Model-Based Clustering of High Dimensional, Mixed-type Data

Aaron O'Brien
StudentID: 2058015
Supervisor: Ioannis Kosmidis

September 14, 2021

Acknowledgements

I would like to acknowledge my supervisor, Ioannis Kosmidis, for the idea that this dissertation aims to explain and implement and the for the ongoing support provided, which extended far beyond the areas referenced within the text.

Abstract

Gaussian mixture models are one of the most popular tools for unsupervised learning. However, these models are limited in the data that they can process and the clusters which they can produce. In recent years, a literature has developed which uses copulas as components for finite mixture models. This approach allows for mixed-mode and mixed-domain data to be modelled. Additionally, it allows for more flexibility in the shapes of clusters that are produced. However, it is challenging to construct high-dimensional copulas and apply them to mixture models. The methods that have achieved this come with a high computational cost and somewhat challenging implementations. This dissertation outlines and implements a pairwise approach to fitting copula mixture models through an Expectation-Conditional Maximisation algorithm. The approach is highly flexible, in both the data it can process and the clusters it can produce. Furthermore, it is simple to extend to an arbitrary number of dimensions and is computationally more efficient than both other copula approaches as well as the more established Gaussian mixture model.

1. Introduction

Finite mixture models present an advancement over non-parametric clustering techniques, such as K-means or hierarchical clustering. Whilst non-parametric methods can be of use in some scenarios, these approaches to unsupervised learning suffer from key limitations. They are unable to provide any measure of confidence regarding the cluster allocation of each observation or the likelihood of the fitted clusters. They do not attempt to model the data-generating process, which may make them less helpful when being applied to a given situation and prevents the construction of simulated data from the fitted model. Additionally, non-parametric approaches will often rely on distance metrics to determine the clusters. This means that they are sensitive to the scale of the observations, which makes it invalid to apply them to certain types of data, such as binary data. Additionally, it typically means that variables must be standardised before fitting models.

Finite mixture models, and in particular the Gaussian mixture model, are able to address some of the limitations of non-parametric approaches. The models provide an approximation to the data generating process by assuming that the clusters are generated by drawing data from separate distributions, with some probability of each observation coming from one of the given distributions (Banfield and Raferty, 1993)[4]. Additionally, finite mixture models offer more flexibility in the shapes of clusters that can be produced. The clusters generated by a Gaussian mixture model are constrained to be ellipsoidal, but can vary in terms of shape, size or orientation, based on the specification of each cluster covariance matrix (Scrucca, 2016)[27]. Additionally flexibility has been added by using alternative distributions that can model a greater variety of dependence structures, see for instance

Andrews, McNicholas and Subedi (2011)[19] who work with multivariate t-distributions. Due to the advantages of finite mixture models, as well some of the attractive properties of the multivariate Gaussian and t-distributions, finite mixture models using these component densities have become some of the most popular unsupervised learning models (Scrucca, 2016)[27]. They have been applied to financial risk models (Haas, Mittnik and Paolella, 2007)[13], facial recognition (Yang and Ahuja, 1998)[8], medical diagnostics (Martis, Chakraborty and Ray, 2009)[16] and many other fields.

However, Gaussian and multivariate-t distributions are still limited as a choice for component densities. One weakness is that the resulting clusters are limited to being symmetrical. An answer to this is to work with skewed versions of both distributions, as in Fruhwirth-Schnatter and Pyne (2010)[18]. However, the skewed alternatives only partially address the lack of flexibility of the models. It allows for asymmetrical clusters to be generated but the clusters are still constricted. Another limitation is that all of these distributions assume that the features of the data are continuous with infinite domain. It is often the case that data-sets contain mixed-domain and mixed-mode data, i.e. data that is continuous but bounded or data that is discrete. For example, medical data will often contain body measurements which are strictly positive or inidicator variables for patient behaviour/attributes that will be binary. Hence, the most common choices for component density are non-applicable to a significant number of data-sets for which there may be interest in building an unsupervised learning model.

An additional concern when working with these models is that they can be computationally expensive when applied to high-dimensional data. Finite mixture models are solved using an EM algorithm and it is required to calculate the density under the assumed distribution and current parameters of each observation. The calculation of the observation densities can be costly to perform, e.g. the calculation of each density when assuming a multivariate Gaussian distribution require the inversion of the estimated covariance matrix.

As a result of the above limitations, it would be valuable to find choices for component densities that can improve upon the flexibility of the clusters that can be fit, the types of data that can be modelled and/or the computational cost of fitting higher dimensional models. The use of copulas as component densities is an approach that is able to address the first two concerns. To address the concern of transitioning into higher dimensions, a pairwise likelihood approach is used to model the density of each observation, or equivalently the likelihood of the model parameters.

The key limitation of the component density choices outlined above is that they are only able to accurately the data if the data is drawn from one of those distributions, or is well modelled as such, and the distributions themselves are not flexible enough to capture the broad variety of dependence structures that different data will present. Sklar (1959)[1] shows that every multivariate distribution can be represented as a copula with appropriately chosen marginals. This in itself addresses both the issue of inflexible clustering and the model being able to only handle continuous, unbounded data. The only arguments that copulas need to take is the cumulative distribution function of the marginal data. As such, as long as each feature within a data-set can be adequately modelled as coming from some known univariate distribution, a copula mixture model will be able to model the data-set. Given the wealth of univariate distributions available, this condition should be straightforward to satisfy. Then, given appropriately chosen marginal distributions, the copula can theoretically model any observed data. In reality, we are constrained by the current set of copulas whose cumulative distribution function, probability density function and partial derivative are known. However, this set of copulas is sufficient to provide significant advantages in terms of cluster flexibility, see e.g. Kosmidis and Karlis (2016)[26].

The issue of how the mixture model scales with dimensions is still outstanding. In some sense, this issue is more challenging to address when working with copulas as the construction of high-dimensional copulas is challenging (Nelsen, 2006)[14]. A body of literature is developing regarding the use of vine copulas, which are constructed through building nested-trees of bivariate copulas to model multivariate distributions. For example, see Kim et al. (2010)[23] or Sahin (2021)[32]. However, these models require a significant number of ad-hoc choices in construction and can be challenging and costly to implement. In this regard, multivariate Gaussian and t-distributions are far more straightforward to work with.

To bridge the gap between bivariate copulas, of which there is a diverse set of options, and multivariate copulas, which are relatively harder to construct, a pairwise likelihood approach is taken. Varin (2011)[20] provides a summary of composite likelihood methods, including the pairwise likelihood approach, and some applications. Rather than modelling each observation density using the true likelihood, the density is modelled as a product of the bivariate density function of each pairwise combination of features. This not only reduces the computational cost compared to the use of the true likelihood, but it circumvents the need to explicitly define a high-dimensional joint distribution of the data. Crucially, all of the component distributions used in this paper are closed under marginalisation. Instead, the model only needs to work with bivaraite copulas. Lindsay (1988)[2] shows that maximum likelihood estimates based on a composite likelihood is unbiased and asymptotically normal, as is the corresponding estimate using the true likelihood. There is a cost in terms of statistical efficiency as the asymptotic variance of the maximum likelihood estimate is greater when working with a composite likelihood compared to when working with the true likelihood.

The rest of the paper is organised as follows: section 2 gives the details of specifying a finite mixture model and further discusses some of the common choices of component densities. Section 3 defines copulas and discusses their use in mixture models as well as the properties of some copulas that are used later in the paper. Section 4 explains the details of the pairwise likelihood approach and the benefits of the approach for Gaussian and copula mixture models. There is also a brief discussion of the implication of a pairwise likelihood approach on the performance metrics that can be used for model comparison. Section 5 details the EM algorithms used to fit the mixture models considered. Section 6 consists of simulation studies that display the flexibility of copula mixture models. Section 7 applies the approach to real data-sets and compares the performance to other methods of unsupervised learning. Section 8 discusses the limitations of the work in this paper and provides ideas for advancement.

2. Finite Mixture Models

2.1 Model Specification

Bouveyron (2019)[28] provides the following characterisation of finite mixture models. Suppose a data set consists of n, d vectors $(x_1, \ldots, x_n)$ with $x_i = (x_{i1}, \ldots, x_{id})$. Then the probability density or mass function of each observation, x_i is approximated as:

$$P(x_i) = \sum_{k=1}^{K} \tau_k f_k(x_i|\theta_k) \tag{1}$$

where K is the number of mixing components, τ_k is the mixing weight for the k^{th} component, $f_k()$ is the k^{th} component density or mass function and θ_k is the parameter vector for that density or mass function. In this case, both K and the mixture density/mass functions are hyperparameters, i.e. the model assumes they are known and they must be optimised through model selection rather than internal model fitting. The aim is to estimate the parameters $\Psi = (\tau_1, .., \tau_k, \theta_1, ..., \theta_k)$. The model log-likelihood is found by calculating:

$$l(\Psi) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \tau_k f_k(x_i|\theta_k). \tag{2}$$

Direct maximisation of the model likelihood is not possible as it involves the log of a summation. Instead, latent indicator variables, z_{ik}, on each observation are assumed, such that $\sum_{k=1}^{K} z_{ik} = 1$. These indicator variables are 1 if x_i is from component k and 0 otherwise. Then, the model log-likelihood for a given component is found by calculating:

$$l(\Psi) = \sum_{i=1}^{N} z_{ik} \text{log} \tau_k f_k(x_i|\theta_k). \tag{3}$$

. From here, an EM algorithm can be used to estimate the parameter vector, $\Psi = (z_1, \ldots, z_n, \tau_1, \ldots, \tau_k, \theta_1, \ldots, \theta_k)$ (McLachlan and Peel, 2000)[9]. See section 5 for the mechanics of the EM algorithms used.

2.2 Common component choices

The most common choice for component densities is the Gaussian distribution. The Gaussian distribution has some attractive properties. Firstly, each step of the EM algorithm has a closed-form solution (Bilmes, 1998)[7]. As a result, numerical optimisation is not necessary, which can reduce the computational cost of model fitting. Additionally, the size, shape, orientation or any combination thereof of each cluster can be fixed. Fixing any combination of these factors is done by holding certain components of the eigenvalue decomposition of the covariance matrix constant across clusters (Celeux and Govaert, 1993)[5]. Gaussian mixture models (GMM) are also closed under marginalisation, meaning that any subset of features will also follow a GMM, with the appropriate subset of the parameter vector, Ψ.

GMMs are only able to produce elliptical clusters (Celeux and Govaert, 1993)[5], which means that they are unable to account for asymmetries within cluster data. Alternative choices for component densities that address this issue are skew-normal and skew-t distributions. Fruhwirth-Schnatter and Pyne (2010)[18] show that using such distributions as component densities allow the resulting clusters to better accomodate asymmetric dependence structures within clusters. Lee and McLachlan (2011)[24] show that models using any of these distributions also have closed-form solutions for each step of the EM algorithm and provide an efficient implementation.

However, models using any of the distributions mentioned so far in this section for component densities are limited by the types of marginal data that they continue. In particular, all of these models require that the marginal distribution of each feature is continuous with infite domain. One approach to address the constraints on data that can be handled, at least when considering bounded-domain, continuous variables is to first transform the data to having an infinite domain and then fit a mixture model using Gaussian component densities. However, two issues exist. Firstly, Dean and Nugent (2013)[22] show that such an approach leads to lower accuracy than working with the non-transformed, bounded-domain data. Secondly, Kosmodis and Karlis (2016)[26] show that the results vary with the choice of transformation. Hence, working with the non-transformed data removes the ad-hoc choice of transformation that can affect the results. Additionally, there is no transformation of feature data that has a discrete domain which can make the data continuous, and so mixture models based on the distributions mentioned so far cannot be used to model any data that contains discrete features.

It is reasonable to search for choices of component densities that can address the limitations raised above: inability to handle more complex dependence structures and inability to handle bounded or mixed-domain feature data.

3. Copulas

3.1 Definition and Properties of Copulas

From Nelsen (2006)[14], a 2-dimensional Copula, $C(u, v)$, is formally defined as any function which maps from the 2-dimensional unit hypercube, $[0,1]^2$, to the 1-dimensional unit interval, $[0,1]$, and satisfies the following two properties:

1. For all u,v in $[0,1]$:
$$C(u,0) = C(0,v) = 0 \tag{4}$$
and
$$C(u,1) = u \text{ and } C(1,v) = v \tag{5}$$

2. For every u1,u2,v1 and v2 in $[0,1]$ such that $u1 \leq u2$ and $v1 \leq v2$:
$$C(u2,v2) - C(u1,v2) - C(u2,v1) + C(u1,v1) \geq 0. \tag{6}$$

An implication of these two properties is that a Copula is non-decreasing in either of its arguments, u or v. One implication of the above definition is the following: suppose X and Y are random variables with distribution functions $F(x) = P(X \leq x$ and $G(y) = P(Y \leq y$ respectively and joint distribution function $H(x,y) = P(X \leq x, Y \leq y)$. A function that maps $(F(X), G(Y))$ in $[0,1]^2$ to $H(X,Y)$ in $[0,1]$ is a copula, as it satisfies the two conditions above (Nelsen, 2003)[12].

The importance of being able to model a 2-dimensional, joint distribution function as a copula is that this is possible for any joint distribution function (Sklar, 1959)[1]:

Sklar's Theorem: *Let H be a 2-dimensional distribution function, with marginals $F(X)$ and $G(Y)$. Then there exists some copula, C, such that $H(x,y) = C(F(x), G(y))$. Conversely, for any distribution functions $F(X)$ and $G(Y)$ and copula C, the function H as defined above is a 2-dimensional distribution function with marginal $F(X)$ and $G(Y)$. Furthermore, if $F(X)$ and $G(Y)$ are continuous, C is unique.*

As such, copulas are a highly flexible tool that are able to exactly estimate any 2-d dimensional distribution function. Unlike GMMs, which are only able to fit elliptical clusters, a copula mixture model will be able to accurately model any cluster data, assuming appropriately chosen marginals and copulas. Additionally, copulas do not have any restrictions on the support of each feature, nor are the features required to be continuous. This property is desirable given that data sets involved mixed-domain and mixed-mode data. Additionally, certain copulas are closed under marginalisation, which is a crucial property to allowing mixture models to be used for high dimensional data whilst using a pairwise likelihood approach, as in this paper. In particular, all Archimedean copulas as well as the Normal copula are closed under marginalisation (Kosmidis and Karlis, 2016)[26]. Sections 4 and 5 provide details as to how this property is used to fit mixture models.

In order fit copula mixture models, it is necessary to work with the bivariate density of any combination of feature data. In particular, suppose a data set consists of n, d vectors $(x_1, \ldots, x_n)$ with $x_i = (x_{i1}, \ldots, x_{id})$. To calculate the likelihood given the marginal distributions, copula and all corresponding parameters, it is necessary to compute $h(x_i, x_j)$

for all $i \neq j$. The structure of the bivariate density will depend on whether x_i, x_j follow continuous or discrete marginals. There are three possible combinations: both marginal distributions are continuous, both are discrete or one is continuous and the other is discrete.

In the continuous-continuous case, partial differentiation with respect to both x_i, x_j leads to:

$$h(x_i, x_j) = c(F_i(x_i), F_j(x_j))f_i(x_i)f_j(x_j) \tag{7}$$

where $c(u, v)$ is the copula density function and F_i, F_j, f_i, f_j are the distribution and density functions of x_i and x_j respectively. Applying the probability mass function in Panagiotelis et al. (2012)[21] to the 2-dimensional case gives:

$$\begin{aligned} h(x_i, x_j) = P(X_i = x_i, X_j = x_j) = C(F_i(x_i), F_j(x_j)) - \\ C(F_i(x_i - 1), F_j(x_j)) - C(F_i(x_i), F_j(x_j - 1)) + C(F_i(x_i - 1), F_j(x_j - 1)) \end{aligned} \tag{8}$$

when both x_i, x_j are discrete. It is important to note that (6) assumes that 1 is the smallest increment supported by the marginal distribution. If this is not the case, (6) can be appropriately adjusted. In case where X_i is continuous and X_j is discrete, $h(x_i, x_j)$ is calculated by first considering a finite difference in X_j and then partially differentiating with respect to X_i:

$$h(x_i, x_j) = \frac{d}{dX_i}[C(F_i(x_i), F_j(x_j)) - C(F_i(x_i), F_j(x_j - 1))] \tag{9}$$

. (7) requires computation of the partial derivative of the copula with respect to X_i. The partial derivative can be identified by applying the chain rule to the copula distribution function and the results of this for each of the Archimedean copulas used in this paper are given in section 3.3. It is also worth nothing that each copula used in this paper are symmetric with respect to their arguments, by construction, i.e. $C(u, v) = C(v, u)$ for all $u, v \in [0, 1]$. The same holds for the copula density functions. It is possible to extend each of these density functions into higher dimensions. However, given that the models used in this paper are fit using pairwise likelihood, (5)-(7) are sufficient to calculate the model log-likelihood.

3.2 Copula Mixture Models

As defined in (1), a mixture model approximates the probability of a given observation as a weighted sum of probabilities of the observation from K distinct mixture densities. A copula mixture model assumes that the joint distribution of each feature, within each mixing component, is modelled by a copula. As such, the component density can be calculated as outlined in section 3.1. Unfortunately, unlike when using normal, skew-normal, student-t or skewed student-t distributions for the mixing components, there is no closed form solution for any of the copulas that are commonly used as mixing components. As such, the m-steps in the EM algorithm for the marginal and copula parameters require numerical optimisation, unlike when using a GMM. An additional concern is that it is non-trivial to construct high-dimensional copulas and very few seamlessly extend to high-dimensions (Nelsen, 2003)[12]. One that does is the Normal copula. Vine copulas can be used to precisely estimate any distribution of any dimensions, (e.g. Kim et al. (2010)[23] or Sahin (2021)[32]). However, they can require somewhat ad-hoc decisions regarding dependence relations between features and are very computationally expensive to both fit and do model selection for. However, the use of pairwise likelihood means that there is no need to construct high-dimensional copulas when modelling high-dimensional data. As earlier mentioned, copula mixture models are able to accurately cluster far more types of data than GMMs. The first reason for this is that they are able to support mixed-mode and mixed-domain data. As the majority of data-sets will involve either mixed-mode, mixed-domain or both types of data, this is a significant advance over GMMs. Additionally, they are able to account for any dependence structure assuming an appropriately chosen copula. In practice, the chosen copula will not perfectly match the true data-generating process, in the same way that GMMs are accepted as an approximation. However, due to the greater flexibility displayed by the clusters fit by copula models, it is expected that copula models will have a higher accuracy than GMMs. As such, copula models are applicable to more data-sets and likely to have a better performance than GMMs.

3.3 Archimedean and Gaussian Copulas

Unlike vine copulas which require a specification of $d(d+1)/2$ separate bivariate copulas for each component distribution(Bedford and Cooke, 2001)[10], the approach used in this paper only requires the specification of one bivariate copula per component. The distribution function of each pairwise combination of features, $H(X_i, X_j)$, is assumed to be equal to this type of copula, given appropriately chosen marginals. By constraining the choice of copulas to those that are closed under marginalisation, the user implicitly defines the joint distribution of the overall feature vector, $x_i = (x_{i1}, \dots, x_{id})$. This is because if a set of variables follow a joint distribution that is closed under marginalisation, any subset of those variables follows the same distribution, with the appropriately chosen subset of parameters. To this end, four copulas are considered in this paper, each of which is closed under marginalisation. The Frank, Clayton and Gumbel copulas are used, each of which is an Archimedean copula, as well as the Normal copula, which is also closed under marginalisation. By construction the Normal copula extends straightforwardly into higher dimensions, while each of the Archimedean copulas can extend into higher dimensional quasi-copulas (Nelsen, 2003)[12]. Hence, the use of these copulas within the pairwise likelihood framework leads to an implicit assumption on the joint distribution of the full observation vector.

Figure 1 shows data drawn from one of each of the Archimedean copulas. In each case, the marginal distribution of each variable was a normal distribution with parameters 10 and 2, i.e. $X_i \sim \mathcal{N}(10, 2)$, for $i = 1, 2$. The definition, density and partial derivative of each copula are below. In each case, the definition of the copula is from Nelsen (2006)[14], while the copula density and partial derivatives are sourced from the documentation of the python package *Copulas* v0.5.0 (https://sdv.dev/Copulas/api/copulas.html). Throughout, it is assumed that $u, v \in [0, 1]$.

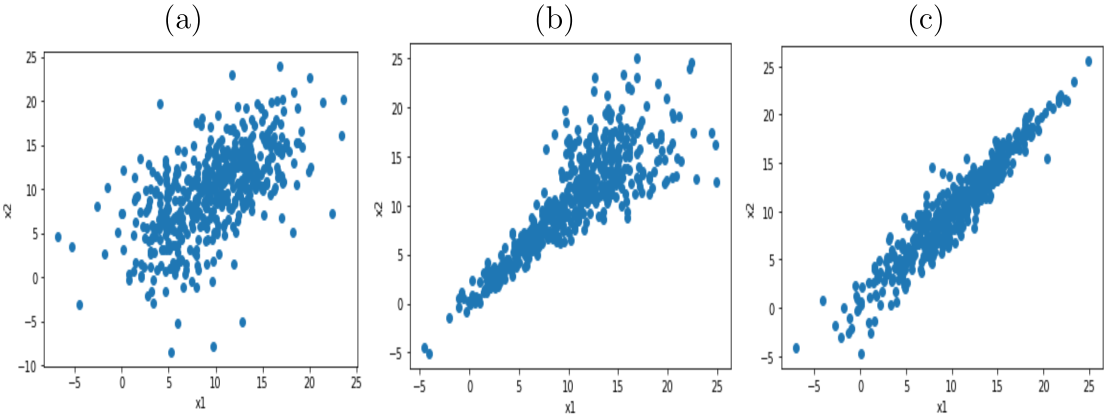

Figure 1: *Artificial data simulated from a composed distribution. In each case, the marginal data is drawn from a normal distribution, with parameters* $\mu = 10$ *and* $\sigma = 2$. *In (a),the joint distribution is a Frank copula, for (b) it is a Clayton copula and in (c) a Gumbel copula. In each case, the copula parameter is* $\theta = 5$.

The Frank copula with parameter $\theta \in (-\infty, +\infty) \setminus (0)$, shown in figure (1)(a), is a symmetric copula given by:

$$C(u, v) = \frac{-1}{\theta} \ln(\frac{1 + (e^{-\theta * u} - 1)(e^{-\theta * v} - 1)}{e^{-\theta} - 1}) \tag{10}$$

The corresponding density function follows by differentiating with respect with to both u and v:

$$c(u, v) = \frac{-\theta(e^{-\theta} - 1)(e^{-\theta*(u+v)})}{((e^{-\theta * u} - 1) * (e^{-\theta * v} - 1) + e^{-\theta} - 1)^2} \tag{11}$$

Both the density and the distribution functions are symmetric with respect to their arguments, i.e. $C(u, v) = C(v, u) andc(u, v) = c(v, u)$. This property is shared by the Clayton, Gumbel and Normal copula. However this is not true for the partial derivative of the copula. In the case of the Frank copula, the partial derivative is:

$$\frac{d}{du} C(u, v) = \frac{(e^{-\theta * u} - 1)(e^{-\theta * v} - 1) + (e^{-\theta * v} - 1)}{(e^{-\theta * u} - 1)(e^{-\theta * v} - 1) + (e^{-\theta} - 1)} \tag{12}$$

The Clayton copula, with parameter $\theta \in (-1, +\infty) \setminus (0)$, shown in figure (1)(b), is an asymmetric distribution function. The function displays greater dependence in the left tail, as shown in figure (1)(b) by the tighter grouping of data in the left tail, compared to the relatively sparse grouping in the right tail. The definition of the copula, copula density and partial derivative are in order below:

- $$C(u,v) = \max([u^{-\theta} + v^{-\theta} - 1]^{\frac{-1}{\theta}}, 0) \tag{13}$$

- $$c(u,v) = (\theta+1)(uv)^{-\theta-1}(u^{-\theta} + v^{-\theta} - 1)^{-}\frac{-2\theta+1}{\theta}) \tag{14}$$

- $$\frac{d}{du}C(u,v) = u^{-\theta-1}(u^{-\theta} + v^{-\theta} - 1)^{\frac{-\theta+1}{\theta}} \tag{15}$$

The Gumbel copula also takes a single parameter, $\theta \in (1, +\infty)$. Like the Clayton copula, it is asymmetric, however it shows greater dependence in the right tail. This can be seen in figure (1)(c), where the data is grouped tighter in the right tail. Again, the definition of the copula, copula density and partial derivative are in order below:

- $$C(u,v) = e^{-((-\ln(u)^{\theta})+(-\ln(v)^{\theta}))^{\frac{1}{\theta}}} \tag{16}$$

- $$c(u,v) = \frac{C(u,v)}{u,v}\frac{((-\ln(u)^{\theta}) + (-\ln(v)^{\theta}))^{\frac{2}{\theta}} - 2}{(\ln(u)\ln(v))^{1-\theta}}(1+(\theta-1)((-\ln(u)^{\theta})+(-\ln(v)^{\theta}))^{\frac{-1}{\theta}}) \tag{17}$$

- $$\frac{d}{du}C(u,v) = C(u,v)\frac{(-\ln(u)^{\theta}) + (-\ln(v)^{\theta}))^{\frac{1}{\theta}-1}}{\theta(-\ln(u)^{1-\theta}} \tag{18}$$

Following the notation of Nelsen (2003)[12], let $N_\rho(x,y)$ represent the standard bivariate normal distribution function with correlation coefficient between x and y of ρ. Then the 2-dimensional Normal copula is given by:

$$C(u,v) = N_\rho(\Phi^{-1}(u), \Phi^{-1}(v)) \tag{19}$$

where Φ is the univariate, standard normal distribution function. Whilst closed-form expressions for the univariate and bivariate standard normal distribution functions do exist, see Drezner and Wesolowsky (1990)[3], there is no closed form expression for Φ^{-1}. As a result, there is no closed-form expression for the copula or its density function. However, both of these can be approximately evaluated (Nelsen, 2003)[12].

4. Pairwise Likelihood

4.1 Definition and Properties of Pairwise Likelihood Functions

Throughout this paper, model fitting is done through maximisation of the pairwise likelihood, rather than the true likelihood. The pairwise likelihood is a type of composite likelihood and so has the properties of a composite likelihood function. In order to form the pairwise likelihood version of (4), we must alter our assumptions slightly. Instead of assuming latent component indicator variables on observation, z_{ik}, we assume these indicators on each pairwise combination of features for each observation; Formally, we assume that there exists, for each (m, j), such that $m \neq j$, z_{ik}^{mj}, where $\sum_{k=1}^{K} z_{ik}^{mj} = 1$ and $P(z_{ik}^{mj} = 1) = \tau_k$ for all k. When using the pairwise likelihood, $\Psi = (z_{1,1}^{1,2}, \ldots, z_{n,K}^{p-1,p}, \tau_1, \ldots, \tau_K, \theta_1, \ldots, \theta_K)$. With this alteration, the pairwise likelihood can be written as:

$$l_{pair}(\Psi) = \sum_{1}^{K} \sum_{i=1}^{N} \sum_{m<j} z_{ik}^{mj} \log(\tau_k f_{2,k}(x_i^{mj} | \theta_k^{mj})) \tag{20}$$

where $f_{2,k}$ is the bivariate density function assumed for cluster k- this function is the same for each pairwise combination of features- and θ_k^{mj} is the corresponding subset of θ given the pair of features $(m, j), m \neq j$.

As with maximum likelihood estimation, the estimates produced using pairwise and other composite likelihood approaches are unbiased. However, they are less efficient. In particular, the asymptotic distribution of the maximum pairwise likelihood estimate will also follow a normal distribution centred about the true parameter values. However, the variance will be the inverse of the Godambe information matrix, rather than the inverse of the Fisher information matrix (Varin, 2011)[20]. As such, the relative statistical efficiency of the pairwise likelihood estimators will be determined by the ratio between the Fisher information and the Godambe information. In certain circumstances, the two information matrices will be equal and so the statistical efficiency of both approaches is the same, see Mardia et al. (2009)[15]. However, this is generally not true and use of composite likelihood comes at the cost of statistical efficiency.

Whilst there are many benefits of using pairwise likelihood, there are two key factors motivating its use in this paper. The first is, as mentioned in section 3.2, it is challenging to construct higher dimensional copulas and often the corresponding distribution functions are difficult to work with. When using a pairwise likelihood approach, it is not necessary to fully specify a high dimensional, joint distribution. Additionally, it is possible to choose copulas as component distributions that are straightforward to work with. The second key benefit of using pairwise likelihood is that the computational cost will scale better with dimensions. When fitting a GMM model using the true likelihood, the cost is $\mathcal{O}(p^3)$. The computation cost is reduced to $\mathcal{O}(p^2)$ when fitting a finite mixture model with any form of component density, including Gaussian as well as all of the copulas used in this paper. See section 5 for details. It is worth noting that one could also assume working independence and therefore use the independence likelihood, as defined in Varin (2011)[20]. Indeed, this would further lower the computation cost and simplify the model- the only distributions that would need to be specified would be the marginal of each parameter. However, this approach does not allow any consideration of dependence between features and so would miss a significant part of the presumed data generation process. Hence, it would be expected that the costs of such an approach would outweigh the benefits.

This means that the choice to use a pairwise likelihood approach concerns a trade off between statistical and computational efficiency, especially in the case of fitting GMMs. A willingness to use a pairwise likelihood approach means that the cost of losing some statistical efficiency is outweighed by the benefit of gaining computation efficiency. One potential caveat to this is that the loss surface of a pairwise likelihood function is often smoother (Liang and Yu, 2003)[11]. As a result, it is possible that the mixture model is more likely to converge to the global optimum. However, in general the consideration to make when using pairwise likelihood is the one mentioned above. In this case, the aim of the paper is to produce a model that seamlessly transitions into higher dimensions and does so with lower computational cost than traditional GMMs. As such, the use of pairwise likelihood is justified.

4.2 Implication for Performance Metrics

One of the implications of using a pairwise likelihood approach in this case is that the number of free parameters being estimated will increase. Rather than estimating $n(k-1)$ indicator variables, z_{ik}, the model is required to estimate $n(k-1)(p)(p-1)/2$ indicator variables, z_{ik}^{mj}. In each case, it is $k-1$ as the k^{th} indicator for the that observation (and pairwise combination of features) is fully determined by the preceding $k-1$ as they must sum to 1. Traditionally, comparison between models that rely on a different number of free parameters would be done using information criterion. However, information criteria generally rely on the fitted model's log-likelihood, e.g. AIC or BIC. It is invalid to make inferences comparing the true and the pairwise likelihood, due to the difference in how they are calculated. Additionally, as the higher dimensional distributions of the copula mixture models fit in this paper are not explicitly specified, it is not possible to compare a copula model fit using the true likelihood to that fit using the pairwise likelihood. As such, where comparison between models is needed, accuracy is used.

5. Model Fitting

In order to fit mixture models, an EM algorithm will be used. The E-step in each case involves estimating the indicator variables, z_{ik} or z_{ik}^{mj} when using pairwise likelihood. The M-steps then estimate the remaining parameters in Ψ so as to maximise the model likelihood. For each of the three finite mixture models considered in this paper, the E-step has a closed form solution. Additionally, the two M-steps of the EM algorithm used for GMMs using the true likelihood have a closed form solution. This is not true for either GMMs or copula mixture models fit using a pairwise likelihood approach. For both of these models, an expectation-conditional maximisation algorithm is used, following supplementary material from Kosmidis (2016)[26]. For each of the CM-steps in both of these algorithms, there is no closed form solution. Parameter estimates are found using numerical optimisation. Section 5.1 gives a general outline of EM algorithms. Sections 5.2-5.4 gives details of the algorithms used to fit the three types of finite mixture model considered in this paper and considers the computational cost of each algorithm. Section 5.5. provides initialisation strategies used.

5.1 General EM Algorithm

The following text gives an abstract description of the EM algorithm, following the work from Bilmes (1998)[7]. Suppose an $n \times p$ data set, X, is observed and supposed to have been generated by some distribution, with parameters Θ. If the maximum likelihood estimate for Θ is analytically intractable based on X, an EM algorithm can be one approach to estimate Θ. An assumption is made that X is an incomplete data set, and so there exist some unobserved variable(s), Z. Additionally, it is assumed that the joint distribution of X and Z is:

$$p(x, z|\Theta) = p(z|\Theta, x)p(x|\Theta) \tag{21}$$

Applying this to finite mixture models, the assumption is that the indicator variables are fully determined by the mixing component densities and the mixing weights for each component. To solve for Θ, define:

$$Q(\Theta, \Theta^{i-1}) = E[\log(p(X, Z|\Theta))|X, \Theta^{i-1}] \tag{22}$$

where Θ^{i-1} are the current parameter estimates, given each iteration of the algorithm so far. In (22), the only unknown is Z- X is the observed data and Θ^{i-1} is used as the current estimate for Θ. Then, the evaluation of (22) is the E-step. The M-step consists of maximising the expectation found in the E-step, i.e.:

$$\Theta^{i} = \text{argmax}_{\Theta} Q(\Theta, \Theta^{i-1}) \tag{23}$$

The algorithm then iterates between each step until convergence. Typically, the algorithm is said to have converged when the improvement in log-likelihood for the most recent iteration is below some threshold. In this paper, this threshold is set at $1e^{-8}$. If it is not possible to complete the M-step exactly, instead some Θ is found which increases the completed-data log-likelihood. In this paper, numerical optimisation with respect to each parameter in Ψ is used to achieve an improved estimate for Θ in each iteration. Each iteration is guaranteed to lead to an increased log-likelihood regardless of if estimates are found analytically or numerically.

5.2 EM Algorithm: GMM Using True Likelihood

As explained is section 2.1, when working with the true likelihood, a finite mixture model, such as the GMM, assumes latent indicator variables for each observation, each cluster, z_{ik}. These variables indicate which mixing component each observation belongs to. With this augmentation in mind, the likelihood of the data being explained by any given mixing component, k, is as given in (2).

The E-step updates the estimates of z_{ik} as follows:

$$z_{ik} = \frac{\tau_k \mathcal{N}(x_i|\mu_k, \Sigma k)}{\sum_{j=1}^{K} \tau_k N(x_i|\mu_j, \Sigma j)} \tag{24}$$

where, $\mathcal{N}()$ is the multivariate normal density function of appropriate dimensions, μ_k is the mean vector and Σ_k is covariance matrix for mixing component k. Then, the M-step requires the re-estimation of the remaining parameters; namely, the mixing weights, τ_k and the parameters of the component distributions, μ_k and Σ_k. Each can be found analytically as follows:

- $$\tau_k = \frac{\sum_{i=1}^{N} z_{ik}}{N} \tag{25}$$
- $$\mu_k = \frac{\sum_{i=1}^{N} z_{ik} x_i}{\sum_{i=1}^{N} z_{ik}} \tag{26}$$
- $$\Sigma_k = \frac{\sum_{i=1}^{N} z_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} z_{ik}} \tag{27}$$

an intuitive interpretation is that the mixing weights are set as the proportion of observations predicted to belong to each component while the component parameters are a weighted version of the maximum likelihood estimates for a multivariate normal distribution, where the weights are determined by the estimated probability of observation i belonging to component k.

5.3 EM Algorithm: GMM Using Pairwise Likelihood

When using a pairwise likelihood approach to fit the GMM, the overall model log-likelihood can be written as:

$$l_{pair}(\Psi) = \sum_{1}^{K} \sum_{i=1}^{N} \sum_{m<j} z_{ik}^{mj} \log(\tau_k \mathcal{N}(x_i^{mj}|\mu_k^{mj}, \Sigma_k^{mj})) \tag{28}$$

where $\mathcal{N}()$ is the bivariate normal density function, $\mu_k^{mj} = [\mu_k^m, \mu_k^j]$ and $\Sigma_k^{mj} = \begin{pmatrix} \sigma_k^{mm} & \sigma_k^{mj} \\ \sigma_k^{jm} & \sigma_k^{jj} \end{pmatrix}$ There is no analytical approach that can solve the maximisation of (28). Simultaneous numerical optimisation of each parameter is also challenging. This is because the parameter constraints are at times tricky and vary between parameters: the τ_k's must each be less than one and sum to one, the bivariate normal means are unconstrained, the variances are constrained to be non-negative and the covariances are constrained to $[-1, 1]$. As there is neither a closed-form solution for the M-step nor an easy to implement numerical optimiser, a conditional approach to conducting the M-step is used, following the supplementary material of Kosmidis (2016).

Expectation/Conditional Maximisation (ECM) algorithms are introduced by Meng and Rubin (1993)[6], to be used in instances when the M-step would be otherwise tricky to conduct. Additionally, they show that ECM algorithms share the converge properties of the general EM algorithm. Importantly, it shares the property of always increasing likelihood with respect to iterations, meaning that the same convergence criterion can be applied. As mentioned earlier, the difference in the method of calculating the log-likelihood makes the use of information criteria invalid when comparing models fit using the true likelihood and the pairwise likelihood. The use of pairwise likelihood will lead to inflated values for the model log-likelihood. One concern could be that as the dimensions grow, this divergence is increases. This could lead to the tolerance of $1e_{-8}$ being too strict and leading to an excessive number of iterations, at least when compared to models fit using the true likelihood with the same tolerance. As such, it would be reasonable to reconsider the tolerance used to determine convergence, potentially raising it if the fit time was too high. This concern is accentuated by the fact that models with more parameters will take longer for each iteration, as the cost is $\mathcal{O}(p^3)$. Hence, setting an inappropriately strict tolerance will cause an unnecessarily high number of relatively expensive iterations. With that said, the model performance and run-time remained reasonable with a tolerance of $1e^{-8}$ for most data-sets fit in this paper and so that tolerance was primarily used, except where indicated.

The E-step is different from that of the true GMM in that it requires the estimation of pairwise, latent indiactor variables z_{ik}^{mj} as follows:

$$z_{ik}^{mj} = \frac{\tau_k \mathcal{N}(x_i^{mj} | \mu_k^{mj}, \Sigma_k^{mj})}{\sum_{j=1}^{K} \tau_k \mathcal{N}(x_i^{mj} | \mu_k^{mj}, \Sigma_k^{mj})} \tag{29}$$

The first M-step estimates the τ_k's has a closed-form solution as follows:

$$\tau_k = 2 \frac{\sum_{i=1}^{N} \sum_{m<j} z_{ik}^{mj}}{Np(p-1)} \tag{30}$$

where p is the number of features. With the τ_k's optimised, it remains to optimise the mean vector and covariance matrix of each component. (28) can be broken down into K separate terms that only depend on the component specific variables. Furthermore, the τ_k's can be taken outside each sum over N for each component and then ignored, as it does not affect maximum likelihood estimate. Then, each component mean vector and covariance matrix can be found by maximising the K component specific likelihoods:

$$Q(\mu_k^1,\dots,\mu_k^p,\sigma_k^{11},\dots,\sigma_k^{pp},\sigma_k^{12},\dots,\sigma_k^{p(p-1)}) = \sum_{i=1}^{N}\sum_{m<j} z_{ik}^{mj}\log(\mathcal{N}(x_i^{mj}|\mu_k^{mj},\Sigma_k^{mj})) \tag{31}$$

where $\mathcal{N}$ is the bivariate normal density function. Whilst there is still no closed form solution to the conditional maximisation of Q with respect to any of the feature means, variances or pairwise covariances, each can be computed numerically with the appropriate constraints in place. The mean vector of each component consists of p terms. The covariance matrix consists of p variance terms and $\frac{p(p-1)}{2}$ covariance terms. This means that a total of $\frac{Kp(p+3)}{2}$ terms must be optimised numerically. Each optimisation requires only a partial calculation of Q, as only some of the terms in Q depend on each parameter. Specifically, both the variance and mean of each feature, μ_k^p and σ_k^{pp} appear in $N(p-1)$ terms in Q. Each covariance term, σ_k^{mj} appears in only N terms. Hence, the models are fit by optimising the restricted Q which contain the relevant terms in order to minimise the cost. Due to the use of pairwise likelihood, only 2×2 matrices must be inverted to calculate the value of the normal density function. This means that the computation cost of each iteration of the ECM algorithm is $\mathcal{O}(p^2)$ compared to $\mathcal{O}(p^3)$ when working with the true likelihood. Given formally, the CM-steps of the ECM algorithm are:

- CM-step 1: for each $j \in [1,\dots,p]$:

$$\mu_k^{j*} = \operatorname*{argmax}_{m\in\mathcal{R}} Q(m,\dots\ \mu_k^p,\sigma_k^{11},\dots,\sigma_k^{pp},\sigma_k^{12},\dots,\sigma_k^{p(p-1)}) \tag{32}$$

- CM-step 2: for each $j \in [1,\dots,p]$:

$$\sigma_k^{jj*} = \operatorname*{argmax}_{s\in(0,\infty)} Q(\mu_k^{1*},\dots,\mu_k^{p*},s,\dots,\sigma_k^{pp},\sigma_k^{12},\dots,\sigma_k^{p(p-1)}) \tag{33}$$

- CM-step 3: for each $j \in [1,\dots,p-1]$ and each $m \in [j,\dots,p]$:

$$\sigma_k^{mj*} = \operatorname*{argmax}_{r\in(-1,1)} Q(\mu_k^{1*},\dots,\mu_k^{p*},\sigma_k^{11*},\dots,\sigma_k^{pp**},r,\dots,\sigma_k^{p(p-1)}) \tag{34}$$

This approach ensures an improvement in the log-likelihood at each iteration and can be straightforwardly implemented using a numerical optimiser.

5.4 EM Algorithm: Copula Mixture Model with Pairwise Likelihood

For the copula mixture models considered in this paper, there is no true likelihood, as they have been constructed using a pairwise likelihood approach and the p-dimensional distribution is implied, not explicitly stated. As such, the following is the only approach considered.

The joint distribution of each feature, given the mixing component, is defined as: $H_k(X_m, X_j) = C_k(F_m(X_m), F_j(X_j))$, where $C_k()$ is the copula used to model the joint distribution of each feature for component k, and F_m, F_j are the marginal distributions of X_m, X_j respectively. It is important to note that this assumes that each pairwise combination of distinct features within a given mixing component is assumed to be from the same family of copula distributions. Furthermore, both the copula and the marginal parameters are assumed to be known, tuning these hyperparameters must occur through model selection techniques. The only distinction between combinations is the copula parameter which is allowed to vary between combinations of features. This is denoted $\Theta_k^{m,j}$. In this paper, each copula considered takes a single value, and so the copula parameters are instead written as $\theta_k^{m,j}$. Defining the parameter(s) of each marginal distribution, F_k^m as Φ_k^m, then the full parameter vector for the pairwise copula mixture model can be written as $\Psi = (z_{1,1}^{1,2}, \ldots, z_{n,K}^{p-1,p}, \tau_1, \ldots, \tau_K, \phi_1^1, \ldots, \phi_K^p, \theta_1^{1,2}, \ldots, \theta_K^{p-1,p})$. The pairwise likelihood of the pairwise copula mixture model is then given by:

$$l_{pair}(\Psi) = \sum_{1}^{K} \sum_{i=1}^{N} \sum_{m<j} z_{ik}^{mj} \log(\tau_k h_k(x_i^m, x_i^j)) \tag{35}$$

The exact form of $h_k(x_i^m, x_i^j)$ will depend on whether x^m and x^j are continuous or discrete. If both are continuous, the density will be defined as in (7). If both are discrete, the density will be defined as in (8). If one is continuous and the other is discrete, the density will be defined as in (9). Again, there is no closed form solution for the M-step and numerical optimisation is will be challenging. As shown in section (3.3), the parameter of each copula used in this paper will have a constraint that must be satisfied. Additionally, the parameters of each marginal distribution will generally have some kind of constraint. Thus, an ECM algorithm is used.

The ECM algorithm used is relatively similar to that used to fit the pairwise GMM. The E-step is similar, with the difference being how the density of the observation is calculated:

$$z_{ik}^{mj} = \frac{\tau_k h_k(x_i^m, x_i^j)}{\sum_{j=1}^{K} \tau_k h_k(x_i^m, x_i^j)} \tag{36}$$

The first M-step is identical:

$$\tau_k = 2\frac{\sum_{i=1}^{N} \sum_{m<j} z_{ik}^{mj}}{Np(p-1)} \tag{37}$$

A similar approach is used to form the CM-steps. The marginal and copula parameters of each mixing component can be maximised separately and the τ_k's can taken outside the summation so that the objective function considered is:

$$Q(\phi_k^1, \ldots, \phi_k^p, \theta_k^{1,2}, \ldots, \theta_k^{p-1,p}) = \sum_{i=1}^{N} \sum_{m<j} z_{ik}^{mj} \log(h_k(x_i^m, x_i^j | \phi_k^m, \phi_k^j, \theta_k^{mj})) \tag{38}$$

From here, CM-step 1 optimises the marginal parameters while CM-step 2 optimises the copula parameters:

- CM-step 1: for each $j \in [1, \ldots, p]$:

$$\phi_k^{j*} = \operatorname*{argmax}_m Q(m, \ldots, \phi_k^p, \theta_k^{1,2}, \ldots, \theta_k^{p-1,p}) \tag{39}$$

- CM-step 2: for each $m \in [1, \ldots, p-1]$ and each $j \in [j, \ldots, p]$:

$$\theta_k^{mj*} = \operatorname*{argmax}_s Q(\phi_k^{1*}, \ldots, \phi_k^{p*}, s, \ldots, \theta_k^{p-1,p}) \tag{40}$$

Again, the computational cost can be decreased noticeably by omitting irrelevant terms from each calculation of Q, depending on which term is currently being optimised. Whenever the marginal parameter(s) of a feature is being optimised, Q will only depend on $N(p-1)$ terms which include the current feature. This reduces to N terms when optimising the pairwise copula parameter, θ_k^{mj}. Again, the cost of each optimisation is constant with respect to the dimensionality of the data, due to the use of pairwise likelihood. The exact number of parameters needing to be optimised will depend on the marginal distributions, e.g. if each marginal distribution is the Normal distribution $2p$ marginals parameters need to be fit compared to p should each marginal be Poisson. However, the number of marginal parameters will always scale linearly with the dimensions, as each marginal distribution will have some constant amount of parameters. The number of copula parameters will be equal to the number of pairwise combinations of features, $\frac{p(p-1)}{2}$, and so scales quadratically with the dimensions. Hence, the computational cost of the model is $\mathcal{O}(p^2)$. The computational cost scales with dimensions at the same rate as the pairwise GMM and grows more slowly with dimensions than does the GMM fit using the true likelihood.

5.5 Initialisation Strategies

Before the EM algorithm can run, the parameters must first be initialised, in particular the marginal parameters, copula parameters and mixing weights. Once these have been initialised, the E-step will produce an initial estimate for the set of indicator variables, Z. Then, the algorithm will iterate between the steps to update the parameter estimates. The following initialisation strategy follows that laid out by Kosmidis and Karlis (2016)[26]. First it is necessary to form K subsets that partition the data-set. This can be done using

K-means or other, relatively inexpensive, non-parametric approaches. Whilst most data-sets are not suited to this form of analysis, e.g. any data-set with binary variables, it is a viable initialisation strategy. On some occasions, the use of K-means can lead to the algorithm consistently converging to poor local optima. If this appears to be a concern, using a random initialisation may be a solution.

Once an initial partition has been formed, the mixing weights are determined by setting them equal to the proportion of observations initially assigned to each mixing component, i.e. $tau_k = \frac{n_k}{N}$, where n_k is the observations initially assigned to component k and N is the total number of observations. In the case of the copula mixture model, the parameters of each marginal distribution for each component, θ_k should be found through maximum likelihood estimation based on the observations currently assigned to that component. This should be a straightforward application of known MLE formulas, as no weighting of observations is required. Then the $\frac{p(p-1)}{2}$ copula parameters for the pairwise combinations of features should be calculated. Again, this happens for each mixing component. In the case of a GMM, after the formation of an initial partition and the initial estimate of mixing weights, the component mean vectors and covariance matrices should be estimated. Again, this can be done straightforwardly using unweighted, known MLE estimates.

6. Simulation Studies

6.1 Asymmetric Tail Dependency

As mentioned in section 2, GMMs are unable to account for asymmetric dependency. As discussed in section 3.3, certain copulas, e.g. the Clayton copula, display such asymmetries and so using them as components allows for a more flexible mixture model. In this simulation, 500 observations are randomly drawn from three component densities, with equal mixing probabilities, i.e. $\tau_1 = \tau_2 = \tau_3 = 1/3$. Each of the cluster densities are a copula with Gaussian marginals. Two clusters come from a Clayton copula, one from a Frank copula. Hence, this sample's only deviation from being generated by a GMM model is the dependence structure that is imposed by the copulas. Figure 2(a) shows the simulated data. Figure 2(b) shows the clusters fitted using a GMM while 2(c) shows the clusters fitted using a copula mixture model, with two Clayton and one Frank Copula and Gaussian marginals.

As expected, the GMM is unable to account for the asymmetric dependence structures that are generated by the Clayton copulas. Instead, it splits the simulations in the right tail of the three clusters into two separate clusters and then groups the remaining simulations into the third cluster. In so doing, it fails to accurately capture the data-generating process underlying the simulated data. The GMM model has a classification accuracy of 68.8% on this simulated data, however this perhaps overstates the useful of the model. Arguably the most interesting feature of the simulated data is the tail dependence in the two Clayton copulas and this is entirely missed by the GMM. Should this have been a real data-set, any inference based on the results of the GMM would likely be unhelpful. By comparison the copula mixture model, with appropriately chosen copulas an marginals, is able to accurately model the simulated data. It accurately captures the dependence structure of the data and produces a classification accuracy of 95.8%, with the only misclassifications being the observations close to the border between component confidence regions.

As mentioned in section 4.2, comparison between models fit using pairwise likelihood and true likelihood on the basis of information criteria is generally invalid. An exception to this is when working with 2-dimensional data, when the two approaches are identical. Hence, in this case, AIC and BIC are valid comparison metrics. The GMM has AIC = 6540 and BIC = 10646, both to the nearest integer. The corresponding values for the copula mixture model are 5173 and 9426. Hence, the copula mixture model outperforms the GMM regardless of which performance metric is considered.

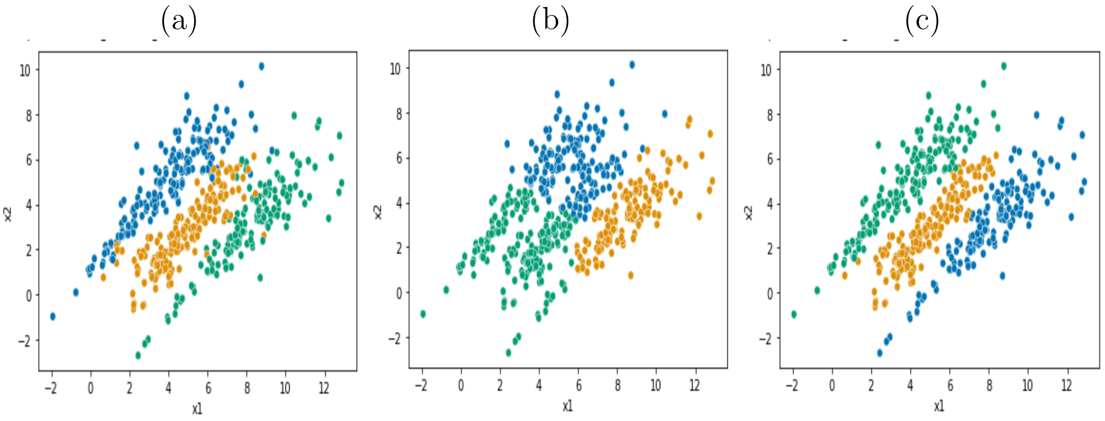

Figure 2: *Plot (a) gives the true clustering of the simulated data. Plot (b) gives the clusters estimated by a GMM. Plot (c) gives the clusters estimated by a copula mixture model.*

6.2 Discrete marginals

GMMs are unable to work with data that is non-continuous or has a bounded domain. The following simulation studies display the flexibility of the copula mixture model in this respect. For this simulation study, 300 observations were drawn from three component distributions with equal mixing proabability, i.e. $\tau_1 = \tau_2 = \tau_3 = 1/3$. The component distributions were 2-dimensional with Poisson marginals and a Frank copula. Table 1 gives the results of the copula mixture model, in terms of the parameter estimates. In general, the parameter estimates are close to the true values, indicating that the copula mixture model is able to accurately estimate the data-generating process.

6.3 Mixed-domain data

This section focuses on the ability of the copula mixture model to implement distributions that are defined on a finite domain. In particular, the simulation consists of 500 samples

Parameter	Component 1	Component 2	Component 3
Copula (θ)	4.80 (5)	5.04 (5)	4.31 (5)
Poisson (λ_1)	31.04 (30)	19.44 (20)	10.13 (10)
Poisson (λ_2)	30.10 (30)	49.52 (50)	40.99 (40)
Mixing weight (τ)	0.33 (0.33)	0.31 (0.33)	0.36 (0.33)

Table 1: *Parameter estimates (true vales) produced by copula mixture model. All values given to 2 decimal places*

drawn from two Clayton copulas and one Gumbel copula. In each case, the copulas have one gamma distribution and one beta distribution as a marginal. As shown in figure 3, this means that the first feature is defined on the positive real numbers, while the second is defined on the closed unit interval, $[0, 1]$. Non-parametric models would be unable to accurately model this data due to the difference in scale, while GMMs would lose efficiency due to the fact they cannot account for the finite domain of the data. As the data is relatively well separated, the copula mixture model achieves an accuracy of 99.4%. As in section 6.2, the marginal parameter estimates are relatively accurate. However, the copula parameter estimates were significantly less accurate. Again, the true value of θ for each mixing distribution was 5. The estimated values were $19, 13$ and 15, to the nearest integer. This is likely because the model incorrectly assigns the relative outliers of the clusters. As a result, a significant amount of the variance in the clusters in lost, and so the model is producing more tightly packed. A result of this is that the copula parameter estimate will be greater, as this leads to lower variance in the observations generated from said copula.

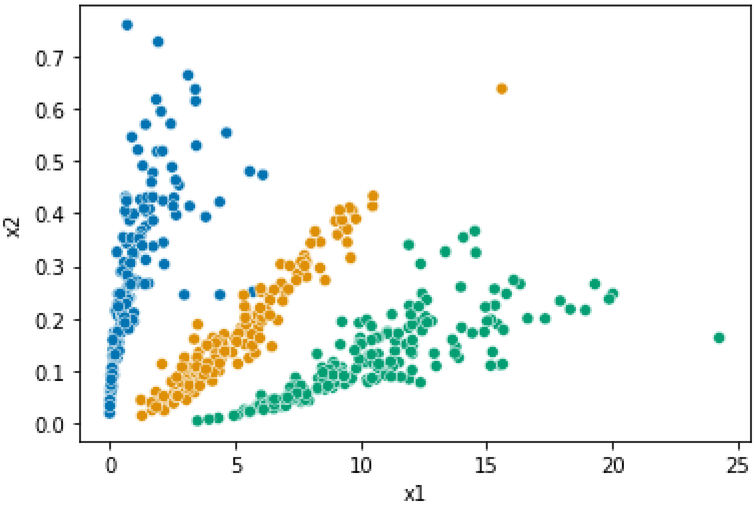

Figure 3: *Simulated mixed-domain data.*

6.3 Mixed-mode data

This simulation study works with mixed-mode data. Again, this a type of data that finite mixture models with multivariate Gaussian or t distributions as components cannot accomodate. 300 samples are drawn from two Frank copulas and One Gumbel copula. In each case, the first marginal distribution is normal and the second is Poisson. The mixing weights are equal, i.e. $\tau_1 = \tau_2 = \tau_3 = 1/3$. Table 2 gives the estimated and true parameters of the model. Again, the estimates of the marginal parameters are relatively accurate and centred around their true values, i.e. there is no appearance of bias across the components. Again, the estimates of the copula parameter, θ, appear to be less accurate. This is likely to be partially explained by the lower sample size in this and the previous simulation study. In this case, the misclassified points generally served to increase the within cluster variance and so the decreased the copula parameter estimates.

Parameter	Component 1	Component 2	Component 3
Copula (θ)	6.31 (5)	1.06 (5)	2.90 (5)
Normal mean (μ)	9.69 (10)	9.11 (10)	9.85 (10)
Normal variance (σ)	1.92 (2)	1.55 (2)	1.99 (2)
Poisson (λ)	10.12 (10)	22.20 (25)	40.09 (40)
Mixing weight (τ)	0.33 (0.33)	0.31 (0.33)	0.36 (0.33)

Table 2: *Parameter estimates (true vales) produced by copula mixture model. All values are given to 2 decimal places.*

6.4 Case Study

The following case study focuses on the data represented in figure 4. The data comes from the institute of Agrophysics of the Polish Academy of Life Sciences. It contains seven measurements taken from 210 observations. Each observation was a kernel of wheat and the kernels belonged to one of three types: Kama, Rosa and Canadian. Correct clustering observations from the same type of wheat is the aim of analysis. Each of the features is strictly positive and feature 2 is bounded to the unit interval, $[0, 1]$. As such, the marginal distributions used are: 6 gamma distributions and 1 beta distribution. A frank copula was used for each mixing component. The model performance is rather poor, with a classification accuracy of 59%. By comparison, Charytanowicz et al. (2010)[17] use a SOM clustering algorithm and are able to achieve a classification accuracy of 90%. One reason for the relatively poor performance is that this implementation of the copula mixture model does not support parametric rotations of copulas. This would likely help the mode given the relationships between some of the variables. Such an implementation is provided by Kosmidis and Karlis (2016)[26] and is shown to be useful at improving performance. Additionally, clustering this data-set is challenging. For example, the pairwise copula mixture model does outperform some methods, such as K-means. Hence, A more flexible approach and a slightly better separated data-set may lead to a higher classification accuracy.

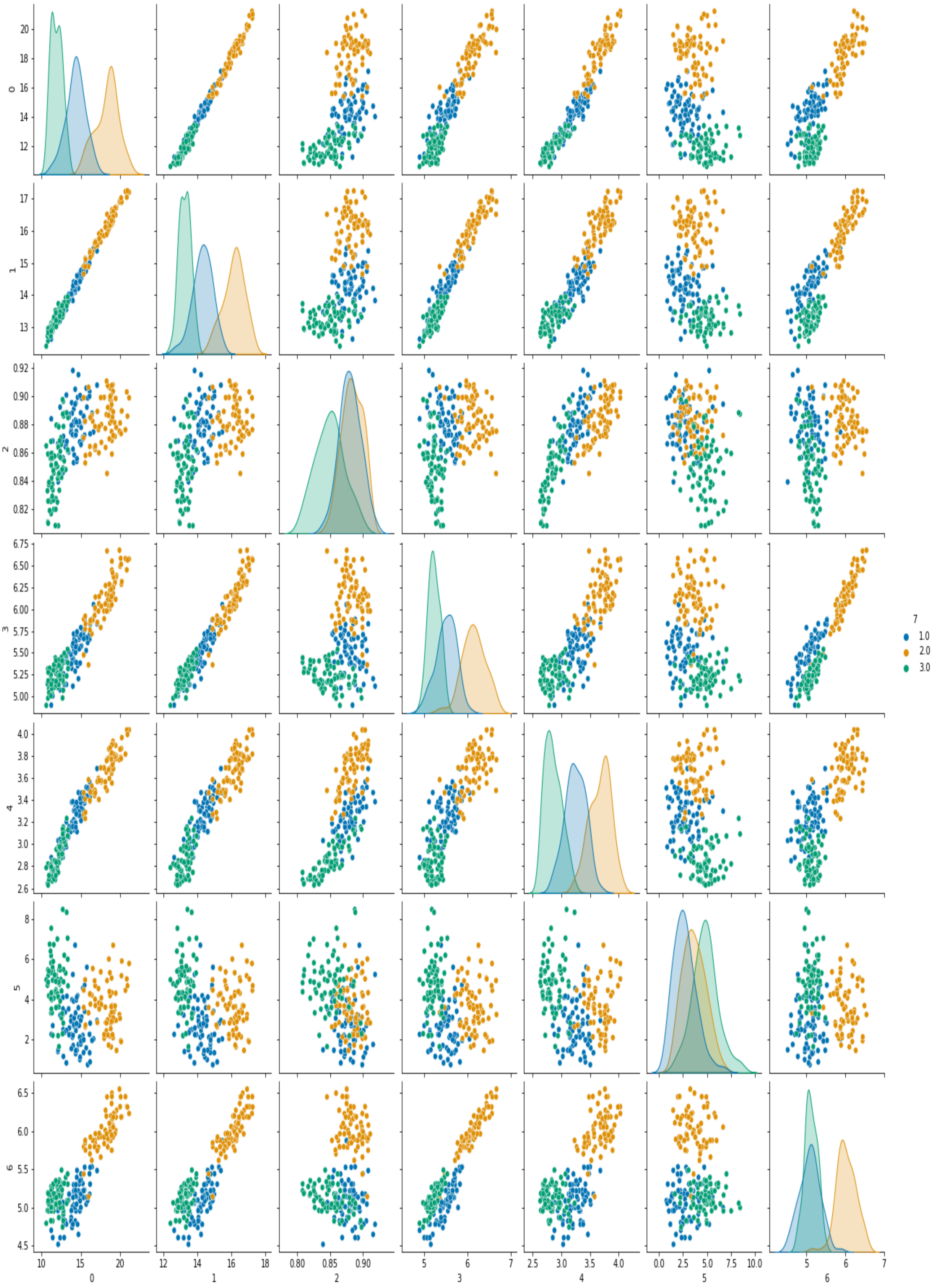

Figure 4: *Pairplot for seeds data.*

7. Discussion

The aim of this paper is to outline and implement a clustering algorithm that is more flexible and computationally efficient than those that are typically used for unsupervised learning, namely GMMs. This is done through a pairwise likelihood approach to copula mixture models, following the work of Kosmidis and Karlis (2016)[26]. However, there are significant improvements that could made over the implementation used for this paper. Firstly, it would be beneficial to support a wider variety of marginal and copula distributions. With respect to marginal distributions, the most commonly used marginals are supported, but a greater variety would allow for more flexible modelling. As mentioned in the introduction, whilst any joint distribution can be exactly approximated using copulas, the quality of the approximation will depend on whether the correct copula is chosen. As such, having a broad range of copulas to choose from as component densities will lead to a more flexible model, which would be expected to have a higher classification accuracy for any given data-set. When using a pairwise likelihood approach, the aim should be to support any copula that is closed under marginalisation. Only 4 are supported in this implementation. The inclusion of further copulas closed under marginalisation, e.g. t-copula or any Archimedean or nested-Archimedean copula, would be an improvement over the current implementation.

An additional approach that can increase model flexibility is allowing for parametric rotations of copulas, as in Kosmidis and Karlis (2016)[26]. At present the implementation will only be able to accurately positive correlations between variables, except in the case of the normal copula. Hence, supporting parametric rotations of copulas will be a significant improvement to the flexibility of the model compared to the current implementation.

With respect to the computational efficiency of the implementation as present, this could be improved through removing dependence on certain packages. Whilst care was taken to reduce the dependence on any package other than Python's built in *Math*[30] module and *NumPy*[29], the numerical optimiser used at each CM-step is from *Scipy*[31] and the pdf and cdf computations for the Gamma and Beta distributions were handled by *OpenTurns*[25]. As the optimiser is the component that contributes most significantly to the algorithms runtime, the current implementation is sub-optimal. A manual implementation of the optimisation steps would produce a far lower computational costs, should it be implemented.

As well as the limitations with the current implementation mentioned above, there are improvements that could be made with the theory of the implementation in this paper. The copula choice for each mixing component is a hyperparameter that will affect the models performance. This paper does not provide a formal heuristic for selecting which copula to choose for a given mixing component. These choices were primarily done through trial and error informed by exploratory data analysis. In the case of the simulation studies, the choice was straightforward due to the fact there was perfect information of the data-generating process which will never be the case in real applications. This approach becomes less available when working with more copula choices or large data-sets, as the cost of hyperparameter-tuning will become infeasible. As such, it would be beneficial to formalise a heuristic for copula selection. It is possible that Q-Q plots could be useful for this purpose, as pairwise combinations of the feature data with a fitted copula could be compared to different copulas and the one with the closest fit can be selected.

An additional concern is that, as the dimensionalty of the data increases, the number of parameters being estimated increases quadratically. As such, issues of over-fitting may arise, unless the sample size can increase sufficiently to offset the quadratically increasing number of free parameters. One possible approach to address over-fitting is to use some kind of shrinkage method. Whilst shrinkage methods introduce bias to a model, they are also able to improve statistical efficiency. For example, the log-likelihood could be penalised based on the scale of dependence between parameters, e.g. using the lasso penalty. For normal copulas, this could be implemented as being equal to the estimated covariances between parameters, whilst for any Archimedean copula, the penalty can be related to Kendall's τ, which will be directly proportional to the copula parameter. As the one of the main aims of this approach is its ability to scale to high dimensions, the implementation of a shrinkage method to prevent over-fitting, which can lead to poor classification accuracy and generalisation, would be a significant advancement.

References

[1] M Sklar. “Fonctions de repartition an dimensions et leurs marges”. In: *Publ. inst. statist. univ. Paris* 8 (1959), pp. 229–231.

[2] Bruce G Lindsay. “Composite likelihood methods”. In: *Contemporary mathematics* 80.1 (1988), pp. 221–239.

[3] Zvi Drezner and George O Wesolowsky. “On the computation of the bivariate normal integral”. In: *Journal of Statistical Computation and Simulation* 35.1-2 (1990), pp. 101–107.

[4] Jeffrey D Banfield and Adrian E Raftery. “Model-based Gaussian and non-Gaussian clustering”. In: *Biometrics* (1993), pp. 803–821.

[5] Gilles Celeux and Gerard Govaert. “Comparison of the mixture and the classification maximum likelihood in cluster analysis”. In: *Journal of Statistical Computation and simulation* 47.3-4 (1993), pp. 127–146.

[6] Xiao-Li Meng and Donald B Rubin. “Maximum likelihood estimation via the ECM algorithm: A general framework”. In: *Biometrika* 80.2 (1993), pp. 267–278.

[7] Jeff A Bilmes et al. “A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models”. In: *International Computer Science Institute* 4.510 (1998), p. 126.

[8] Ming-Hsuan Yang and Narendra Ahuja. “Gaussian mixture model for human skin color and its applications in image and video databases”. In: *Storage and retrieval for image and video databases VII*. Vol. 3656. International Society for Optics and Photonics. 1998, pp. 458–466.

[9] David Peel and Geoffrey J McLachlan. “Robust mixture modelling using the t distribution”. In: *Statistics and computing* 10.4 (2000), pp. 339–348.

[10] Tim Bedford and Roger M Cooke. “Probability density decomposition for conditionally dependent random variables modeled by vines”. In: *Annals of Mathematics and Artificial intelligence* 32.1 (2001), pp. 245–268.

[11] Gang Liang and Bin Yu. “Maximum pseudo likelihood estimation in network tomography”. In: *IEEE Transactions on Signal Processing* 51.8 (2003), pp. 2043–2053.

[12] Roger B Nelsen. “Properties and applications of copulas: A brief survey”. In: *Proceedings of the first brazilian conference on statistical modeling in insurance and finance*. Citeseer. 2003, pp. 10–28.

[13] Markus Haas, Stefan Mittnik, and Marc S Paolella. “Modelling and predicting market risk with Laplace–Gaussian mixture distributions”. In: *Applied Financial Economics* 16.15 (2006), pp. 1145–1162.

[14] Roger B Nelsen. *An introduction to copulas*. Springer Science & Business Media, 2007.

[15] Kanti V Mardia et al. “Maximum likelihood estimation using composite likelihoods for closed exponential families”. In: *Biometrika* 96.4 (2009), pp. 975–982.

[16] Roshan Joy Martis, Chandan Chakraborty, and Ajoy K Ray. “A two-stage mechanism for registration and classification of ECG using Gaussian mixture model”. In: *Pattern Recognition* 42.11 (2009), pp. 2979–2988.

[17] Małgorzata Charytanowicz et al. “Complete gradient clustering algorithm for features analysis of x-ray images”. In: *Information technologies in biomedicine*. Springer, 2010, pp. 15–24.

[18] Sylvia Frühwirth-Schnatter and Saumyadipta Pyne. “Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions”. In: *Biostatistics* 11.2 (2010), pp. 317–336.

[19] Jeffrey L Andrews, Paul D McNicholas, and Sanjeena Subedi. “Model-based classification via mixtures of multivariate t-distributions”. In: *Computational Statistics & Data Analysis* 55.1 (2011), pp. 520–529.

[20] Cristiano Varin, Nancy Reid, and David Firth. “An overview of composite likelihood methods”. In: *Statistica Sinica* (2011), pp. 5–42.

[21] Anastasios Panagiotelis, Claudia Czado, and Harry Joe. “Pair copula constructions for multivariate discrete data”. In: *Journal of the American Statistical Association* 107.499 (2012), pp. 1063–1072.

[22] Nema Dean and Rebecca Nugent. “Clustering student skill set profiles in a unit hypercube using mixtures of multivariate betas”. In: *Advances in Data Analysis and Classification* 7.3 (2013), pp. 339–357.

[23] Daeyoung Kim et al. “Mixture of D-vine copulas for modeling dependence”. In: *Computational Statistics & Data Analysis* 64 (2013), pp. 1–19.

[24] Sharon X Lee and Geoffrey J McLachlan. “On mixtures of skew normal and skew

$$t$$

-distributions". In: *Advances in Data Analysis and Classification* 7.3 (2013), pp. 241–266.

[25] Michaël Baudin et al. "OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation". In: *Handbook of Uncertainty Quantification.* Ed. by Roger Ghanem, David Higdon, and Houman Owhadi. Cham: Springer International Publishing, 2016, pp. 1–38. ISBN: 978-3-319-11259-6. DOI: `10.1007/978-3-319-11259-6_64-1`. URL: `https://doi.org/10.1007/978-3-319-11259-6_64-1`.

[26] Ioannis Kosmidis and Dimitris Karlis. "Model-based clustering using copulas with applications". In: *Statistics and computing* 26.5 (2016), pp. 1079–1099.

[27] Luca Scrucca et al. "mclust 5: clustering, classification and density estimation using Gaussian finite mixture models". In: *The R journal* 8.1 (2016), p. 289.

[28] C. Bouveyron. *Model-based Clustering and Classification for Data Science.* 1st ed. Cambridge: Cambridge University Press, 2019.

[29] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585 (2020), pp. 357–362. DOI: `10.1038/s41586-020-2649-2`.

[30] Guido Van Rossum. *The Python Library Reference, release 3.8.2.* Python Software Foundation, 2020.

[31] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: `10.1038/s41592-019-0686-2`.

[32] Özge Sahin and Claudia Czado. "Vine copula mixture models and clustering for non-Gaussian data". In: *arXiv preprint arXiv:2102.03257* (2021).