scheme is proposed to approximate GMCM's likelihood function while providing analytical gradient for the same, and 4) a view of GMCM as a series of bijective mappings is presented that makes it amenable to modern probabilistic programming frameworks and leverage their built-in automatic differentiation capabilities.

The plan of this paper is as follows. Sections 1.1 presents a short literature review on multivariate copula construction. Section 2 describes the GMCM framework and highlights the challenges with the estimation of its parameters (Section 2.1). A view of GMCM as a series of bijective mappings is presented in Section 3, wherein a numerical scheme is also proposed to obtain an intractable mapping (Section 3.1). The source of non-identifiability of GMCM is discussed in section 4 and a solution is proposed. A correct derivation of EM algorithm for GMCM is presented in Section 5. Results on synthetic datasets are included in section 6 to corroborate the claims, before concluding in section 7 with remarks on a few future research directions.

### 1.1. Multivariate Copula Construction

The literature on copulas has been dominated by bivariate copula models with a rich set of parametric families to choose from (Nelsen, 1999b). Although the idea of higher dimensional copula construction is not new (see Genest. et al., 1995; Joe, 1993; Kojadinovic & Yan, 2010), the literature on it is relatively recent. For instance, Bedford & Cooke (2002); Kurowicka & Joe (2010); Czado (2010) proposed synthesis of multivariate copula from bivariate copulas by assuming a tree-structured dependency between the random variables. This idea was extended to directed acyclic graphs (see Elidan, 2010; Hanea et al., 2006), giving rise to *Copula Bayesian Networks*. In high-dimensional settings, the recovery of a sparse inverse covariance structure of a Gaussian copula was studied by Liu et al. (2009) yielding *non-paranormal* models.

Copula-based construction to capture multi-modality — a frequently observed trait in real-world data—was first addressed by Tewari et al. (2011) with the proposal of GMCM. Bilgrau et al. (2016) furthered this work by noting certain challenges with the parameter estimation of GMCM and proposed practical solutions for the same. This was coupled with an improved implementation of the model as an open-source package (Bilgrau et al., 2017) in R. Rajan & Bhattacharya (2016) extended GMCM to construct flexible generative models for mixed (continuous and discrete) datatypes. Kasa et al. (2019) addressed the scalability of GMCM estimation in high-dimensional settings that are common in gene-expression datasets. The role of automatic differentiation, to obtain gradients of GMCM's intractable likelihood function, was explored by Kasa & Rajan (2022). Nevertheless, these works only partly addressed the fundamental issues of GMCM pertaining to parameter non-identifiability and the intractability of its likelihood function (and its gradient), thereby motivating this work.

Recently, there have been some interesting developments in the area of *model-based clustering* using copulas (Kosmidis

& Karlis, 2016; Mazo, 2017; Marbac et al., 2017; Rey & Roth, 2012). The motivation there is to overcome the restrictive normality assumption by cleverly using copulas, from *known* parametric families, to capture the dependence in each mixing component. For instance, the *Gaussian Copula Mixture Model* (GCMM) (Marbac et al., 2017) employs a Gaussian copula for modeling the component-wise dependencies. Although with very similar motivations (and names), the two lines of work (GMCM and GCMM) are fundamentally different, as the goal in GMCM is to seek a *single* copula distribution to capture the entire multimodal dependence structure.

Another related (albeit rather remotely) line of work pertains to deep generative models a.k.a *normalizing flows* (NFs), which has garnered significant attention in the Machine Learning community (see Kobyzev et al., 2021, for a comprehensive review on NFs). The core idea behind NFs is to transform a simple base distribution, such as an isotropic Gaussian, via bijective mappings that are carefully crafted using deep neural networks. Endowed with such mappings, one can compose highly expressive generative models for continuous data, resulting in best-in-class performances for the task of multivariate density estimation. The similarities and dissimilarities are drawn between GMCM and the NF-based models.

### 1.2. Notation

Lowercase letters are used for scalars, lowercase boldface letters for vectors, uppercase letters for matrices, and Greek letters for model parameters or functions. Unless otherwise stated, vectors are column vectors. Subscripts are used to denote an element of a vector or a matrix. For example, $\boldsymbol{x}_i$ and $X_{ij}$ denote the $i^{th}$ and the $(i,j)^{th}$ elements of a vector $\boldsymbol{x}$ and a matrix $X$, respectively. Likewise, $X_{i:}$ (or $X_{:i}$) represents the $i^{th}$ row (or column) of a matrix $X$. Subscripts are also used to indicate dimension-specific functions. For instance, the marginal distribution, induced by a joint distribution $\Psi$, along the $j^{th}$ dimension is denoted as $\Psi_j$. Superscripts are reserved to indicate parameter association. For example, $\Theta^i$ denotes parameters associated with some entity $i$. Table 2 in appendix A lists frequently appearing symbols in the paper for quick reference.

## 2. Gaussian Mixture Copula Model

**Definition 2.1.** A $m$-component *Gaussian Mixture Copula* (GMC) distribution, parameterized by $\Theta = \{\boldsymbol{\mu}^l, \Sigma^l, \alpha^l\}_{l=1}^m$, defines a joint distribution of a vector $\boldsymbol{u}$, whose constituent elements are uniformly distributed, i.e. $\boldsymbol{u}_j \sim$ Uniform(0,1), $j \in \{1, 2, \cdots d\}$. The GMC density function is given by Equation (1).

$$\zeta(\boldsymbol{u};\Theta) = \left( \frac{\psi\left(\Psi^{-1}(\boldsymbol{u});\Theta\right)}{\prod_{r=1}^{d} \psi_r\left(\Psi_r^{-1}\left(\boldsymbol{u}_r\right);\Theta^r\right)} \right) \tag{1}$$

The symbol $\psi(\cdot;\Theta)$ denotes the joint density function of a GMM parameterized with $\Theta = \{\boldsymbol{\mu}^l, \Sigma^l, \alpha^l\}_{l=1}^m$, where $\boldsymbol{\mu}^l \in \mathbb{R}^d$, $\Sigma^l \in \mathbb{S}_+^d$ and $\alpha^l \in \mathbb{R}_+$s.t. $\sum \alpha^l = 1$ denote the mean vector, the covariance matrix and the mixing proportion of the $l^{th}$ component, respectively. The marginal densities induced by the GMM are denoted by $\psi_r(\cdot;\Theta^r)$, with $\Theta^r \subset \Theta$ being the subset of parameters corresponding to the $r^{th}$ dimension. Also, $\Psi_r(\cdot)$ $\left(\text{and } \Psi_r^{-1}(\cdot)\right)$ is the cumulative distribution function (and its inverse) of the GMM along the $r^{th}$ margin, and $\Psi^{-1}(\boldsymbol{u}) = [\Psi_1^{-1}(\boldsymbol{u}_1), \cdots, \Psi_d^{-1}(\boldsymbol{u}_d)]$. This definition directly follows from the *inversion method* of constructing copulas from any multivariate distribution (in this case, a Gaussian Mixture distribution) with continuous margins (see Nelsen, 1999a, chapter 3). Since all the elements of a sample $\boldsymbol{u} \in [0,1]^d$ from GMC distribution are uniformly distributed, one can transform those via arbitrary univariate quantile functions $F_j^{-1}(u_j;\lambda_j)$ $(j \in \{1, 2, \cdots d\})$, with their respective dimension specific parameters $\lambda_j$s. This feature allows one to model the marginal and the joint behavior of a multivariate dataset independently (a hallmark of any copula-based model construction).

### 2.1. MLE challenges in GMCM

Maximum Likelihood Estimation (MLE) in GMCM amounts to estimating both the copula parameters ($\Theta$ in Definition 2.1) and the marginal parameters ($\lambda_j s$) in conjunction. Nevertheless, a computationally efficient and consistent estimator proposed by Joe & Xu (1996), where the marginal distributions are learned first followed by the estimation of copula parameters, is quite pervasive in practice. Along the same lines, this paper also assumes the marginal distributions to be arbitrary but known and tackles the much harder problem of estimating the GMC parameters by maximizing the log-likelihood function $\ell_\zeta(\Theta|U)$ given by Equation (2),

$$\ell_\zeta(\Theta|U) = \sum_{i=1}^{n} \log\left[\zeta(U_{:i};\Theta)\right], \tag{2}$$

where the function $\zeta(\cdot)$ is the GMC density function given by Equation (1). Assuming that from a training dataset $X \in \mathbb{R}^{d\times n}$ the marginal distributions have been learned, the matrix $U \in [0,1]^{d\times n}$ can then formed after transforming the dataset $X$ via the learned marginal distribution functions i.e., $U_{j:} = F_j(X_{j:};\lambda_j), j \in \{1, 2, \cdots, d\}$. Being a continuous and smooth function, $\ell_\zeta(\Theta|U)$ can be maximized using any gradient-based algorithm, however, the task is computationally expensive. The primary culprit is the inverse

function $\Psi^{-1}$ appearing in the expression of $\zeta(\boldsymbol{u};\Theta)$, which doesn't admit a closed analytical form. To further elaborate, let's look at the cumulative distribution function (Equation 3) of a univariate $m$-component Gaussian mixture, along the $r^{th}$ dimension (note that $\Theta^r = \{\alpha^l, \boldsymbol{\mu}_r^l, \Sigma_{rr}^l\}_{l=1}^m$). It is easy to verify that the corresponding inverse, $\boldsymbol{z}_r = \Psi_r^{-1}(\boldsymbol{u}_r)$, cannot be written explicitly, thus necessitating a numerical solution for the same.

$$\boldsymbol{u}_r = \Psi_r(\boldsymbol{z}_r;\Theta^r) = \frac{1}{2}\sum_{l=1}^{m}\alpha^l\left[1+\text{erf}\left(\frac{\boldsymbol{z}_r-\boldsymbol{\mu}_r^l}{\sqrt{2\Sigma_{rr}^l}}\right)\right] \tag{3}$$

Bilgrau et al. (2016) proposed an efficient inversion scheme based on linear interpolation on a sufficiently sized grid and exploiting the fact that $\Psi_r(\boldsymbol{z}_r)$ is monotonic. Furthermore, they used an empirical approximation of the *error function*, erf$(\cdot)$, in Equation 3. Although, these measures alleviate some issues, obtaining $\Psi_r^{-1}(\boldsymbol{u}_r)$ remains the bottleneck in the evaluation of the likelihood function in Equation (2). The overall cost to evaluate this function for $n, d \gg m$ turns out to be $O(mdg + nd\log g)$, where $g$ is the grid size used for interpolation. The first term is due to the cost involved in evaluating (3) on $g$ grid-points for $d$ dimensions. The second term is the cost of linear interpolation of $d$, $n$-dimensional vectors.

Obtaining the partial derivatives of (2), for gradient-based MLE, w.r.t $\Theta^r$ is even more challenging. In addition to the fact that the likelihood function lacks a closed form, it comprises logarithms of summands with exponential terms in both numerator and denominator, thus, making the derivation of analytical derivatives nontrivial. As a result, prior works (Tewari et al., 2011; Bilgrau et al., 2016) relied on finite difference (FD) approximation of the gradient of the GMCM log-likelihood function. Although effective for small problems, this scheme scales poorly with problem dimensions. To make this point clear, let us first understand the computational complexity of FD gradient approximation. Since the GMC distribution has $O(md+md^2)$ parameters, the complexity of FD approximation becomes $O(md^2C_{\ell(\Theta|U)})$ (for $d \gg m$), with $C_{\ell(\Theta|U)}$ being the cost to evaluate the function in (2) (details of which are provided in the previous paragraph). Therefore, the overall complexity of FD gradient-based MLE scales as $O(m^2d^3g + nmd^3\log g)$. The grid size, $g$, dependent complexity, and the possibility of low-quality gradients because of extensive approximations, call for improvements in GMC parameter estimation. This paper does that in three ways; 1) by proposing a numerical scheme to evaluate $\Psi_r^{-1}(\boldsymbol{u}_r;\Theta^r)$ while providing analytical partial derivatives for the same, 2) by deriving a correct formulation of the EM algorithm for GMCM, which eluded previous attempts at it, and 3) by presenting a view of GMCM that involves bijective transformations of a base distribution, which paves the way for GMCM to benefit from modern probabilistic programming languages (PPLs).

## 3. GMCM as a transformed distribution

As noted earlier, there has been a recent surge in approaches to compose joint distributions by transforming a simple base distribution (e.g., isotropic Gaussian) through a series of bijective transformations, a.k.a Normalizing Flows or NFs Kobyzev et al. (2021). As long as the transformations (both the forward and the inverse) and the determinant of the corresponding Jacobian matrices are well-defined, one can trivially chain any arbitrary set of bijections to the base distribution to yield highly expressive joint distributions. The likelihood function evaluation is done by invoking the *change of variable* formula (Mood et al., 1973, Chapter V.5), and the gradient of the same is obtained via automatic differentiation. Modern PPL languages, such as TensorFlow-Probability (Dillon et al., 2017) and Pyro (Bingham et al., 2019) offer succinct and convenient APIs to construct such transformed distributions, which is undoubtedly a boon for communities of researchers and practitioners alike.

GMCM can also be synthesized as a transformed distribution using the aforementioned PPL constructs. This is illustrated via a synthetic 2-D example in Figure 3, wherein the base distribution is a 2-component GMM (Figure 3a). The base distribution is then transformed via two bijective mappings; the first (Figure 3b) comprises marginal distribution functions of the base GMM distribution, $\Psi_r(\cdot)$, and the second, the quantile functions, $F_r^{-1}(\cdot)$, of desired marginal distributions (Figure 3c). Hence, the generative process induced by GMCM can be specified as follows,

$$\boldsymbol{z} \in \mathbb{R}^d \sim \text{GMM}(\Theta)$$

$$\boldsymbol{u} \in [0,1]^d = [\Psi_1(\boldsymbol{z}_1;\Theta^1), \Psi_2(\boldsymbol{z}_1;\Theta^2), \cdots, \Psi_d(\boldsymbol{z}_d;\Theta^d)] \tag{4a}$$

$$\boldsymbol{x} \in \mathbb{V}^d = [F_1^{-1}(\boldsymbol{u}_1), F_2^{-1}(\boldsymbol{u}_2), \cdots, F_d^{-1}(\boldsymbol{u}_d)]. \tag{4b}$$

Note that the vector space $\mathbb{V}^d$ is formed by the support of the marginal distribution functions $F_1, F_2, \cdots, F_d$ i.e. $\mathbb{V}^d \equiv \text{supp}(F_1) \times \text{supp}(F_2) \cdots \times \text{supp}(F_d)$. The aforementioned generative process is also depicted in Figure 2 along with the propagated density function in the transformed space $X$. It is not too hard to verify that this joint density can be further written as the product of GMC density function (Equation (1)) and the separately learned marginal densities $f_r(\boldsymbol{x}_r)$.

$$p(\boldsymbol{x};\Theta) = \zeta(\boldsymbol{u};\Theta)\cdot\prod_{r=1}^{d} f_r(\boldsymbol{x}_r), \tag{5}$$

While conceptually similar, there are a few notable differences between GMCM and the NF-based distributions. First, in GMCM the base distribution encodes the parameters of

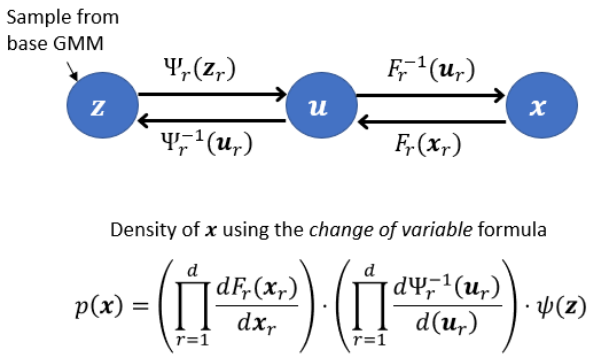


*Figure 2.* Illustration of the transformations involved in a GMCM. Samples $\boldsymbol{z}$ from the base distribution (a GMM) are transformed via two dimension-wise bijective mappings. The first corresponds to the marginal CDFs (of the joint GMM) along each dimension, while the latter comprises quantiles of a priori learned marginal CDFs. The joint density in the transformed space $X$ can be obtained by inverting a sample to the base space $Z$ and invoking the *change of variable* formula as shown in the bottom portion of the figure. The bracketed expressions correspond to the determinants of the Jacobians of the two transformations. The aforementioned transformations are implemented, in the contributed code (Tewari, 2023), using the *Bijector* class available in the TensorFlow probability package.

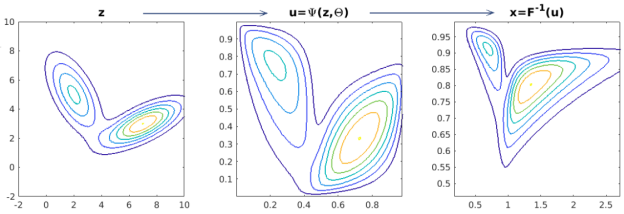


*Figure 3.* Illustration of the transformations induced by a GMCM. The left panel shows the density contours of a 2-component GMM with parameters $\boldsymbol{\alpha} = \{0.45,\ 0.55\}$, $\boldsymbol{\mu} = \{[2\ 5], [7\ 3]\}$ and $\Sigma = \{[1.5\ -1.3; -1.3\quad 3], [3\quad 1.2; 1.2\quad 1]\}$. The middle panel shows the contours under the transformation by the marginal distribution functions [Equation (3)]. The right panel shows the transformation by the quantile functions of $Lognormal(0, 0.5)$ and $Beta(10, 2)$ distributions along $x$ and $y$ dimensions, respectively. Note that $\boldsymbol{x} \in \mathbb{R}_+ \times [0,\ 1]$ owing to the Lognormal and Beta marginals.

interests (i.e., those that induce a dependency structure), while in NFs, those reside within the bijective mappings. Second, the bijections in GMCM are dimension-wise independent with separate parameters. In NF-based distributions, the bijections intricately couple different dimensions. Third, by virtue of the second point and the fact that the base distribution (GMM) is marginalizable, GMCM is also marginalizable. The latter is a significant advantage over non-marginalizable NF-based distributions, wherein some flexibility is sacrificed in favor of expressivity (see Gilboa et al., 2021, for a detailed discussion on this subject). Note that the goal here is to compare and contrast GMCM with the other state-of-the-art multivariate models, and not to prescribe one over the other. Putting it differently, GMCM can be another multivariate modeling tool in the repertoire of data modelers, which offers the simplicity and flexibility (marginalization) of GMMs while being more expressive.

### 3.1. A numerical scheme to compute GMM quantiles

Section 2.1 emphasized the need for a method to compute a univariate GMM's quantile, $\Psi_r^{-1}(\boldsymbol{u}_r;\ \Theta^r)$, that does better than the linear scaling of previously proposed interpolation-based methods. Here, a computationally efficient alternative is proposed where the desired inversion is sought as the root of the expression $\boldsymbol{u}_r - \Psi_r(\boldsymbol{z}_r; \Theta^r)$, thus opening the door to a rich set of root-finding algorithms such as the *Secant method*. Since, the forward function $\Psi_r(\boldsymbol{z}_r; \Theta^r)$ is twice continuously differentiable and the root sought is a simple root, the convergence of the Secant method is near quadratic (Díez, 2003). Therefore, employing a root-finding algorithm to obtain the inverse would need far less number of function evaluations than an interpolation-based inversion. Chandrupatla's algorithm (Chandrupatla, 1997) further offers an improvement over the Secant method for functions that are flatter around the root; a highly likely behavior of GMM CDFs, $\Psi_r(\boldsymbol{z}_r; \Theta^r)$, when the mixing components are well separated. Therefore, the contributed GMCM code (Tewari, 2023) employs the parallel implementation of Chandrupatla's algorithm via the TensorFlow Probability package.

Efficient computation of GMM quantiles ($\Psi_r^{-1}$), however, only partly solves the problem. The partial derivatives of $\Psi_r^{-1}(\cdot; \Theta^r)$ are also needed with respect to $\Theta^r = \{\alpha_l, u_r^l, \sigma_r^l\}_{l=1}^m$. Interestingly, these partial derivatives can be obtained analytically despite the quantile function lacking a closed form. Appendix D provides the derivation of the aforementioned partial derivatives. PPLs, such as TensorFlow-Probability, allow one to easily embed such custom derivatives so as to allow autodiff to use the same when backpropagating gradients through expression graphs. One may argue the need for analytical derivatives of $\Psi_r^{-1}(\cdot; \Theta^r)$ at all, when PPLs, in principle, can autodiff through iterative, numerical routines such as the secant method. While the argument is correct, autodiff through such routines can easily produce "...large expression graphs, which can lead to floating point precision errors, excessive memory usage, and slow computation." (Margossian, 2018).

Putting it all together (efficient quantile computation and analytical partial derivatives), roughly two orders of magnitude speedup is observed in GMCM likelihood gradient computation over the finite-difference (FD) method. This is

shown in Figure 4 which plots the CPU times to obtain the FD and analytical gradients on synthetic datasets of different dimensions ($d \in \{10, 30, 50, 70, 90, 110\}$) and batches sizes ($n \in 1, 10, 100$).

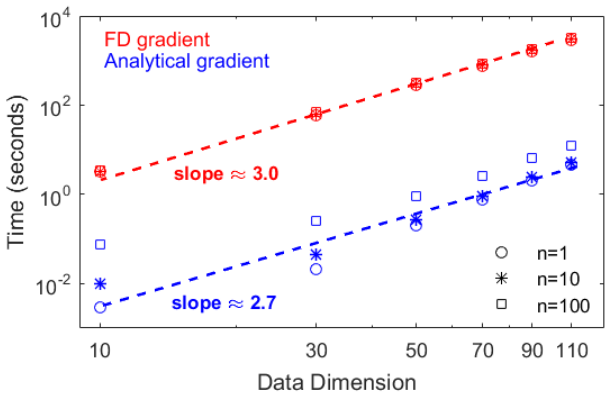


*Figure 4.* Plots (log-log scale) of the CPU time vs. data dimension for the analytical and the FD gradient computations of the GMC likelihood function (Equation (2)). The effect of batch size ($n = 1, 10, 100$) on CPU times is also shown. The dashed lines are the robust linear fits corresponding to the batch size of $n = 10$. The cubic complexity with data dimensions is apparent for both the approaches from the slopes of the linear fits, which is in line with the discussion on the computation complexity in Section 2.1.

## 4. Identifiability of GMCM

Another issue that plagues GMCM is that of *non-identifiability*. Identifiability is a key property that determines if a generative model's true parameters can be learned asymptotically with the number of samples. Finite mixture models are known to suffer from the issue of parameter non-identifiability since the likelihood is invariant under a permutation of component labels (Stephens, 2000). This is commonly known as *label switching* problem. However, GMCM suffers from another form of parameter non-identifiability as stated in the theorem below.

**Theorem 4.1.** *Let $U$ be a dataset generated by a $m$-component Gaussian Mixture Copula distribution with true parameters set $\Theta^* = \{\boldsymbol{\mu}^{l*}, \Sigma^{l*}, \alpha^{l*}\}_{l=1}^m$. Denote the log-likelihood of the observed data, with respect to the true model, as $\ell_\zeta(\Theta^*|U)$. Define another parameter set $\Theta = \{A\boldsymbol{\mu}^{l*} + \boldsymbol{b},\ A^T\Sigma^{l*}A,\ \alpha^{l*}\}_{l=1}^m$, where $A$ is any diagonal positive definite matrix and $\boldsymbol{b}$ a real-valued vector. Then, $\ell_\zeta(\Theta|U) = \ell_\zeta(\Theta^*|U)$.*

Refer to Appendix B.1 for the proof. A practical repercussion of this result is that the true parameters of a GMC distribution can never be uniquely identified −even after addressing the label switching problem− because the likelihood function has infinitely many maximizers. Readers can refer to White (1982) for a detailed exposition on the subject of identifiability in parametric models. Bilgrau et al. (2016) noted this form of non-identifiability in GMCM, although did not prove it. They proposed an ad hoc solution that involved enforcing the first component to have zero mean and unit variance along each dimension. Nevertheless, as noted in their paper, the non-identifiability issue persisted under certain conditions. Here an alternative solution, formalized in Theorem 4.2, is proposed that renders GMCM identifiable up to the permutation of component labels.

**Theorem 4.2.** *Denote $\boldsymbol{g} \in \mathbb{R}^d$ and $\boldsymbol{h} \in \mathbb{R}_+^d$ as any real-valued vectors; the latter being strictly positive. A $m$-component, $d$-dimensional Gaussian Mixture Copula distribution parametrized by $\Theta = \{\boldsymbol{\mu}^l, \Sigma^l, \alpha^l\}_{l=1}^m$ is identifiable, up to the permutation of component labels, if the following two conditions are met for the specified $\boldsymbol{g}$ and $\boldsymbol{h}$.*

$$\sum_{l=1}^m \alpha^l\, \boldsymbol{\mu}_r^l = \boldsymbol{g}_r\,, \qquad \forall r \in \{1, 2, \cdots, d\} \tag{6}$$

$$\sum_{l=1}^m \left[\alpha^l \left(\Sigma_{rr}^l + (\boldsymbol{\mu}_r^l)^2\right)\right] - \boldsymbol{g}_r^2 = \boldsymbol{h}_r, \qquad \forall r \in \{1, 2, \cdots, d\} \tag{7}$$

The proof is given in Appendix B.2. The choice of $\boldsymbol{g}$ and $\boldsymbol{h}$ is rather arbitrary. For convenience, the former can be set as $\mathbf{0}^d$ (vector of all zeros) and the latter $\mathbf{1}^d$ (vector of all ones). During the parameter estimation, these constraints can be specified in the form of suitable priors, e.g. Gaussian priors as shown in Equations (8) and (9), resulting in a well-defined and unique Maximum A Posteriori (MAP) solution. The strength of these priors can be controlled by the parameter $\sigma$ (larger values lead to weaker priors). During experimentation, a value of $\sigma = 0.01$ worked well in balancing the trade-off between these priors and the GMCM likelihood.

$$\mathcal{N}\Big(\sum_{l=1}^m \alpha^l\, \boldsymbol{\mu}_r^l \;\; \boldsymbol{g}_r, \sigma\Big), \tag{8}$$

$$\mathcal{N}\Big(\sum_{l=1}^m \left[\alpha^l \left(\Sigma_{rr}^l + (\boldsymbol{\mu}_r^l)^2\right)\right] - \boldsymbol{g}_r^2 \;\; \boldsymbol{h}_r, \sigma\Big), \tag{9}$$

$$\forall r \in \{1, 2, \cdots, d\}$$

## 5. The EM algorithm for GMCM

The EM algorithm has garnered popularity for MLE in Gaussian mixture models given that it 1) automatically satisfies the probabilistic constraints and positive definiteness of covariance matrices, 2) doesn't require explicit gradients, and 3) dispenses with the learning rate needed for gradient-based approaches (Xu & Jordan, 1996). The underpinning of EM

is a two-step process, the *Expectation* (E)-step that finds a lower bound of the *incomplete data* log-likelihood function (e.g., in Equation (2)), and the *Maximization* (M)-Step optimizes this lower bound (either partially or fully) to arrive at the next iterate. A repeated application of the E and M steps ensures a monotonic increase of the data log-likelihood until local convergence is achieved. Readers may refer to Bilmes (1997) and Salakhutdinov et al. (2003) for detailed treatments on the EM algorithm for GMMs. Although the EM algorithm for GMCMs would follow the same general construct, the E and the M steps are considerably harder than those for GMM. The previous attempts at it (Tewari et al., 2011; Bhattacharya & Rajan, 2014) do not, systematically, derive and maximize the true lower bound of the incomplete data log-likelihood. Instead, certain assumptions are made that allow tweaking of the GMM's EM algorithm for learning GMCM's parameters. As a result, both of these algorithms need additional checks, at each iteration, to ensure a monotonically increasing likelihood function. They are referred as pseudo-EM (PEM) algorithms for later benchmarking experiments. In summary, a provably correct EM algorithm has remained elusive for GMCM, and this paper closes that gap.

### 5.1. E-Step

The ensuing derivation closely follows the exposition in Bilmes (1997), which presents the EM algorithm for GMM in great detail. Assuming access to $\boldsymbol{y}$, a $n$-dimensional vector of latent variables that co-occurs with the observed data $U$, the *complete data* log-likelihood function can be written as,

$$\ell_{comp}(\Theta|U,\boldsymbol{y}) = \sum_{i=1}^{n} \log\left(\frac{\alpha^{\boldsymbol{y}_i}\phi\left(Z_{:i};\Theta^{\boldsymbol{y}_i}\right)}{\prod_{r=1}^{d}\psi_r\left(Z_{ri};\Theta^r\right)}\right). \quad (10)$$

The latent variable $\boldsymbol{y}_i$ denotes the index of the Gaussian component from which the dependence of the $i^{th}$ data sample $U_{:i}$ is derived. The function $\phi(\cdot)$ is the multivariate Gaussian density, $\Theta^{\boldsymbol{y}_i}$ and $\Theta^r$ represent the parameters associated with the component $\boldsymbol{y}_i$ and the dimension $r$, respectively. Also, $Z_{:i}$ and $Z_{ri}$ are used to denote $\Psi^{-1}\left(U_{:i}\right)$ and $\Psi_r^{-1}(U_{ri})$, respectively. Note that the denominator does not depend on the latent variable $\boldsymbol{y}_i$, since the marginal densities, $\psi_r(\cdot)$ are not component specific. The E-step involves the derivation of the expected value of the complete data log-likelihood (Equation 10) with respect to the posterior distribution of the latent variables given the data and the current parameter estimates, say $\hat{\Theta}$. This posterior distribution in this case is $P(\boldsymbol{y}|U,\hat{\Theta}) = \prod_{j=1}^{n} P\left(\boldsymbol{y}_j|U_{:j},\hat{\Theta}\right)$. Following some tedious but straightforward manipulations (refer to Appendix C for details), the expectation of complete data log-likelihood, $Q(\Theta,\hat{\Theta})$, can be written as,

$$\begin{aligned} Q(\Theta,\hat{\Theta}) = &\sum_{i=1}^{n}\sum_{\boldsymbol{y}_i=1}^{m}\left(\log(\alpha^{\boldsymbol{y}_i}) - \frac{\log(|\Sigma^{\boldsymbol{y}_i}|)}{2}\right)G_{i\boldsymbol{y}^i} \\ &- \sum_{i=1}^{n}\sum_{\boldsymbol{y}_i=1}^{m}\left(\frac{\bar{Z}_{:i}^T(\Sigma^{\boldsymbol{y}_i})^{-1}\bar{Z}_{:i}}{2}\right)G_{i\boldsymbol{y}^i} \\ &- \sum_{i=1}^{n}\sum_{r=1}^{d}\log\left(\psi_r(Z_{ri};\Theta^r)\right), \end{aligned} \quad (11)$$

where $\bar{Z}_{:i} = Z_{:i} - \mu^{\boldsymbol{y}_i}$ is the mean adjusted vector and $G_{i\boldsymbol{y}^i}$ is given by Equation(15). It should be noted that, unlike the GMM, the E-step in GMCM does not completely remove the logarithm over a sum of exponential terms (see in the third term of Equation 11). Thus, the maximization of (11) does not yield, unlike the case of GMM, closed-form updates for the model parameters $\Theta$; thereby necessitating a gradient-based M-step. Therefore, it can be argued that the EM algorithm does not enjoy the same benefits for GMCMs —as it does for GMMs—over direct likelihood maximization. Nevertheless, the accurately derived E-step can still be maximized (or partially maximized) with a gradient-based M-step while guaranteeing a monotonically increasing incomplete data log-likelihood function (unlike the PEM algorithms proposed in previous works). When the M-step is carried out partially, it would result in a *generalized*-EM (GEM) algorithm. Section 6 compares the proposed GEM with the aforementioned PEM algorithms.

## 6. Experimental Results

This section provides empirical evidence to support the claims made in this paper using synthetic and real-world datasets. The GMCM is coded up in Python using the TensorFlow-Probability package[1]. The experiments aim to convey three key messages, 1) GMCM becomes identifiable in accord with the statement of Theorem 4.2, 2) the proposed EM algorithm outperforms the previously published PEM algorithms, and 3) density estimation via GMCM is comparable with the state-of-the-art models on UCI benchmark datasets. For other real-world applications, readers may refer to (Bilgrau et al., 2012; Wang et al., 2014; Yu et al., 2013; Bayestehtashk & Shafran, 2015), where GMCMs are used for tasks such as prediction, classification, anomaly detection, dependence characterization, etc.

For the first experiment, a 3-dimensional synthetic dataset is generated by randomly instantiating an arbitrary 2-component GMC distribution (ground truth). One thousand random samples are generated from this distribution to yield a matrix $U \in [0\ 1]^{3\times 1000}$, which serves as the training dataset to learn a GMC distribution via direct likelihood

[1]Code used in this analysis is available here (Tewari, 2023).

maximization. The maximization is carried out using the Adam optimizer (Kingma & Ba, 2017) with the default learning rate of $1\text{E}^{-3}$. The emphasis here is on the ability to recover the parameters of the true data-generating GMC distribution. Figure 5 shows the evolution of the parameters for the two cases; *unregularized* (without the identifiability priors as given by Equations (8) and (9)) and *regularized* (with identifiability priors). Starting from the same initial point, the plots show the evolution of the iterates until numerical convergence (up to 2000 iterations). The green squares show the true GMC parameters. The divergence of iterates from the ground truth is quite apparent for the unregularized case. On the contrary, the iterates converge to the true parameter values for regularized MLE. Although the results are shown only for the mean parameters $\{\boldsymbol{\mu}^l\}_{l=1}^3$, the same holds true for other GMC parameters e.g. covariance parameters. Bear in mind that the identifiability priors do not impose any restrictive assumption on the generative process of the data. They are merely making an ill-posed problem (with multiple equivalent solutions), well-posed (with a unique global solution) (see Theorem (4.2)).

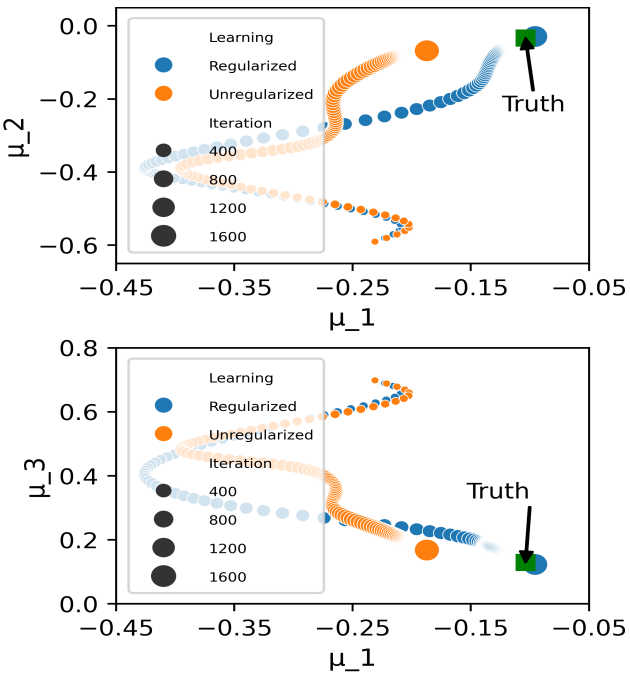


*Figure 5.* The manifestation of identifiability in GMC distribution is shown empirically. Both plots show the evolution of the mean parameters ($\mu_i$), from the same initial point until numerical convergence (2000 iterations), with and without regularization priors given by Equations (8) and (9). The true values of the parameters are shown by green squares.

The second experiment compares the performance of the proposed generalized EM algorithm with the two pseudo-EM algorithms published in Bhattacharya & Rajan (2014) and Tewari et al. (2011), referred to here as $\text{PEM}_1$ and $\text{PEM}_2$, respectively. The key performance indicator here is the log-likelihood value attained at the convergence of these algorithms. To ensure an exhaustive comparison, 100 datasets are generated by following the same procedure as in experiment 1. For each dataset, the GMC parameters are learned by the three algorithms with identical initialization. Figure 6(a) plots the log-likelihood vs. iteration, from the three algorithms, for one such dataset. GEM can be seen to converge to a higher log-likelihood value compared to $\text{PEM}_1$ and $\text{PEM}_2$. This observation is quite consistent over other datasets. Figure 6(b) summaries the results over all the datasets by showing the box-plots of the log-likelihood ratios, $\log\left(\frac{\mathcal{L}(\Theta^{GEM}|U)}{\mathcal{L}(\Theta^{PEM_1}|U)}\right)$ and $\log\left(\frac{\mathcal{L}(\Theta^{GEM}|U)}{\mathcal{L}(\Theta^{PEM_2}|U)}\right)$, of the converged models. A significantly positive median and the quantile values confirm the superior performance of GEM over $\text{PEM}_1$ and $\text{PEM}_2$.

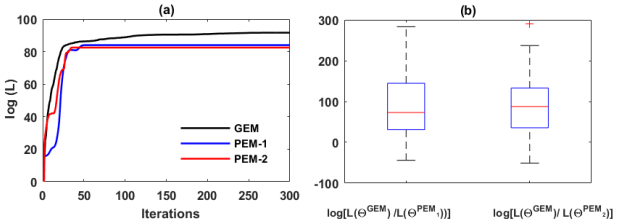


*Figure 6.* (a) log-likelihood vs. iteration for the three EM algorithms on a simulated dataset (b) Box-plots of converged log-likelihood ratios, $\log\left(\frac{\mathcal{L}(\Theta^{GEM}|U)}{\mathcal{L}(\Theta^{PEM_1}|U)}\right)$ and $\log\left(\frac{\mathcal{L}(\Theta^{GEM}|U)}{\mathcal{L}(\Theta^{PEM_2}|U)}\right)$, by repeating this experiment on 100 such simulated datasets.

Finally, GMCM is learned on several density estimation benchmarks from the UCI repository, after following the pre-processing step described in (Papamakarios et al., 2017). The test log-likelihoods of GMCM with several other marginalizable and non-marginalizable density models are presented in Table 6. In order to have the same complexity, the mixture models (GMCM and GMM) were instantiated with the 40 mixing components. This number was ascertained by a grid search over $\{10,20,30,40,50\}$ and tracking the likelihood of validation datasets. The test log-likelihood numbers for the other models are lifted from other benchmark studies (Papamakarios et al., 2017; Gilboa et al., 2021). The performance of GMCM is found to be comparable (and some time better e.g., on Miniboone) to its marginalizable counterparts. However, the non-marginalizable variants clearly have superior performance, but at the cost of losing the ability to marginalize. Nevertheless, the goal here is not to prescribe one modeling framework over the other (that choice is largely application dependent), but rather to estab-

*Table 1.* Average test log-likelihood achieved by different density estimation models on UCI benchmark datasets. The models are grouped as *non-marginalizable* (top panel) and *marginalizable* (bottom panel). The best-performing model for each group and dataset is shown in bold. Error bars correspond to two standard deviations. The references for the methods in the top panel can be found in Gilboa et al. (2021).

| NON-MARG. | POWER | GAS | HEPMASS | MINIBOONE |
|---|---|---|---|---|
| KINGMA 2018 | 0.17 ± .01 | 8.15 ± .40 | -18.92 ± .08 | -11.35 ± .07 |
| GRATHWOHL 2019 | 0.46 ± .01 | 8.59 ± .12 | -14.92 ± .08 | -10.43 ± .04 |
| HUANG 2018 | 0.62 ± .01 | 11.96 ± .33 | -15.08 ± .40 | -8.86 ± .15 |
| OLIVA 2018 | 0.60 ± .01 | **12.06 ± .02** | -13.78 ± .02 | -11.01 ± .48 |
| DE CAO 2019 | 0.61 ± .01 | 12.06 ± .09 | -14.71 ± .38 | -8.95 ± .07 |
| BIGDELI 2020 | 0.97 ± .01 | 9.73 ± 1.14 | **-11.3 ± .16** | **-6.94 ± 1.81** |
| GILBOA 2021 | **1.78 ± .12** | 8.43 ± .04 | -18.0 ± 0.91 | -18.6 ± .47 |
| **MARG.** | **POWER** | **GAS** | **HEPMASS** | **MINIBOONE** |
| GAUSSIAN | -7.74 ± .02 | -3.58 ± .75 | -27.93 ± .02 | -37.24 ± 1.07 |
| GMM | -0.26 ± .11 | 5.84 ± 1.13 | -23.88 ± .28 | -20.91 ± 1.26 |
| GMCM | 0.14 ± .06 | 6.13 ± .14 | -22.23 ± .18 | **-16.41 ± .78** |
| GILBOA 2021 | **0.57 ± .01** | **8.92 ± .11** | **-20.08 ± .06** | -29.01 ± .06 |

lish GMCM as a useful addition to data modelers' repertoire of tools. Moreover, average log-likelihood is only one of the metrics for model evaluation and need not imply good performance on the task at hand (Theis et al., 2015).

## 7. Discussion

This paper addresses a few outstanding issues with the parameter estimation of the GMCM. These issues have withheld its use as a mainstream data modeling tool, despite its superior expressivity than the closely related and widely used GMM. The first one is that of parameter non-identifiability which has been well-acknowledged but loosely addressed in previous works. The proposed identifiability priors mitigate this issue in a principled manner. The second issue pertains to the intractability of GMCM's likelihood. While this inherent issue persists, a superior numerical scheme and the associated analytical partial derivatives, presented in this paper, go a long way to alleviate the issue. Lastly, the paper also presents a provable correct *generalized*-EM algorithm for GMCM. Previous attempts at it, proposed parameter update rules that do not maximize the true lower bound of the GMCM log-likelihood. As a result, additional checks and corrections were needed to ensure a monotonically increasing log-likelihood during EM updates. The paper also argues that the GMCM does not enjoy the same benefits, as the GMM, from the EM algorithm, thus prescribes direct likelihood maximization for parameter estimation.

An interesting unexplored aspect pertains to the Bayesian parameter estimation of GMCM. Previously, the non-identifiability of GMCM would have caused performance issues in posterior approximations via both sampling or variational inference methods. How well the proposed identifiability priors (Equations (8) and (9)) improve the performance of posterior approximation methods remains to be seen. Another direction would be to investigate GMCM under the conditioning operation. Although GMCM is closed under marginalization, it need not be under conditioning. The latter would be a useful property; e.g. when GMCM is intended to be used for regression applications. A remaining direction would be to induce sparsity in the precision matrices of the Gaussian components. This would bring significant benefits in settings when data is scarce, and there's a significant risk of over-training. Finally, the applicability of GMCM for multimodal posterior approximation, arising in a myriad of ill-posed inverse problems (Franck & Koutsourelakis, 2015; Tewari et al., 2022; Zhang & Curtis, 2021; Tewari & Paiva, 2022), could be worth exploring.

## References


Bayestehtashk, A. and Shafran, I. Efficient and accurate multivariate class conditional densities using copula. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2015)*, pp. 3936–3940. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178709.

Bedford, T. and Cooke, R. M. Vines: A new graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068, 2002. ISSN 00905364.

Bhattacharya, S. and Rajan, V. Unsupervised learning using gaussian mixture copula model. In *21st International Conference on Computational Statistics COMPSTAT 2014*, 2014.

Bilgrau, A., E., P., Rasmussen, J., J., H., D., K., and Boegsted, M. Gmcm: Unsupervised clustering and meta-analysis using gaussian mixture copula models. *Journal of Statistical Software, Articles*, 70(2):1–23, 2016. ISSN 1548-7660. doi: 10.18637/jss.v070.i02.

Bilgrau, A. E., Bergkvist, K. S., Kjeldsen, M. K., Falgreen, S., Rodrigo-Domingo, M., Schmitz, A., Bødker, J. S., Nyegaard, M., Johnsen, H. E., Dybkær, K., et al. Quantification of reproducibility of microarray experiments by semi-parametric mixture models applied to the detection of differentially expressed genes in b-cell subpopulations. In *Forskningens Dag 2012*, 2012.

Bilgrau, A. E., Boegsted, M., and Eriksen, P. S. Gmcm: Fast estimation of gaussian mixture copula models. https://github.com/AEBilgrau/GMCM, 2017.

Bilmes, J. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, ICSI, 1997.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., and Goodman, N. D. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20: 28:1–28:6, 2019.

Chandrupatla, T. R. A new hybrid quadratic/bisection algorithm for finding the zero of a nonlinear function without using derivatives. *Advances in Engineering Software*, 28(3):145–149, 1997. ISSN 0965-9978. doi: https://doi.org/10.1016/S0965-9978(96)00051-8. URL https://www.sciencedirect.com/science/article/pii/S0965997896000518.

Cherubini, U., Luciano, E., and Vecchiato, W. *Copula Methods in Finance*. The Wiley Finance Series. Wiley, 2004. ISBN 9780470863459.

Czado, C. Pair-copula constructions of multivariate copulas. In Jaworski, P., Durante, F., Härdle, W., and Rychlik, T. (eds.), *Copula Theory and Its Applications*, pp. 93–109, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M. D., and Saurous, R. A. Tensorflow distributions. *ArXiv preprint arXiv:1711.10604*, abs/1711.10604, 2017.

Durante, F. and Sempi, C. Copula theory: An introduction. In Jaworski, P., Durante, F., Härdle, W., and Rychlik, T. (eds.), *Copula Theory and Its Applications*, pp. 3–31, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

Díez, P. A note on the convergence of the secant method for simple and multiple roots. *Applied Mathematics Letters*, 16(8):1211–1215, 2003. ISSN 0893-9659. doi: https://doi.org/10.1016/S0893-9659(03)90119-4.

Elidan, G. Copula bayesian networks. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 23*, pp. 559–567. Curran Associates, Inc., 2010.

Elidan, G. *Copulae in Mathematical and Quantitative Finance: Proceedings of the Workshop Held in Cracow, 10-11 July 2012*, chapter Copulas in Machine Learning, pp. 39–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

Franck, I. and Koutsourelakis, P. Multimodal, high-dimensional, model-based, bayesian inverse problems with applications in biomechanics. *Journal of Computational Physics*, 329, 12 2015. doi: 10.1016/j.jcp.2016.10.039.

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

Genest, C., Gendron, M., and Bourdeau-Brien, M. The advent of copulas in finance. *The European Journal of Finance*, 15(7-8):609–618, 2009. doi: 10.1080/13518470802604457.

Gilboa, D., Pakman, A., and Vatter, T. Marginalizable density models. *ArXiv preprint arXiv:2106.04741*, 2021.

Hanea, A. M., Kurowicka, D., and Cooke, R. M. Hybrid method for quantifying and analyzing bayesian belief nets. *Quality and Reliability Engineering International*, 22(6):709–729, 2006. doi: 10.1002/qre.808.

Joe, H. Parametric families of multivariate distributions with given margins. *Journal of Multivariate Analysis*, 46 (2):262 – 282, 1993. ISSN 0047-259X.

Joe, H. and Xu, J. J. The estimation method of inference functions for margins for multivariate models, 1996.

Kasa, S. R. and Rajan, V. Automatic differentiation in mixture models. *ArXiv preprint arXiv:1812.05928*, 2018.

Kasa, S. R. and Rajan, V. Improved inference of gaussian mixture copula model for clustering and reproducibility analysis using automatic differentiation. *Econometrics and Statistics*, 22:67–97, 2022. ISSN 2452-3062. doi: https://doi.org/10.1016/j.ecosta.2021.08.010. The 2nd Special issue on Mixture Models.

Kasa, S. R., Bhattacharya, S., and Rajan, V. Gaussian mixture copulas for high-dimensional clustering and dependency-based subtyping. *Bioinformatics*, 36(2): 621–628, 08 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz599.

Kim, J. M., Jung, Y. S., Sungur, E. A., H.Han, K., Park, C., and Sohn, I. A copula method for modeling directional dependence of genes. *BMC Bioinformatics*, 9(1):1–12, 2008. doi: 10.1186/1471-2105-9-225.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *ArXiv preprint arXiv:1412.6980*, 2017.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, 2021. doi: 10.1109/TPAMI.2020.2992934.

Kojadinovic, I. and Yan, J. Modeling Multivariate Distributions with Continuous Margins Using the Copula R Package. *Journal of Statistical Software*, 34(i09), 2010. doi: http://hdl.handle.net/10.

Kosmidis, I. and Karlis, D. Model-based clustering using copulas with applications. *Statistics and Computing*, 26 (5):1079–1099, Sep 2016. ISSN 1573-1375. doi: 10.1007/s11222-015-9590-5.

Kurowicka, D. and Joe, H. (eds.). *Dependence Modeling: Vine Copula Handbook*. World Scientific Publishing Co. Pte. Ltd., 2010.

Li, Q., Brown, J. B., Huang, H., and Bickel, P. J. Measuring reproducibility of high-throughput experiments. *The Annals of Applied Statistics*, 5(3):1752–1779, 09 2011. doi: 10.1214/11-AOAS466.

Liu, H., Lafferty, J., and Wasserman, L. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *J. Mach. Learn. Res.*, 10:2295–2328, December 2009. ISSN 1532-4435.

Ma, C. and Wang, X. P. Application of the gini correlation coefficient to infer regulatory relationships in transcriptome analysis. *Plant physiology*, 160(1):190–203, 2012.

Marbac, M., Biernacki, C., and Vandewalle, V. Model-based clustering of gaussian copulas for mixed data. *Communications in Statistics - Theory and Methods*, 46(23):11635–11656, 2017. doi: 10.1080/03610926.2016.1277753.

Margossian, C. C. A review of automatic differentiation and its efficient implementation. *ArXiv preprint arXiv:1811.05031*, 2018.

Mazo, G. A semiparametric and location-shift copula-based mixture model. *Journal of Classification*, 34(3): 444–464, Oct 2017. ISSN 1432-1343. doi: 10.1007/s00357-017-9243-9.

Mood, A., Graybill, F., and Boes, D. *Introduction to the Theory of Statistics*. International Student edition. McGraw-Hill, 1973. ISBN 9780070428645.

Nelsen, R. B. *An Introduction to Copulas*, chapter Methods of Constructing Copulas, pp. 51–108. Lecture notes in statistics. Springer, 1999a. ISBN 9780387986234.

Nelsen, R. B. *An Introduction to Copulas*. Lecture notes in statistics. Springer, 1999b. ISBN 9780387986234.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. NIPS'17, pp. 2335–2344, Red Hook, NY, USA, 2017. Curran Associates Inc.

Rajan, V. and Bhattacharya, S. Dependency clustering of mixed data with gaussian mixture copulas. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pp. 1967–1973. AAAI Press, 2016. ISBN 978-1-57735-770-4.

Rey, M. and Roth, V. Copula mixture model for dependency-seeking clustering. In Langford, J. and Pineau, J. (eds.), *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 927–934, New York, NY, USA, 2012. ACM.

Rychlik, T. Copulae in reliability theory(order statistics, coherent systems). In Jaworski, P., Durante, F., Härdle, W., and Rychlik, T. (eds.), *Copula Theory and Its Applications*, pp. 187–206, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

Salakhutdinov, R., Roweis, S., and Ghahramani, Z. On the convergence of bound optimization algorithms. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, UAI'03, pp. 509–516, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. ISBN 0-127-05664-5.

Stephens, M. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000. ISSN 1467-9868. doi: 10.1111/1467-9868.00265.

Tewari, A. Gaussian Mixture Copula Model, 2023. URL https://github.com/ashutosh-tewari/GMCM.

Tewari, A. and Paiva, A. R. Modeling and mitigation of occupational safety risks in dynamic industrial environments. *ArXiv preprint arXiv:2205.00894*, 2022.

Tewari, A., Giering, M., and Raghunathan, A. Parametric characterization of multimodal distributions with non-gaussian modes. In *ICDM Workshops*, pp. 286–292. IEEE Computer Society, 2011. ISBN 978-0-7695-4409-0.

Tewari, A., Wheelock, B., Clark, J., Foster, D., Li, M., and Marzouk, Y. *Subsurface uncertainty quantification with deep geologic priors: A variational Bayesian framework*, pp. 1745–1749. 2022. doi: 10.1190/image2022-3750132.1.

Theis, L., Oord, A. v. d., and Bethge, M. A note on the evaluation of generative models, 2015.

Wang, Y., Infield, D. G., Stephen, B., and Galloway, S. J. Copula-based model for wind turbine power curve outlier rejection. *Wind Energy*, 17(11):1677–1688, 2014. ISSN 1099-1824. doi: 10.1002/we.1661.

White, H. Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–25, 1982.

Xu, L. and Jordan, M. I. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996. doi: 10.1162/neco.1996.8.1.129.

Yu, J., Chen, K., Mori, J., and Rashid, M. M. A gaussian mixture copula model based localized gaussian process regression approach for long-term wind speed prediction. *Energy*, 61:673–686, 2013. ISSN 0360-5442. doi: http://dx.doi.org/10.1016/j.energy.2013.09.013.

Zhang, X. and Curtis, A. Bayesian full-waveform inversion with realistic priors. *Geophysics*, 86(5):A45–A49, 08 2021.

# A. Glossary of frequently used symbols

*Table 2.* Symbols and descriptions

| Symbol | Description |
|---|---|
| $d$ | data dimensions |
| $m$ | number of components in the mixture model |
| $n$ | number of data samples |
| $\Theta = \{\boldsymbol{\mu}^l, \Sigma^l, \alpha^l\}_{l=1}^m$ | parameter of $m$-component, $d$-dimensional GMM |
| $\Theta^r = \{\boldsymbol{\mu}_r^l, \Sigma_{rr}^l, \alpha^l\}_{l=1}^m$ | parameters of marginal GMM along the $r^{th}$ dimension ($\Theta^r \subset \Theta$) |
| $f_r$ | arbitrary univariate density function |
| $F_r, F_r^{-1}$ | the distribution and the quantile function corresponding to $f_r$ |
| $F$ | vector function defined as $F = [F_1, F_2, \cdots, F_d]$ |
| $\psi, \Psi$ | joint density and distribution function of GMM in $\mathbb{R}^d$ |
| $\psi_r, \Psi_r$ | density and distribution function of the of the GMM along the $r^{th}$ dimension |
| $\Psi_r^{-1}$ | quantile function corresponding to $\Psi_r$ |
| $\Psi^{-1}$ | vector function defined as $\Psi^{-1} = [\Psi_1^{-1}, \Psi_2^{-1}, \cdots, \Psi_d^{-1}]$ |
| $\boldsymbol{x} \in \mathbb{V}^d$ | real valued vector whose distribution is sought |
| $\boldsymbol{u} \in [0,1]^d = F(\boldsymbol{x})$ | vector of uniformly distributed random variables |
| $\boldsymbol{z} \in \mathbb{R}^d = \Psi^{-1}(\boldsymbol{u})$ | vector with quantile values of GMM marginals |
| $X \in \mathbb{V}^{d\times n}$ | matrix of $n$, $\boldsymbol{x}$ vectors arranged columnwise |
| $U \in [0,1]^{d\times n}$ | matrix of $n$, $\boldsymbol{u}$ vectors arranged columnwise |
| $Z \in \mathbb{R}^{d\times n}$ | matrix of $n$, $\boldsymbol{z}$ vectors arranged columnwise |
| $\boldsymbol{y}$ | $n$-dimensional vector such that $\boldsymbol{y}_i \in \{1, 2, \cdots, m\}$ |

# B. Proofs

## B.1. Theorem 4.1

**Proof**: Let $\boldsymbol{z} \in \mathbb{R}^d$ is drawn from a $m$-component Gaussian mixture distribution with parameters $\Theta^* = \{\boldsymbol{\mu}^{l*}, \Sigma^{l*}, \alpha^{l*}\}_{l=1}^m$. Define strictly increasing transformations i.e. $\boldsymbol{w}_r = a_r \boldsymbol{z}_r + b_r$, with $a_r \in \mathbb{R}^+$ and $b_r \in \mathbb{R}$. Then $\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots \boldsymbol{w}_d]$ has a Gaussian mixture distribution with parameters $\Theta = \{A\boldsymbol{\mu}^{l*} + \boldsymbol{b},\ A^T\Sigma^{l*}A,\ \alpha^{l*}\}_{l=1}^m$, where $A = \text{diag}([a_1, a_2, \cdots a_d])$ and $\mathbf{b} = [b_1, b_2, \cdots b_d]$.

The vector $\boldsymbol{u} \in [0,1]^d$, such that $\boldsymbol{u}_r = \Psi_r(\boldsymbol{z}_r; \Theta^{*r}); r = 1, 2, \cdots, d$, has the joint density function (see definition 2 of GMC distribution) given by equation (12).

$$\zeta(\boldsymbol{u}; \Theta^*) = \left( \frac{\psi\left(\boldsymbol{z}; \Theta^*\right)}{\prod_{r=1}^{d} \psi_r\left(z_r; \Theta^{r*}\right)} \right) \tag{12}$$

Likewise, the density function of $\boldsymbol{v} \in [0,1]^d$ such that $\boldsymbol{v}_r = \Psi_r(\boldsymbol{w}_r; \Theta^r); r = 1, 2, \cdots, d$ has the density function given by equation (13).

$$\zeta(\boldsymbol{v}; \Theta) = \left( \frac{\psi\left(\boldsymbol{w}; \Theta\right)}{\prod_{r=1}^{d} \psi_r\left(\boldsymbol{w}_r; \Theta^{r}\right)} \right) \tag{13}$$

However, since cumulative distribution function values remain invariant under strictly increasing transformations, we have $\boldsymbol{u} \equiv \boldsymbol{v}$. This means $\boldsymbol{u}$ and $\boldsymbol{v}$ have the same generative distribution, or equivalently $\zeta(\mathbf{u}; \Theta^*) = \zeta(\mathbf{u}; \Theta)$. Therefore, the corresponding likelihood functions, defined on a dataset with $n$ samples ($U \in [0,1]^{d\times n}$), are equal for the two parameter configurations $\Theta^*$ and $\Theta$ i.e. $\ell_\zeta(\Theta|U) = \ell_\zeta(\Theta^*|U)$. This completes the proof.

B.2. Theorem 4.2

**Proof**: The proof is straightforward in the light that the LHS expression in Equations (6) and (7) correspond to the mean and the variance of the marginals of a GMM, respectively. Affixing the vectors $\boldsymbol{g}$ and $\boldsymbol{h}$ to the pre-specified values, therefore, prohibits the transformations that cause non-identifiability (refer to Theorem 4.1). Intuitively, by fixing the margins of the GMM (as we only seek the dependence structure it encodes), we eliminate different parameter configurations that encode the same dependence structure. Conversely, any parameter update that abides by the constraints specified in Theorem 4.2, would result in non-increasing transformations, thus mitigating the non-identifiability noted in Theorem 4.1.

## C. Derivation of E-step

Let's denote the posterior distribution on the latent variables, given the current iteration $\hat{\Theta}$, as $P(\boldsymbol{y}|U,\hat{\Theta}) = \prod_{j=1}^{n} P\left(\boldsymbol{y}_j|U_{:j},\hat{\Theta}\right)$. The expectation of the complete data log-likelihood in Equation (10), with respect to $P(\boldsymbol{y}|U,\hat{\Theta})$, can be written as

$$Q(\Theta,\hat{\Theta}) = \sum_{\boldsymbol{y}^1=1}^{m}\sum_{\boldsymbol{y}^2=1}^{m}\dots\sum_{\boldsymbol{y}^n=1}^{m}\left[\left(\sum_{i=1}^{n} H_{i\boldsymbol{y}^i}\right)\prod_{j=1}^{n} G_{j\boldsymbol{y}^j}\right] \text{ where,} \tag{14}$$

$$H_{i\boldsymbol{y}^i} = \log\left(\frac{\alpha^{\boldsymbol{y}^i}\phi(Z_{:i};\Theta^{\boldsymbol{y}_i})}{\prod_{r=1}^{d}\psi_r\left(Z_{ri};\Theta^r\right)}\right), \quad \text{and} \quad G_{j\boldsymbol{y}^j} = P\left(\boldsymbol{y}^j|U_{:j},\hat{\Theta}\right)$$

The $(j,l)^{th}$ element of the matrix $G$ denotes the posterior probability of the $l^{th}$ component given the $j^{th}$ sample and the current parameter estimate $\hat{\Theta}$, and is computed as

$$G_{jl} = P\left(\boldsymbol{y}^j = l|U_{:j},\hat{\Theta}\right) = \frac{\alpha^l\phi\left(\Psi^{-1}(U_{:j});\hat{\Theta}^l\right)}{\sum_{i=1}^{m}\alpha^i\phi\left(\Psi^{-1}(U_{:j});\hat{\Theta}^i\right)}. \tag{15}$$

The expression in 14 can be expanded to obtain,

$$\begin{aligned} Q(\Theta,\hat{\Theta}) = &\sum_{\boldsymbol{y}^1=1}^{m} H_{1\boldsymbol{y}^1}G_{1\boldsymbol{y}^1}\prod_{j=1,j\neq 1}^{n}\sum_{\boldsymbol{y}^j=1}^{m} G_{j\boldsymbol{y}^j} \\ &+\sum_{^2=1}^{m} H_{2\boldsymbol{y}^2}G_{2\boldsymbol{y}^2)}\prod_{j=1,j\neq 2}^{n}\sum_{\boldsymbol{y}^j=1}^{m} G_{j\boldsymbol{y}^j}+\dots \\ &\dots+\sum_{\boldsymbol{y}^n=1}^{m} H_{n\boldsymbol{y}^n}G_{n\boldsymbol{y}^n}\prod_{j=1,j\neq n}^{n}\sum_{\boldsymbol{y}^j=1}^{n} G_{j\boldsymbol{y}^j} \end{aligned} \tag{16}$$

Given that $\sum_{\boldsymbol{y}^j=1}^{m} G_{j\boldsymbol{y}^j} = 1,$ for all $j \in \{1,\dots,n\}$, the above equation simplifies as.

$$Q(\Theta,\hat{\Theta}) = \sum_{i=1}^{n}\sum_{\boldsymbol{y}^i=1}^{m} H_{i\boldsymbol{y}^i}G_{i\boldsymbol{y}^i}. \tag{17}$$

Thereafter, it is easy to establish 1-to-1 correspondence between Equations (17) and (11) by expanding the term $H_{i\boldsymbol{y}^i}$.

## D. Partial derivatives of $\Psi_r^{-1}(\cdot)$

Let's say that $z_r = \Psi_r^{-1}(u)$. Even though $\Psi_r^{-1}(\cdot)$ does not have a closed-form, its partial derivatives can be obtained analytically via its forward function $\Psi_r(\cdot)$, and by invoking Euler's chain rule, as shown in Equation (18).

$$\frac{dz_r}{d\theta} = -\frac{\left(\frac{d\Psi_r(z_r)}{d\theta}\right)_z}{\left(\frac{d\Psi_r(z_r)}{dz_r}\right)_\theta} \tag{18}$$

The expression in the denominator is identical for all the partial derivatives and is simply the density functionof the univariate GMM, i.e

$$\frac{\partial\left(\Psi_r(z_r)\right)}{\partial z_r} = \psi_r(z_r). \tag{19}$$

The partial derivatives of the numerator w.r.t $\theta \in \{\alpha_k, \mu_k, \Sigma_k\}$ can be derived, as follows, by applying of matrix calculus identities.

Derivative of $z_r$ w.r.t to $\alpha_k$

$$\frac{\partial\left(\Psi_r(z_r)\right)}{\partial \alpha_k} = \frac{1}{2}\left[1 + \text{erf}\left(\frac{z_r - \mu_{k,r}}{\sqrt{2\Sigma_{r,k}}}\right)\right] \tag{20}$$

Derivative of $z_r$ w.r.t to $\mu_k$

$$\frac{\partial\left(\Psi_r(z_r)\right)}{\partial \mu_k} = -\frac{\alpha_k}{\sqrt{2\pi\Sigma_{r,k}}}\exp\left(-\frac{(z_r - \mu_{k,r})^2}{2\Sigma_{r,k}}\right) \tag{21}$$

Derivative of $z_r$ w.r.t to $\Sigma_k$

$$\frac{\partial\left(\Psi_r(z_r)\right)}{\partial \Sigma_k} = -\sum_{l=1}^{m}\frac{\alpha_l}{\sqrt{2\pi\Sigma_{r,l}}}\exp\left(-\frac{(z_r - \mu_{r,l})^2}{2\Sigma_{r,l}}\right) \times \frac{(z_r - \mu_{r,l})}{2\Sigma_{r,l}} \times \frac{\partial\left(\Sigma_{r,l}\right)}{\partial \Sigma_k} \tag{22}$$