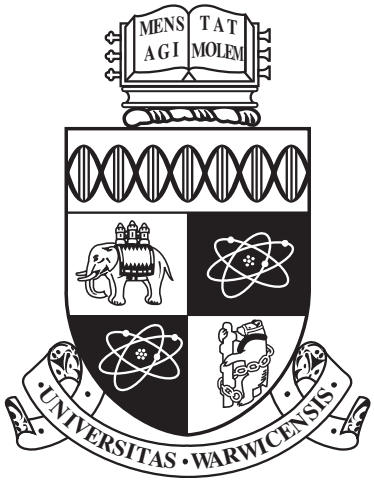

Shrinkage-driven unsupervised learning with clustering applications

by

Ben Barlow

Dissertation

Supervised by Ioannis Kosmidis

BSc (Hons) Data Science

Department of Computer Science

May 2020

THE UNIVERSITY OF
WARWICK

Contents

Abstract

Model-based clustering has largely focused on mixtures where the component distributions correspond to the multivariate normal distribution. A detailed review of such mixtures, their estimation through expectation-maximization algorithm and an analysis of their comprehensive implementation `mclust` are all given here. Later, a new flexible family of regularized copula-based mixture models are introduced for cluster analysis, which are considered an extension of mixtures of copulas offered in recent literature. The technique sees a shrinkage-driven approach to selecting dependence structure permitted by unconstrained parameterization of the copula parameter using angles. An appropriate procedure for estimation through expectation-conditional-maximization (ECM), framework for model selection that maximizes cluster separation and in-depth discussion regarding computational details are all included. The new clustering technique is illustrated by the study of simulated and real data and is shown to outperform previous proposals. In addition, the report encompasses a description and documentation of the R package rcbmm, which facilities the new methods and contains the first general ECM implementation available in R. Finally, limitations of the regularized copula-based mixture model and its implementation are outlined alongside a suggestion for future research directions.

Keywords Model-based clustering · Cluster analysis · Mixture models · Copula · ECM Algorithm

Acknowledgements

I would like to offer my special thanks to Ioannis Kosmidis, the supervisor of this dissertation, for invaluable support throughout the project. The assistance enabled challenging concepts to be understood and provided constructive advice on how to overcome problems that arose during development of the project's algorithms. The absence of this help would have forced a less ambitious project direction to be followed, which would have resulted in the output of the project (`rcbmm`) offering a less significant contribution to the field of model-based clustering.

1 Introduction

1.1 Limited families of finite mixture models

The goal of cluster analysis is to separate the data into subgroups, where each member of a subgroup has a common feature which they do not share with members of other subgroups. The application of finite mixture models to data for the purposes of clustering has been of increasing popularity since the publication of comprehensive literature Banfield and Raftery (1993) and McLachlan and Peel (2004). The approach is based on the assumption that $\mathbb{R}^p$-valued data samples $\boldsymbol{x}_1 \ldots, \boldsymbol{x}_n$ are independent, identically distributed according to the density or probability mass function (pmf)

$$h(\boldsymbol{x}_i; \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j), \tag{1}$$

where $\sum_{j=1}^{K} \pi_j = 1$ and $\pi_j \in (0, 1)$. The parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$ contains $\boldsymbol{\theta}_j$ which parameterizes the density (or pmf) of the jth mixture component $f_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ $(j = 1, \ldots, K)$. The component densities (or pmf) $f_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ often arise from the same parametric distribution family, although this need not be the case in general. In the majority of literature, the model-based clustering approach sees fitting a family of mixture models all according to (1), followed by selecting the most appropriate model from within this family using some criterion.

For continuous data, a common choice in recent decades for the densities $f_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ has been the multivariate normal distribution. Such models are very well established in the literature (for example, see Banfield and Raftery 1993; Celeux and Govaert 1995; Fraley and Raftery 2002), due in part to the convenience offered in estimation using the EM algorithm and the easy visualization of fitted components and mixture densities stemming from the property of closure under marginalization, but also due to the availability for modelling through the `mclust` package since its release in 1998 (Fraley and Raftery, 1998a) in the R software (RCore, 2016).

The challenge imposed by modelling mixture components using the multivariate normal distribution is the resultant clusters are limited to being elliptical in shape. The Gaussian distribution is often too restricted to formalize the cluster shapes one is interested in. In order to capture skewness and kurtosis of a single cluster, several normally distributed mixture components may be necessary, thus leading to incorrect inference about the number of clusters in the data (Jasra, 2006). Hennig (2010) proposes merging mixture components and interpreting their union as a single cluster, which can allow for the fitting of a single non-elliptical cluster whilst avoiding significant overestimation of the number of clusters in the underlying distribution. However, the merging problem defined by this study is highly subjective and not statistically identifiable, and hence the intervention of a statistician is necessary due to the absence of an "optimal" method for merging.

To address such practical issues formally, alternative solutions have aimed at creating more flexible families of mixture models where the component densities $f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)$ themselves capture skewness and kurtosis within the data. The additional parameter (the degrees of freedom) introduced by mixtures of multivariate t distributions (for example, see McLachlan and Peel 2004; Lin et al. 2004; Andrews and Mcnicholas 2012) allows for the regulation of heavy tails of each component, thus enabling mixture components to accommodate outliers better than the multivariate normal distribution. Other applications of finite mixture models have been to the univariate skew-normal (Lin et al., 2007) and skew-t distribution (Lin et al., 2007), as well as their corresponding multivariate distributions (see, for example, Lin and Lin 2010; Frühwirth-Schnatter and Pyne 2010; Lee and Mclachlan 2014). All of the aforementioned studies overcome the limited shape of multivariate normal mixture components and enable skewness and kurtosis to be captured in a single cluster, which results in a more trivial relationship between the number of components in the mixture and the number of clusters found in the data.

Whilst the construction of more parsimonious models than multivariate normal mixtures has been achieved since the turn of the millennium, all of the approaches outlined force data to obey very strict marginal properties. In addition, further complications are observed when the gaps between the "true" clusters are small. Kosmidis and Karlis (2016), *Example 1.1*, demonstrates that current methods are incapable of capturing the true shape of clusters under such conditions. As a result of the limitations discussed, a new framework that allows a wider range of exotic shapes for mixture components is needed, without forfeiting any of the flexibility offered by current proposals.

1.2 Flexible families of finite mixture models

The significance of copulas in the modelling of multivariate data follows directly from the flexibility they offer; prescribed marginal distributions alongside a flexible framework for describing dependence is available when modelling with copulas. Many families of copulas have been outlined in literature; both Nelsen (2007) and Joe (1997) provide an extensive analysis.

The application of finite mixture models to copula functions is inviting as the flexibility they offer when modelling multivariate data has a direct effect on the range of shapes that can be achieved by mixture components, and they allow for the modelling of data with continuous and non-continuous features in a natural way. The most common copula families in model-based clustering are the Archimedean, which includes Clayton's, Frank's and Gumbel's copula models, and the Elliptical, which consists of the Gaussian and t-copula models.

Their importance within clustering has not yet been investigated in great detail, although, Jajuga and Papla (2006), Di Lascio and Giannerini (2012) and Vrac et al. (2012) are some examples of attempts that have been made. Kosmidis and Karlis (2016) gives a review encompassing model specifications for copula-based mixture models for continuous and non-continuous data, efficient procedures for estimation, and a discussion on topics such as the closure properties of copula-based mixtures under marginalization and parametric rotations in the sample space for continuous, real-valued data.

This dissertation aims to extend the ideas presented in Kosmidis and Karlis (2016) for continuous data, by defining a new family of regularized copula-based mixture models (RCBMM) where the component densities are modelled with the Gaussian copula. Previous examples of the application of regularization to finite mixture models are Bhattacharya and McNicholas (2014) for overcoming the shortcomings of BIC for model selection in higher dimensions and Fraley and Raftery (2007) for avoiding singularities during estimation when the component distributions are normally distributed, but an application to copula-based mixtures has not yet been made. The structure of the remainder of the report is as follows.

Section 2 provides a specification for Gaussian mixture models and details for maximum likelihood estimation through expectation-maximization (EM) algorithm, together with a detailed review of modelling such mixtures via `mclust` and its approach towards model selection and obtaining initial parameter estimates for EM. A different mixture model specification and details regarding estimation through expectation-conditional-maximization (ECM) algorithm is presented in Sect. 3 for mixtures based on copulas. Section 4 introduces a method for reparameterizing correlation matrices in order to achieve unconstrained optimization, which is coupled with the copula-based mixture specification given in Sect. 3 to define a new family of regularized copula-based mixture models in Sect. 5. A shrinkage-driven approach towards controlling the model's dependence structure, details of estimation and an appropriate framework for model selection are all discussed here. Sect. 6 outlines the challenges of estimation of regularized copula-based mixtures in terms of obtaining starting values, the approach towards achieving unconstrained optimization, numerical issues and the availability of parallelization due to component-wise updates. The application of the methodology to real and simulated data is given in Sect 7. Sections 8 & 9 respectively specify the implementation of the project's associated software and review the approach towards management of the project in terms of methodology, time management, risk mitigation and the success of tools used. The report concludes in Section 10, which separately discusses some limitations of the regularized copula-based model and its software, as well as considering the project's contribution to the field of model-based clustering and guidance for further research.

2 Gaussian mixture models

2.1 Model specification

Given multivariate data $\boldsymbol{x}_{obs} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T$, with $\dim(\boldsymbol{x}_i) = p$, the likelihood of a mixture model with density (or pmf) (1) with K components is

$$L(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}) = \prod_{i=1}^{n} \sum_{j=1}^{K} \pi_j f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j), \tag{2}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^T$ are the mixing proportions and the component density (or pmf) $f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)$ is parameterized by the parameter vector $\boldsymbol{\theta}_j$ $(j = 1, \ldots, K)$.

The Gaussian mixture model, which assumes all subgroups within the population are generated from the normal distribution, has received particular attention in the statistical literature. A detailed analysis of finite mixture models in general, with an emphasis on Gaussian mixtures, can be found in Titterington et al. (1985), McLachlan and Basford (1988) and McLachlan and Peel (2004). In the multivariate case, the component distributions $f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)$ are parameterized by their mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$ as $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j^T, \boldsymbol{\Sigma}_j^T)^T$ and defined as

$$f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) = \frac{1}{(\sqrt{2\pi})^p} \frac{1}{\sqrt{\det(\boldsymbol{\Sigma}_j)}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j) \Big\}. \tag{3}$$

The resultant mixture components are centred at $\boldsymbol{\mu}_j$ with their geometric features, such as shape, volume and orientation, defined by $\boldsymbol{\Sigma}_j$. The components are ellipsoidal in shape with the density of data increasing as they approach $\boldsymbol{\mu}_j$. The standard Gaussian mixture model has p and $\frac{1}{2}p(p-1)$ independent parameters for each mixture component's mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$, respectively, and $K-1$ independent parameters for the mixing proportions. This leads to a total of $(K-1)+(K\times p)+\frac{K}{2}p(p-1)$ parameters for a model comprising K mixture components.

A number of attempts have been made to reduce the number of parameters required for modelling the covariance matrices $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K$. Common examples include modelling $\boldsymbol{\Sigma}_j = \lambda_j \boldsymbol{I}$ as a spherical component varying in volume for each index j, or $\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}$ where components are not limited to a spherical shape but limited to sharing their shape with all other components in the mixture (Friedman and Rubin, 1967). Banfield and Raftery (1993), Celeux and Govaert (1995) and Fraley and Raftery (2002) parameterize $\boldsymbol{\Sigma}_j$ with its eigenvalue decomposition

$$\boldsymbol{\Sigma}_j = \lambda_j \boldsymbol{D}_j \boldsymbol{A}_j \boldsymbol{D}_j^T, \tag{4}$$

where λ_j is a constant controlling the volume of the mixture component, $\boldsymbol{D}_j$ is the orthogonal matrix of eigenvectors controlling the component's orientation, and $\boldsymbol{A}_j$ is a diagonal matrix, with $\det(\boldsymbol{A}_j) = 1$, controlling the shape of the density contours. The imposition of various constraints on the covariance structure leads to a family of models, and the best model is typically selected via some criterion.

An arbitrarily close modelling of any distribution can be achieved via Gaussian mixture models by increasing the number of mixture components. However, increasing the number of parameters by $p + \frac{1}{2}p(p-1)$ with each additional mixture component can quickly lead to a model that overfits to the data. Hence, finding the optimal number of mixture components K is regarded as one of the most prominent issues in model-based clustering.

For a fixed K and fixed covariance structure, the parameters of the model can be estimated using two commonly used maximum likelihood approaches; namely the mixture approach or the classification approach. The mixture approach, which is used for the remainder of this report, sees the maximization of (2) over the parameters $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$; a detailed description of which is given by the following introduction to the EM algorithm. A framework for the classification approach can be found in Celeux and Govaert (1995).

2.2 Estimation using Expectation-Maximization algorithm

The expectation-maximization algorithm (EM algorithm; Dempster et al. 1977) is a general procedure for performing maximum likelihood estimation (see Appendix A) when the observations are viewed as incomplete data. It retains its name due to the iterative approach adopted between the expectation step (E-step) and maximization step (M-step), and is a remarkable concept in the field because of the generality of the associated theory.

As remarked earlier, the EM algorithm sparked interest in the modelling of data via finite mixture models since its simplicity allows for the estimation of Gaussian mixtures in a straightforward manner. It aims to find an appropriate root of

$$\frac{\partial \log L(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}} = \mathbf{0}, \tag{5}$$

where $\boldsymbol{\Psi} = (\boldsymbol{\pi}^T, \boldsymbol{\theta}^T)^T$ are the parameters of the mixture model. The logarithmic function $\log L(\boldsymbol{\Psi})$ is called the incomplete data log-likelihood of the model and defined as

$$l(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}) = \sum_{i=1}^{n} \log \sum_{j=1}^{K} \pi_j f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j). \tag{6}$$

When performing estimation of the parameters of a Gaussian mixture model, the unlabelled observed data $\boldsymbol{x}_{obs} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T$ are paired with their component-label vectors $\boldsymbol{z} = (\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_n^T)^T$ to form the complete data $\boldsymbol{x}_c = (\boldsymbol{x}_{obs}^T, \boldsymbol{z}^T)^T$. Hence, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ are taken to be realizations of independent, identically distributed random vectors $Z_1, \ldots, Z_n$, which are indicator vectors corresponding to n random variables all taking a multinomial distribution of one draw from K categories with probabilities $\pi_1, \ldots, \pi_K$. Thus, $\boldsymbol{z}_i = ((\boldsymbol{z}^1)_i, \ldots, (\boldsymbol{z}^K)_i)^T$, where $(\boldsymbol{z}^j)_i$, which is the ith observation in the jth column of $\boldsymbol{z}$, is defined as

$$(\boldsymbol{z}^j)_i = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ is belongs to the } j\text{th subgroup in the data,} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Since the random vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ are unknown during estimation, in the context of EM $z_{ij} \in [0, 1]$ instead represents the soft probability that the observed value $\boldsymbol{x}_i$ belongs to the jth component in the mixture. The complete data pair $(\boldsymbol{x}_{obs}^T, \boldsymbol{z}^T)^T$ leads to the complete data log-likelihood

$$l_c(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}_{obs}, \boldsymbol{z}) = \sum_{i=1}^{n} \sum_{j=1}^{K} z_{ij} \{\log \pi_j + \log f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)\}. \tag{8}$$

The "sum-log-sum" nature of (6) causes maximization of the incomplete data log-likelihood to be numerically challenging, thus the EM algorithm maximizes the expectation of the complete data log-likelihood (8) by repeating the E-step and M-step until convergence is achieved. In the expectation step, a soft assignment is made to compute the posterior probability $z_{ij} = (\boldsymbol{z}^j)_i$ of the ith observation belonging to the jth mixture component using the current estimates of the parameters. In the maximization step, these probabilities are used in weighted maximum likelihood estimation to update the estimates of the parameters $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, with the aim of the updates being the maximization of the conditional expectation of the complete data log-likelihood given the conditional probabilities z_{ij} previously calculated $(i = 1, \ldots, n, j = 1, \ldots, K)$.

Starting with an initial estimate $\boldsymbol{\Psi}^{(0)} = (\boldsymbol{\pi}^{(0)T}, \boldsymbol{\theta}^{(0)T})^T$ of the parameters, at the $(l+1)$th iteration of the algorithm, where the current value of the parameters is $\boldsymbol{\Psi}^{(l)} = (\boldsymbol{\pi}^{(l)T}, \boldsymbol{\theta}^{(l)T})^T$, the following updates are performed:

- *E-step*: Compute the conditional expectation of the complete data log-likelihood $l_c(\boldsymbol{\Psi}; \boldsymbol{x})$ given the data $\boldsymbol{x}_{obs}$ and current estimate of the parameters $\boldsymbol{\Psi}^{(l)}$, written as

$$Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(l)}) = E_{\boldsymbol{\Psi}^{(l)}}\{l_c(\boldsymbol{\Psi}; \boldsymbol{x}_{obs})\}, \tag{9}$$

  which is linear in the unobserved data z_{ij} as demonstrated by (8). Hence, the E-step reduces to computing the conditional expectation of the unobserved data given the observed data,

$$\begin{aligned} E_{\boldsymbol{\Psi}^{(l)}}(z_{ij}; \boldsymbol{x}_{obs}) &= z_{ij}^{(l+1)} \\ &:= \frac{\pi_j^{(l)} f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j^{(l)})}{\sum_{m=1}^{K} \pi_m^{(l)} f_m(\boldsymbol{x}_i; \boldsymbol{\theta}_m^{(l)})}, \end{aligned}$$

  where $f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j^{(l)})$ is as defined in (3) and $\boldsymbol{\theta}_j^{(l)}$ is used as the current estimate of $\boldsymbol{\theta}_j$. The resultant set of values $z_{ij}^{(l+1)}$ $(i = 1, \ldots, n, j = 1, \ldots, K)$ represents the posterior probability of membership

of sample $\boldsymbol{x}_i$ to the jth component in the mixture using the current estimates of the parameters $\boldsymbol{\Psi}^{(l)}$.

- *M-step 1*: Set

$$\pi_j^{(l+1)} = \frac{\sum_{i=1}^{n} z_{ij}^{(l+1)}}{n}.$$

to maximize (9) w.r.t the mixing proportions $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_K$.

- *M-step 2*: Maximize

$$\sum_{i=1}^{n}\sum_{j=1}^{K} z_{ij}^{(l+1)} \log f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)$$

w.r.t the parameter vectors $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ thereby maximizing (9), by performing the updates (for proof, see appendix B)

$$\boldsymbol{\mu}_j^{(l+1)} = \frac{\sum_{i=1}^{n} z_{ij}^{(l+1)} \boldsymbol{x}_i}{\sum_{i=1}^{n} z_{ij}^{(l+1)}}, \qquad \boldsymbol{\Sigma}_j^{(l+1)} = \frac{\sum_{i=1}^{n} z_{ij}^{(l+1)} (\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(l+1)})^T (\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(l+1)})}{\sum_{i=1}^{n} z_{ij}^{(l+1)}}.$$

The EM algorithm repeats the E-step and M-step until either the absolute or relative increase in the log-likelihood is less than than a fixed ϵ. The convergence criterion is assessed using the incomplete data log-likelihood $l(\boldsymbol{\Psi}^{(l)}; \boldsymbol{x}_{obs})$ in (6), evaluated after each iteration of the algorithm using the current estimate of the parameters $\boldsymbol{\Psi}^{(l)}$. Dempster et al. (1977) demonstrated that the likelihood increases with each iteration of the algorithm and convergence to a local maximum is guaranteed under fairly mild conditions, with the value of the maximum depending on the starting values of the parameters $\boldsymbol{\Psi}^{(0)}$.

2.3 Initialization of the EM algorithm

Despite the EM algorithm's convenient convergence properties, convergence to a global optimum is not guaranteed (McLachlan and Krishnan 2007, *Sect. 1*). Under the condition that the likelihood surface has multiple maxima, which is usually the case for Gaussian mixture models, the result of EM's estimation is heavily dependent on the starting values. In general, there are two sets of methods for achieving good results using EM for model-based clustering, referred to as stochastic and deterministic.

The former sees the creation of multiple models using a selection of random starting values. The log-likelihood of all of the associated models is assessed, and the model with maximal log-likelihood is chosen. The *RndEM* method of Maitra (2009) uses random starting values drawn from a feasible region to produce multiple corresponding models. The requirement for a range of models follows from the obvious intuition that a run of the EM algorithm using one set of starting values generated randomly is likely to produce poor results. A more rigorous method, *emEM* of Biernacki et al. (2003), uses a short run of the EM algorithm for multiple sets of starting values, and then proceeds by performing one long run until convergence is met using the model that achieved the highest log-likelihood during the short runs. The *RndEM* method can be interpreted identical to *emEM*, with the difference being that the short EM phase stops after the very first parameter estimation. It is clear *emEm* is computationally more expensive, however, this results in *RndEM* possessing the prominent advantage that it can be attempted for a larger number of random starting values under the same time constraints.

Deterministic methods for obtaining starting values aim to find a hard partitioning (clustering) of the data, and the clustering's associated parameters are used as the starting values for EM. An example of such methods is the k-means algorithm (MacQueen et al., 1967). Another method with the same goal as k-means for obtaining an initial partition of the data is hierarchical clustering. Hierarchical methods can be either *agglomerative*, where each data point begins as its own cluster and groups are merged at each step of the algorithm, or *divisive*, in which one or more groups are split at each stage.

During the application of model-based hierarchical agglomerative clustering (MBHAC) to mixture models, the merging is performed by recursively joining the two clusters that have the smallest dissimilarity; commonly assessed by maximizing the classification likelihood (see, Celeux and Govaert 1995, *Sect. 2.2*) which is an alternative to (6) in terms of mixture model likelihood functions. Using MBHAC

to obtain starting values does not guarantee that the EM will converge to a global optimum, but it usually provides reasonable starting points.

In summary, there are an array of methods that all aim to achieve the best results using EM. The disadvantage of deterministic approaches is the incapability of providing more than one starting point for EM, while stochastic methods can be computationally more intense. The underlying argument behind all approaches is a good initialization is crucial since the EM algorithm tends to produce meaningful results if started from reasonable starting values (Wu, 1983).

2.4 Model selection

A common challenge in any unsupervised cluster analysis problem is selecting the number of components K. If a value too small is chosen then the model fails to capture the full clustering complex found in the data, and a value too large can lead to overfitting. A factor to consider when choosing K is the complexity of the model. An increase in model complexity can result in a decrease in the number of components needed to fully capture the clustering formation in the dataset. Hence, the range of shapes offered by the distribution of each mixture component can have a significant effect on the number of clusters needed. For example, consider a subset of data generated from an unknown ellipsoidal cluster. If a Gaussian mixture model with equal-volume spherical components is used to model the data, then more than one component is needed to model the ellipsoidal cluster.

The flexibility offered by a mixture component, belonging to a Gaussian mixture defined in Sect. 2.1, is dependent on the parameterization of the component's covariance matrix. The component's volume, orientation and shape are respectively controlled by λ_j, $\boldsymbol{A}_j$ and $\boldsymbol{D}_j$ in (4), and constraining one or more to be fixed irrespective of the index j results in the imposition of cross-cluster constraints on the mixture. For example, fixing $\lambda_1 = \cdots = \lambda_K = \lambda$ causes each cluster to occupy an equal volume in the mixture. Unconstrained covariance matrices allow a greater flexibility in terms of modelling the observed data, however, imposing cross-cluster constraints reduces the number of parameters to be estimated thereby dampening the effects of overfitting to the observed data.

Varying the number of components K alongside imposing various constraints on the covariance structure sees the creation of a family of Gaussian mixture models. The optimal parameterization and number of components must be selected using some criterion; Fraley and Raftery (1998b) amongst others select both simultaneously using the Bayesian Information Criterion (BIC; Schwarz et al. 1978). The fit of a mixture model improves as more mixture components are added, thus the log-likelihood (6) is strictly increasing as a function of K. As a result, the comparison of the log-likelihood for the purposes of model selection does not suffice. Instead, as the likelihood increases with the number of components, BIC penalizes the log-likelihood by adding a penalty term for the number of independent parameters estimated. The BIC of a model with K components and covariance parameterization denoted by M is given by

$$\mathrm{BIC}_{K,M} = 2l(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}_{obs}) - v \log n, \tag{10}$$

where $l(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}_{obs})$ is the maximized incomplete data log-likelihood (6), v is the number of independent parameters estimated and n is the number of observations in $\boldsymbol{x}_{obs}$. The BIC can be used to compare models with different numbers of components, or differing covariance parameterizations, or both. The pair (K, M) that maximizes $\mathrm{BIC}_{K,M}$ is selected as the optimal model.

2.5 `mclust` implementation

2.5.1 Introduction to `mclust`

The populairty of model-based clustering via Gaussian mixtures encouraged the production of the comprehensive R package `mclust` (Scrucca et al., 2016), and the straightforward mechanism it offers for estimating such mixtures in turn caused the prevalence of Gaussian mixture models to continue to rise. The package allows for automated and efficient estimation of Gaussian mixtures for a variety of purposes.

The implementation provides a system for model-based clustering, classification and density estimation based on Gaussian mixtures. It also supports functions for performing single E- and M-steps and methods for visualizing fitted models with clustering, classification and density estimation results all provided. For multivariate data, a variety of covariance structures are achieved through eigenvalue decomposition in (4); cross-cluster constraints are imposed on mixture components' volume, shape and

orientation by constraining $\lambda_j, \boldsymbol{A}_j$ and $\boldsymbol{D}_j$, respectively, to be equal or variable across groups. For univariate data, the available models are simply E for equal variance and V for variable variance, however, univariate data is not relevant here.

2.5.2 Initialization

Before performing EM for model estimation, `mclust` uses MBHAC to obtain an initial partitioning of the data which allows for starting values to be found for EM. Given observed data $\boldsymbol{x}_{obs} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T$, the algorithm performs $n-1$ steps in which it merges two clusters, which results in one cluster containing n data points having started from n clusters all containing one data point. Since `mclust` assumes the mixture components take multivariate normal distribution, the parameters of $f_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ are $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j^T, \boldsymbol{\Sigma}_j^T)^T$ $(j = 1, \ldots, K)$. When the covariance $\boldsymbol{\Sigma}_j$ is allowed to vary across clusters, the criterion to be minimized at each hierarchical stage is

$$W_K = \sum_{j=1}^{K} n_j \log \left| \frac{\boldsymbol{W}_j}{n_j} \right|, \tag{11}$$

where $\boldsymbol{W}_j$ and n_j are the sample cross-product matrix for the jth group and number of observations in the jth group $(j = 1, \ldots, K)$, respectively. This approach is equivalent to maximizing the classification likelihood of the model as introduced in Sect. 2.3.

A drawback of hierarchical methods is the computational effort required is proportional to the square of the number of observations, which is especially challenging for large datasets. However, `mclust`'s MBHAC implementation has the advantage of being independent of the number of mixture components. This is computationally convenient in practice, since a single run of MBHAC can be stored and used to supply the starting values for a whole family of models fitted using EM.

Another shortcoming of MBHAC is the difficulties it encounters in the presence of coarse data, which is generated from the rounding of continuous data when measured or from data with a discrete nature. In such cases, multiple merging decisions on a given iteration of the MBHAC algorithm result in obtaining the same value in (11). Hence, the ordering of variables and permutation of observations can have a significant effect on the initial partitioning formed using MBHAC. An application in Scrucca et al. (2016) achieved BIC values -2810.777 and -2821.339 using a Flea beetles dataset available in the R package `tourr` (Wickham et al., 2011), using an equal number of mixture components and the same covariance parameterization in both models. The difference in BIC, which is considered significant since it is greater than 10 (Kass and Raftery, 1995), was provoked by simply reordering the variables.

Scrucca and Raftery (2015) overcame this issue by projecting the data through a suitable transformation, with the aim of enhancing separation amongst groups, prior to identifying an initial partitioning. This removes the dependence on the ordering of the variables, and the results of the paper suggest an improved model fitting process and more accurate clustering results. The transformations are integrated into the MBHAC function `hc()` offered by `mclust`, and the transformation of choice is specified by using the function `mclust.options()`. The details of the transformations proposed in the paper are summarised here.

Let $\hat{\boldsymbol{x}}_{obs}$ be the $n \times p$ centred matrix corresponding to the observed data $\boldsymbol{x}_{obs}$ and $\hat{\boldsymbol{\Sigma}} = \{s_{ij}\} = \hat{\boldsymbol{x}}_{obs}^T \hat{\boldsymbol{x}}_{obs}/n$ be the $p \times p$ sample covariance matrix. The singular value decomposition of $\hat{\boldsymbol{x}}_{obs}$ is

$$\hat{\boldsymbol{x}}_{obs} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T = \sum_{i=1}^{r} \lambda_i \boldsymbol{u}_i \boldsymbol{v}_i^T, \tag{12}$$

where $\lambda_1 \geq \cdots \geq \lambda_r > 0$ are the singular values, $\boldsymbol{u}_i$ the right singular vectors and $\boldsymbol{v}_i$ the left singular vectors of the centred data matrix. The rank $r \leq \min\{n, p\}$ is the rank of $\hat{\boldsymbol{x}}_{obs}$. The transformations of interest are now introduced.

- *Data sphering* (`SPH`): An elliptically shaped symmetric group of data points is transformed to a spherically shaped collection of points by applying

  $$\boldsymbol{Z}_{SPH} = \hat{\boldsymbol{x}}_{obs} \boldsymbol{V} \boldsymbol{D}^{-1} \sqrt{n} = \boldsymbol{U}\sqrt{n},$$

  where $\boldsymbol{V}$ is the matrix of eigenvectors and $\boldsymbol{D}^{-1}\sqrt{n} = \mathit{diag}(\sqrt{n}/\lambda_i)$ the diagonal matrix of square root inverse of eigenvalues from the spectral decomposition of the sample marginal covariance $\hat{\boldsymbol{\Sigma}}$.

The features are uncorrelated with unit variances and centred at zero, since $\mathbb{E}(\boldsymbol{Z}_{SPH}) = \boldsymbol{0}$ and the variance-covariance matrix of $\boldsymbol{Z}_{SPH}$ is the identity matrix $\boldsymbol{I}_p$.

- *PCA scores for covariance matrix* (`PCS`): This transformation is obtained as

$$\boldsymbol{Z}_{PCS} = \hat{\boldsymbol{x}}_{obs}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{D},$$

where the features are again centred at zero since $\mathbb{E}(\boldsymbol{Z}_{PCS}) = \boldsymbol{0}$ and uncorrelated since $Var(\boldsymbol{Z}_{PCS}) = \boldsymbol{D}^2/n = diag(\lambda_i^2/n$. In contrast to the previous transformation, the variances are now decreasing and equal to the eigenvalues of $\hat{\boldsymbol{\Sigma}}$.

An extension of the singular value decomposition (12) can be used to allow for two more transformations. Define $\boldsymbol{S} = diag(s_1^2, \ldots, s_p^2)$ as the diagonal matrix of sample variances, then the centred and scaled data matrix $\hat{\boldsymbol{x}}_{obs}\boldsymbol{S}^{-1/2}$ can be decomposed as

$$\hat{\boldsymbol{x}}_{obs}\boldsymbol{S}^{-1/2} = \boldsymbol{U}^*\boldsymbol{D}^*\boldsymbol{V}^{*T} = \sum_{i=1}^{r} \lambda_i^* \boldsymbol{u}_i^* \boldsymbol{v}_i^{*T},$$

- *PCA scores from correlation matrix* (`PCR`): The principl component transformation of the correlation matrix is given by

$$\boldsymbol{Z}_{PCR} = \hat{\boldsymbol{x}}_{obs}\boldsymbol{S}^{-1/2}\boldsymbol{V}^* = \boldsymbol{U}^*\boldsymbol{D}^*,$$

where $\mathbb{E}(\boldsymbol{Z}_{PCR}) = \boldsymbol{0}$ and $Var(\boldsymbol{Z}_{PCR}) = \boldsymbol{D}^{*2}/n = diag(\lambda_i^{*2}/n)$. Hence, the features are centred at zero, and uncorrelated with variances equal to the eigenvalues of the marginal sample correlation matrix $\boldsymbol{S}^{-1/2}\hat{\boldsymbol{\Sigma}}_{obs}\boldsymbol{S}^{-1/2}$.

- *Scaled SVD projection* (`SVD`): The scaled SVD projection is defined as

$$\boldsymbol{Z}_{SVD} = \hat{\boldsymbol{x}}_{obs}\boldsymbol{S}^{-1/2}\boldsymbol{V}^*\boldsymbol{D}^* = \boldsymbol{U}^*\boldsymbol{D}^{*1/2},$$

where $\mathbb{E}(\boldsymbol{Z}_{SVD}) = \boldsymbol{0}$ and $Var(\boldsymbol{Z}_{SVD}) = \boldsymbol{D}^*/n = diag(\lambda_i^*/n)$. In this case, the variances are equal to the square root of the eigenvalues of the marginal sample correlation matrix $\boldsymbol{S}^{-1/2}\hat{\boldsymbol{\Sigma}}_{obs}\boldsymbol{S}^{-1/2}$. Again, the features are centred at zero and uncorrelated.

MBHAC commonly finds sub-optimal partitions resulting in the subsequent EM procedure converging to a local maximum. Applying the transformations aims to enhance the separation between groups to assist MBHAC in providing reasonable starting values to EM, thereby alleviating some of the risk of achieving a sub-optimal fit in the subsequent model. There is no evidence to suggest one transformation performs better than another, hence, it is advised that all are used for a given initialization task and the most appropriate starting values are selected by assessing the corresponding likelihood of the model achieved by each transformation. The transformations defined are not considered the work of this dissertation, but the work of Scrucca and Raftery (2015).

2.5.3 Model Selection

The model selection framework in `mclust` uses BIC by default. For multivariate data, `mclust`'s ability to fit mixtures with $K \in \{1, \ldots, 9\}$ components and 14 possible characterizations for the within-group covariance matrix results in a total of 140 models that can be constructed. When construing a family of models by the function call `Mclust()`, the model space can be restricted by specifying the arguments `G` and `modelNames`, representing the number of components and covariance parameterization, respectively. For instance, parsing G= $(2, 3)$ and *modelNames*= ("VVV","VVE") produces a family of 4 models; one for each permutation of the two parameters. The model of interest in this report is "VVV", which stands for "variable volume, variable shape, variable orientation", characterizing a particular parameterization of a multivariate normal component distribution's variance-covariance matrix.

2.6 Limitations of GMMs and `mclust`

Despite the efficiency of the EM implementation, convenient initialization procedure and flexibility in covariance parameterizations offered by `mclust`, the package fundamentally suffers from a number of limitations following from the components taking a Gaussian distribution.

Firstly, clusters are limited to occupying an elliptical shape which is inappropriate in the common case that the underlying clustering complex of subgroups in the observed data is not elliptical. Moreover, Gaussian mixtures force the observed data to obey very strict marginal properties, namely each feature is assumed to be normally distributed, which is unrealistic in a number of applications and also prevents the modelling of mixed- or bounded-domain data in a natural way. Whilst a transformation can be applied to such data to allow for the modelling using normal mixtures, the results of Dean and Nugent (2013) suggests better results are obtained when working with observed data directly. Copulas instead offer a framework that allows one to work directly with bounded-domain data. Moreover, the flexible choice of marginal distributions results in clusters that occupy an exotic range of shapes.

3 Copula-based mixture models

3.1 Model specification

It must be stated, the concept of using copulas to define families of mixture models is not new (see, for example, Arakelian and Karlis 2014; Kosmidis and Karlis 2016), however, the methodology has received significantly less attention in the literature than Gaussian mixtures. A copula $C(u_1, \ldots, u_p; \boldsymbol{\psi})$ is a multivariate cumulative distribution function with uniform marginals, often regulated by a singular parameter $\boldsymbol{\psi}$ which aims to describe the dependence structure of the distribution. A statistical modelling problem of multivariate data using copulas can be decomposed into two steps: identify the marginal distributions of the data and link these marginals by selecting an appropriate copula function. Sklar's theorem (see, Nelsen 2007, *Sect. 2.3*) shows that every multivariate distribution can be written via a copula representation.

In the context of model-based clustering, copula functions and a selection of marginal distributions can be used to capture the distribution of the mixture components in a model defined by (1). The component densities (or pmf for non-continuous data) f_j correspond to the distribution function

$$F_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = C_j(G_1(x_{i1}; \boldsymbol{\gamma}_{j1}), \ldots, G_p(x_{ip}; \boldsymbol{\gamma}_{jp}); \boldsymbol{\psi}_j), \qquad (j = 1, \ldots, K) \tag{13}$$

where $G_1, \ldots, G_p$, which respectively have associated parameters $\boldsymbol{\gamma}_{j1}, \ldots, \boldsymbol{\gamma}_{jp}$ for the jth mixture component, are univariate marginal cumulative distribution functions, and C_j is the distribution function of the copula. The separation of the statistical modelling problem into marginal properties and dependence properties is reflected formally, by the decoupling of a mixture component's parameters $\boldsymbol{\theta}_j$ into marginal parameters $\boldsymbol{\gamma}_j = (\boldsymbol{\gamma}_{j1}^T, \ldots, \boldsymbol{\gamma}_{jp}^T)^T$ and copula parameter $\boldsymbol{\psi}_j$ as $\boldsymbol{\theta}_j = (\boldsymbol{\gamma}_j^T, \boldsymbol{\psi}_j^T)^T$. It is assumed that the density $c_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) = \partial^p \{C_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j)\} / \partial x_{i1} \ldots \partial x_{ip}$ exists, thus the density of the jth component for continuous marginals is defined as

$$f_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = c_j(G_1(x_{i1}; \boldsymbol{\gamma}_{j1}), \ldots, G_p(x_{ip}; \boldsymbol{\gamma}_{jp}); \boldsymbol{\psi}_j) \prod_{t=1}^{p} g_t(x_{it}; \boldsymbol{\gamma}_{jt}) \tag{14}$$

where, omitting the component index, $g_t(x_{it}; \boldsymbol{\gamma}_t) = \partial G_t(x_{it}; \boldsymbol{\gamma}_t) / \partial x_{it}$. The case where $G_1, \ldots, G_P$ are discrete distribution functions is not discussed here.

In comparison to a Gaussian mixture where each component takes a multivariate normal distribution, each component in a copula-based model can arise from a different parametric distribution family by varying the choice of copula C_j. This creates a mechanism for constructing diverse families of models, with flexibility enhanced further by the dividing of the multivariate distribution into two components, namely marginal distributions and the copula function. Moreover, an immediate consequence of Sklar's theorem is copula-based mixture models match the flexibility offered by families of mixture models described in the literature. Hence, a powerful framework is defined that encompasses all known mixture models and has the ability to capture a range of cluster shapes that other methods cannot, thereby overcoming the limited elliptical cluster shapes found in Gaussian mixture models.

3.2 Estimation using Expectation-Conditional-Maximization algorithm

3.2.1 Expectation-Conditional-Maximization algorithm: general case

Despite the intuitive interpretation and numerical stability of the EM algorithm, there are cases in which other algorithms are preferred. Little and Rubin (1987) present a variety of applications in which complete data maximum likelihood estimation is complicated. In such cases, one can aim to increase the objective function Q in (9) during the M-step instead of maximizing it. This intuition leads to a class of algorithms referred to as GEM algorithms, however, in general they require further specification on the increasing Q function to guarantee convergence. The ECM algorithm of Meng and Rubin (1993) is a subclass of GEM that, under the right conditions, conveniently shares all the convergence properties of the EM algorithm.

Suppose a general vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_S)$, partitioned into S subvectors, parameterizing a function to maximize Q. The ECM algorithm separates the EM algorithm's second maximization step into S computationally simpler conditional-maximization (CM) steps, where the sth step aims to maximize Q over $\boldsymbol{\theta}$ with some function, $g_s(\boldsymbol{\theta})$, of the parameters constrained ($s = 1, \ldots, S$). The resulting algorithm is computationally less intense due to the simplicity of conditional complete data maximum likelihood estimation. For a more detailed review of the general ECM algorithm accompanied by motivating examples see Meng and Rubin (1993).

3.2.2 Expectation-Conditional-Maximization algorithm: copula-based mixture models

A direct consequence of the decoupling of $\boldsymbol{\theta}_j$ ($j = 1, \ldots, K$) into the marginal parameters and copula parameter is the maximization of $\sum_{j=1}^{K}\{\sum_{i=1}^{n} z_{ij} \log f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)\}$ becomes complicated and the traditional EM algorithm no longer suffices. The ECM algorithm is hence used for estimation of copula-based mixtures, such that the maximization in the EM algorithm's complicated M-step 2 is separated into two CM-steps. The aim of the two CM-steps is to first maximize the conditional expectation of the complete data log-likelihood with respect to the marginal parameters given the current value of the copula parameters, and then maximize with respect to the copula parameters given the updated value of the marginal parameters.

Thus, starting with $\boldsymbol{\Psi}^{(0)} = (\boldsymbol{\pi}^{(0)T}, \boldsymbol{\theta}^{(0)T})^T$, at the $(l+1)$th iteration of the ECM the following steps are performed.

- *E-step*: Same as EM, such that the densities $f_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ correspond to (14) instead of taking multivariate normal distribution (3).
- *M-step 1*: Same as EM.
- *CM-step 1*: Maximize the conditional expectation of the complete data log-likelihood in (8) over constrained $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$, with the function $g_1(\boldsymbol{\theta}) = (\boldsymbol{\psi}_1^T, \ldots, \boldsymbol{\psi}_K^T)^T$ fixed, by maximizing

$$\sum_{i=1}^{n}\sum_{j=1}^{K} z_{ij}^{(l+1)} \Big\{ \log c_j(G_1(x_{i1}; \boldsymbol{\gamma}_{j1}), \ldots, G_p(x_{ip}; \boldsymbol{\gamma}_{jp}); \boldsymbol{\psi}_j^{(l)}) + \sum_{i=1}^{p} \log g_t(x_{it}; \boldsymbol{\gamma}_{jt}) \Big\}$$

  w.r.t $\boldsymbol{\gamma}_{11}, \ldots, \boldsymbol{\gamma}_{jp}, \ldots, \boldsymbol{\gamma}_{K1}, \ldots, \boldsymbol{\gamma}_{Kp}$ to obtain updated values $\boldsymbol{\gamma}_{11}^{(l+1)}, \ldots, \boldsymbol{\gamma}_{1p}^{(l+1)}, \ldots, \boldsymbol{\gamma}_{K1}^{(l+1)}, \ldots, \boldsymbol{\gamma}_{Kp}^{(l+1)}$ for the marginal parameters.

- *CM-step 2*: Maximize the conditional expectation of the complete data log-likelihood in (8) over constrained $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$, with the function $g_2(\boldsymbol{\theta}) = (\boldsymbol{\gamma}_{11}^T, \ldots, \boldsymbol{\gamma}_{jp}^T, \ldots, \boldsymbol{\gamma}_{K1}^T, \ldots, \boldsymbol{\gamma}_{Kp}^T)^T$ fixed, by maximizing

$$\sum_{i=1}^{n}\sum_{j=1}^{K} z_{ij}^{(l+1)} \Big\{ \log c_j(G_1(x_{i1}; \boldsymbol{\gamma}_{j1}^{(l+1)}), \ldots, G_p(x_{ip}; \boldsymbol{\gamma}_{jp}^{(l+1)}); \boldsymbol{\psi}_j) \Big\}$$

  w.r.t $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_k$ to obtain updated values $\boldsymbol{\psi}_1^{(l+1)}, \ldots, \boldsymbol{\psi}_K^{(l+1)}$ for the copula parameters.

The main difference between EM and ECM is explained by considering the objective function to maximize Q in (9), which represents the expectation of the complete data log-likelihood (8). The aim of EM's M-step 2 on the $(l+1)$th iteration of the algorithm is to maximize $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(l)})$ over $\boldsymbol{\theta}$ in $\boldsymbol{\Psi} = (\boldsymbol{\pi}^T, \boldsymbol{\theta}^T)^T$. A single CM-step instead maximizes Q over $\boldsymbol{\theta}$ with some function of $\boldsymbol{\theta}$, $g_s(\boldsymbol{\theta})$, fixed at the value of it's

most recent estimate computed by ECM. Therefore, each CM-step is said to increase Q rather than maximize it.

3.3 Gaussian copula

The copula of interest in this dissertation is the Gaussian copula (proposed by Li 2000) from the Elliptical family. The Gaussian copula is flexible since the copula parameter $\boldsymbol{\psi}$ is a $p \times p$ unstructured correlation matrix taking values in $[-1, 1]$, thereby allowing for equal degrees of positive and negative dependence. The distribution function is given by

$$C(u_1, \ldots, u_p, \boldsymbol{\psi}) = \Phi_p(\Phi^{-1}\{u_1\}, \ldots, \Phi^{-1}\{u_p\}; \boldsymbol{\psi}), \tag{15}$$

where $\Phi^{-1}\{.\}$ is the inverse distribution function of a standard univariate normal distribution, and $\Phi_p(...)$ is the distribution function of a standard p-variate normal distribution. Assuming the density $c(u_1, \ldots, u_p; \boldsymbol{\psi}) = \partial^p\{C(u_1, \ldots, u_p; \boldsymbol{\psi})\}/\partial u_1 \ldots \partial u_p$ exists, it follows

$$c(u_1, \ldots, u_p; \boldsymbol{\psi}) = \phi_p(\Phi^{-1}\{u_1\}, \ldots, \Phi^{-1}\{u_p\}; \boldsymbol{\psi}),$$

where $\phi_p(...)$ is the density function of a standard p-variate normal distribution.

Since the copula parameter of the Gaussian copula is a $p \times p$ correlation matrix, $\boldsymbol{\psi}_j$ has the general form

$$\boldsymbol{\psi} = \begin{pmatrix} 1 & \rho_{12} & \ldots & \rho_{1p} \\ \rho_{12} & 1 & \ldots & \vdots \\ \vdots & \vdots & \ddots & \rho_{p-1,p} \\ \rho_{1p} & \ldots & \rho_{p-1,p} & 1 \end{pmatrix},$$

where $\rho_{ij} \in [-1, 1]$ represents the correlation between the i and jth variable $(i, j = 1, \ldots, p, i < j)$.

4 Reparameterization of the copula parameter

4.1 Motivation for reparameterizing the copula parameter

Similarly to constraining elements of the eigenvalue decomposition of a mixture component's covariance matrix in a Gaussian mixture model, a shrinkage method can be applied to a copula-based mixture component's copula parameter to achieve a range of parameterizations for the dependence structure of the model.

Example 1 *Suppose a copula-based mixture model with $K = 3$ components, capturing a clustering pattern in observed data $\boldsymbol{x}_{obs} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, with $\dim(\boldsymbol{x}_i) = 2$ $(i = 1, \ldots, n)$. Two marginal distribution functions G_1 and G_2 are required and the Gaussian copula is chosen as the copula function of each mixture component. Hence, the component density $f_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$, which corresponds to (14) with C_j corresponding to the the Gaussian copula's distribution function (15), is parameterized by the parameter vector $\boldsymbol{\theta}_j = (\boldsymbol{\gamma}_j^T, \boldsymbol{\psi}_j^T)^T$, where the elements on the minor diagonal of the 2×2 correlation matrix $\boldsymbol{\psi}_j$ take value $\rho_j \in [-1, 1]$,*

$$\boldsymbol{\psi}_j = \begin{pmatrix} 1 & \rho_j \\ \rho_j & 1 \end{pmatrix} \qquad (j = 1, 2, 3).$$

To fit the model, one can apply ECM and instead maximize a penalized form of the log-likelihood (6) defined as

$$l'(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}) = \sum_{j=1}^{3} \Big\{ \Big(\sum_{i=1}^{n} \log \pi_j f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) \Big) + \lambda \rho_j^2 \Big\} \tag{16}$$

where $\lambda \geq 0$ is a parameter controlling the magnitude of an L_2 penalty applied to ρ_j $(j = 1, 2, 3)$.

The introduction of regularization on ρ_j encourages the parameter to approach 0, which in turn causes $\boldsymbol{\psi}_j$ $(j = 1, \ldots, K)$ to approach the identity matrix which is referred to as zero correlation for the remainder of this report. By varying the value of λ, it is possible to achieve a whole range of dependence structures for the mixture components.

However, maximizing (16) during an iteration of ECM via constrained optimization of the copula parameter on the interval [-1,1] is computationally unattractive. A method of reparameterizing each $\boldsymbol{\psi}_j$ to allow for unconstrained optimization is necessary.

4.2 An unconstrained parameterization of the copula parameter

Consider a $p \times p$ correlation matrix $\boldsymbol{R} = (R_{ij})$ with each element constrained on $[-1, 1]$ $(i, j = 1, \ldots, p)$. A flexible representation of $\boldsymbol{R}$ is achieved by exchanging it with a $(p-1) \times (p-1)$ matrix of angles $\boldsymbol{\Theta} = (\theta_{ij})$, parameterized by $\frac{1}{2}p(p-1)$ angles in the range $(0, \pi]$. The replacement of $\boldsymbol{R}$ with $\boldsymbol{\Theta}$ is performed by first exchanging $\boldsymbol{R}$ with its upper-triangular Cholesky factor $\boldsymbol{X}$, such that $\boldsymbol{R} = \boldsymbol{X}^T\boldsymbol{X}$, followed by exchanging $\boldsymbol{X}$ with angles using the following identities.

$$X_{11} = 1, \qquad X_{jj} = \prod_{k=1}^{j-1} \sin\theta_{k,j-1}, \qquad X_{ij} = \cos\theta_{i,j-1} \prod_{k=1}^{i-1} sin\theta_{k,j-1} \qquad (i, j = 1, \ldots, p, i < j) \tag{17}$$

By re-expressing $\sin\theta_{ij}$ and $\cos\theta_{ij}$ as s_{ij} and c_{ij}, respectively, the Cholesky factor $\boldsymbol{X}$ is expressed as

$$\boldsymbol{X} = \begin{pmatrix} 1 & c_{11} & c_{12} & \cdots & c_{1,p-1} \\ & s_{11} & c_{22}s_{12} & \cdots & c_{2,p-1}s_{1,p-1} \\ & & s_{22}s_{12} & & c_{3,p-1}s_{2,p-1}s_{1,p-1} \\ & & & \ddots & \vdots \\ & & & & c_{p-1,p-1}\prod_{m=1}^{p-2} s_{m,p-1} \\ & & & & \prod_{m=1}^{p-1} s_{m,p-1} \end{pmatrix}. \tag{18}$$

The matrix of hyperspherical coordinates (18) has the following convenient properties. Firstly, the requirement $(\theta_{ij}) \in (0, \pi]$ ensures (i) $\boldsymbol{X}^T\boldsymbol{X}$ is always non-negative definite, and (ii) the Cholesky factor $\boldsymbol{X}$ is unique. Secondly, the trigonometric expression (17) ensures $(\boldsymbol{X}^T\boldsymbol{X})_{ij} \in [-1, 1]$ $(i, j = 1, \ldots, p)$, with all elements on the major diagonal $(i = j)$ equal to 1. Hence, $(\boldsymbol{X}^T\boldsymbol{X})$ always satisfies the requirements of a correlation matrix.

If $(\boldsymbol{x}^1), \ldots, (\boldsymbol{x}^p)$ represent the columns of $\boldsymbol{X}$, then $(\boldsymbol{x}^2), (\boldsymbol{x}^3), \ldots, (\boldsymbol{x}^p)$ imply a collection of $1, 2, \ldots, p-1$ angles. Define a $(p-1) \times (p-1)$ upper triangular matrix $\boldsymbol{\Theta} = (\theta_{i,j})$, such that the angles corresponding to the jth column of $\boldsymbol{\Theta}$ correspond to the $(j+1)$th column of $\boldsymbol{X}$, formally

$$\theta_{i,j} = \begin{cases} \arccos(X_{i,j+1}) & (i = 1, j = 1, \ldots, p-1) \\ \arccos\left(X_{i,j+1} / \prod_{k=1}^{i-1} \sin(\arccos(X_{k,j+1})) \right) & (i = 2, \ldots, p-1, j = 2, \ldots, p-1, i < j). \end{cases} \tag{19}$$

The definition of $\boldsymbol{\Theta} = (\theta_{ij})$ in (19) is obtained by performing elementary algebra on the identities (17).

It is clear a one-to-one relationship exists between the correlation matrix $\boldsymbol{R}$ and matrix of angles $\boldsymbol{\Theta}$. Given a $p \times p$ unstructured correlation matrix $\boldsymbol{R}$, its Cholesky factor $\boldsymbol{X}$ always exists which allows the construction of a $(p-1) \times (p-1)$ matrix of angles $\boldsymbol{\Theta} = (\theta_{ij})$ via (19). Conversely, a matrix of angles $\boldsymbol{\Theta}$ defines a given $\boldsymbol{X}$, which produces a correlation matrix $\boldsymbol{R} = \boldsymbol{X}^T\boldsymbol{X}$. The uniqueness of the mapping in both directions arises from constraining the angles $\theta_{ij} \in (0, \pi]$.

Reflecting on the motivation for the reparameterization outlined in Section 4.1, the angles θ_{ij} as parameters are unconstrained on $(0, \pi]$, and further transformation can facilitate unconstrained parameterization on the whole real line.

For the benefit of the remainder of the report, define $f_{\boldsymbol{\Theta}} : [-1, 1]_{(p \times p)} \to (0, \pi]_{(p-1) \times (p-1)}$ as the function that exchanges a $p \times p$ unstructured correlation matrix $\boldsymbol{R}$ with a $(p-1) \times (p-1)$ upper-triangular matrix of angles $\boldsymbol{\Theta}$, and $f_{\boldsymbol{\Theta}}^{-1}$ it's inverse which always exists. In addition, define $\boldsymbol{\Theta}_d^*$ as the $d \times d$ upper-triangular matrix of angles with all non-zero entries equal to $\frac{\pi}{2}$,

$$\boldsymbol{\Theta}_d^* = \begin{pmatrix} \frac{\pi}{2} & \frac{\pi}{2} & \cdots & \frac{\pi}{2} \\ & \frac{\pi}{2} & \cdots & \frac{\pi}{2} \\ & & \ddots & \vdots \\ & & & \frac{\pi}{2} \end{pmatrix}. \tag{20}$$

The significance of this matrix is explained by the following example.

Example 2 *Consider the correlation matrix* $\boldsymbol{R} = \boldsymbol{I}_3$*, where* $\boldsymbol{I}_3$ *is the* 3×3 *identity matrix. It is observed that the upper-triangular Cholesky factor* $\boldsymbol{X}$ *of* $\boldsymbol{R}$ *is*

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 & 0 \\ & 1 & 0 \\ & & 1 \end{pmatrix}.$$

and the matrix of angles produced using (19) is

$$\boldsymbol{\Theta} = \begin{pmatrix} \frac{\pi}{2} & \frac{\pi}{2} \\ & \frac{\pi}{2} \end{pmatrix} = \boldsymbol{\Theta}_2^*.$$

It is clear that a 3×3 *correlation matrix* $\boldsymbol{R}$ *with major diagonal entries all* 1 *and all other elements taking value* 0 *is achieved if and only if* $\theta_{ij} = \pi/2$ *(*$1 \leq i \leq j \leq p-1$*). More generally, if* $\boldsymbol{R} = \boldsymbol{I}_p$ *(*$p \geq 2$*) then* $f_{\boldsymbol{\Theta}}(\boldsymbol{R})$ *defines the unique* $(p-1) \times (p-1)$ *upper-triangular matrix of angles* $\boldsymbol{\Theta}_{p-1}^*$*.*

5 Regularized copula-based mixture models

The goal is to define a new method for fitting a regularized copula-based mixture model of K components to n samples of p-dimensional data $\boldsymbol{x}_{obs} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T$, using maximum likelihood estimation for the marginal parameters of each component and a flexible parameterization of the copula parameter of the mixture components to allow shrinkage driven methods to control the dependence structure of the mixture.

5.1 Model specification

The model is specified as a mixture of Gaussian copula model with K components, each corresponding to the distribution function (13) with C_j chosen as the Gaussian copula's distribution function defined (15). Hence, the cumulative ditribution function of the jth component ($j = 1, \ldots, K$) is

$$F_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = \Phi_p(\Phi^{-1}\{G_1(x_{i1}; \boldsymbol{\gamma}_{j1})\}, \ldots, \Phi^{-1}\{G_p(x_{ip}; \boldsymbol{\gamma}_{jp})\}; \boldsymbol{\psi}_j),$$

where $\Phi^{-1}(.)$ is the inverse distribution function of a standard univariate normal distribution and $\Phi_p(...)$ is the distribution function of a standard p-variate normal distribution with correlation matrix $\boldsymbol{\psi}_j$. The jth component distribution's parameter vector $\boldsymbol{\theta}_j$ is partitioned into the marginal parameters and copula parameter as $\boldsymbol{\theta}_j = (\boldsymbol{\gamma}_j^T, \boldsymbol{\psi}_j^T)^T$.

The marginal parameters of the jth component $\boldsymbol{\gamma}_j = (\boldsymbol{\gamma}_{j1}^T, \ldots, \boldsymbol{\gamma}_{jK}^T)^T$ parameterize the univariate marginal cumulative distribution functions $G_1, \ldots, G_p$, which are arbitrarily chosen allowing the convenient construction of mixture models that can handle any type of continuous data. The copula parameters $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_K$ are $p \times p$ unstructured correlation matrices taking values in $[-1, 1]$, that are re-expressed as angles during estimation using the method described in Sect. 4.2.

The mixture model takes density as defined in (1), where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^T$ are the mixing proportions, and component density $f_j(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ ($j = 1, \ldots, K$) is expressed as

$$f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) = \phi_p(\Phi^{-1}\{G_1(x_{i1}; \boldsymbol{\gamma}_{j1})\}, \ldots, \Phi^{-1}\{G_p(x_{ip}; \boldsymbol{\gamma}_{jp})\}; \boldsymbol{\psi}_j) \times \prod_{t=1}^{p} \frac{g_t(x_{it}; \boldsymbol{\gamma}_{jt})}{\phi_1(\Phi^{-1}\{G_t(x_{it}; \boldsymbol{\gamma}_{jt})\})}, \tag{21}$$

where $\phi_p(...)$ is the density function of a standard p-variate normal distribution with correlation matrix $\boldsymbol{\psi}_j$, and $g_t(x_{it}; \boldsymbol{\gamma}_t) = \partial G_t(x_{it}; \boldsymbol{\gamma}_t)/\partial x_{it}$ is the density function for the tth marginal ($j = 1, \ldots, K, t = 1, \ldots, p$). The specification of the model is concluded by re-expressing the density (1) as

$$h(\boldsymbol{x}_i; \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j \Bigg\{ \phi_p(\Phi^{-1}\{G_1(x_{i1}; \boldsymbol{\gamma}_{j1})\}, \ldots, \Phi^{-1}\{G_p(x_{ip}; \boldsymbol{\gamma}_{jp})\}; \boldsymbol{\psi}_j) \times \prod_{t=1}^{p} \frac{g_t(x_{it}; \boldsymbol{\gamma}_{jt})}{\phi_1(\Phi^{-1}\{G_t(x_{it}; \boldsymbol{\gamma}_{jt})\})} \Bigg\}. \tag{22}$$

The number of independent parameters in the model is now considered. Omitting the component index j, if $\Lambda_t = |\boldsymbol{\gamma}_t|$ represents the number of parameters required for the tth marginal distribution, then $\sum_{t=1}^{p} \Lambda_t$ marginal parameters are required for the marginal parameters of the jth component. Since each copula parameter $\boldsymbol{\psi}_j$ $(j = 1, \ldots, K)$ represents a $p \times p$ unstructured correlation matrix, $\frac{1}{2}p(p-1)$ parameters are required to model the dependence structure of each component distribution. Finally, $(K-1)$ parameters are required for the mixing proportions. Hence, the total number of parameters to be estimated is

$$q = (K-1) + (K \times \sum_{t=1}^{p} \Lambda_t) + (K \times \frac{1}{2}p(p-1)). \tag{23}$$

5.2 A shrinkage-driven approach to selecting dependence structure

The specification so far does not define a new statistical framework for model-based clustering. In fact, Kosmidis and Karlis (2016) applied an identical mixture of Gaussian copula model for varying K to a real-world dataset, and even extended the model defined in Sect. 5.1 to allow $\boldsymbol{\psi}_j$ exchangeable as well as unstructured $(j = 1, \ldots, K)$. To introduce a new flexible mechanism for model-based clustering, a regularization term is appended to the log-likelihood of the model to encourage the copula parameter $\boldsymbol{\psi}_j$ to approach zero correlation $(j = 1, \ldots, K)$, thereby affecting the dependence structure of the model. The approach can be considered an unconstrained generalization of the methodology outlined in Example 1. The incomplete data log-likelihood becomes

$$l'(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}_{obs}) = \left(\sum_{i=1}^{n} \log \sum_{j=1}^{K} \pi_j f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) \right) - \lambda \left(\sum_{j=1}^{K} \Big\{ \sum_{\omega \in \Theta_j} (\omega - \frac{\pi}{2})^2 \Big\} \right), \qquad \lambda \in \mathbb{R}_{\geq 0} \tag{24}$$

where the first parenthesis is the incomplete data log-likelihood (6) of an unregularized mixture model, and the second parenthesis forms an L_2 penalty whose magnitude is controlled by a tuning hyperparameter $\lambda \geq 0$, and $\boldsymbol{\Theta}_j = f_{\boldsymbol{\Theta}}(\boldsymbol{\psi}_j)$ is the matrix of angles corresponding to the jth component's copula parameter as introduced in Section 4.2. As Example 2 explains, if the $(p-1) \times (p-1)$ upper-triangular matrix $\boldsymbol{\Theta}$ is such that all non-zero elements are equal to $\frac{\pi}{2}$, then its corresponding correlation matrix is the identity matrix $\boldsymbol{I}_p$. Hence, the L_2 penalty encourages $\boldsymbol{\psi}_j$'s associated matrix of angles to approach $\boldsymbol{\Theta}^*_{(p-1)}$ in (20), which in turn causes $\boldsymbol{\psi}_j$ to approach zero correlation. By varying the value of λ, an infinite amount of dependence structures can be achieved by the mixture. This provides a new mechanism for model-based clustering, where the dependence structure is controlled by a hyperparameter λ.

It follows the corresponding complete data log-likelihood is

$$l'_c(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}_{obs}, \boldsymbol{z}) = \left(\sum_{i=1}^{n} \sum_{j=1}^{K} z_{ij} \{ \log \pi_j + \log f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) \} \right) - \lambda \left(\sum_{j=1}^{K} \Big\{ \sum_{\omega \in \Theta_j} (\omega - \frac{\pi}{2})^2 \Big\} \right),$$

which, for convenience, can be re-expressed by performing elementary algebra as

$$l'_c(\boldsymbol{\pi}, \boldsymbol{\theta}; \boldsymbol{x}_{obs}, \boldsymbol{z}) = \sum_{j=1}^{K} \left\{ \Big\{ \sum_{i=1}^{n} z_{ij} \{ \log \pi_j + \log f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) \} \Big\} - \lambda \sum_{\omega \in \Theta_j} (\omega - \frac{\pi}{2})^2 \right\}. \tag{25}$$

5.3 Estimation using Expectation-Conditional-Maximization algorithm

Fix $K \in \{2, \ldots, 9\}$ amd $\lambda \geq 0$. The parameter vector $\boldsymbol{\Psi}$ of the model is defined as $\boldsymbol{\Psi} = (\boldsymbol{\pi}^T, \boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$, where the jth mixture component's parameter vector $\boldsymbol{\theta}_j = (\boldsymbol{\gamma}_{j1}^T, \ldots, \boldsymbol{\gamma}_{jp}^T, \boldsymbol{\psi}_j^T)^T$ $(j = 1, \ldots, K)$. Starting with an initial estimate $\boldsymbol{\Psi}^{(0)}$ for $\boldsymbol{\Psi}$, at the $(l+1)$th iteration of the algorithm the following steps are performed.

- *E-step*: Compute the posterior probability of membership z_{ij} of sample $\boldsymbol{x}_i$ to the jth component

$$z_{ij}^{(l+1)} = \frac{\pi_j^{(l)} f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j^{(l)})}{\sum_{m=1}^{K} \pi_m^{(l)} f_m(\boldsymbol{x}_i; \boldsymbol{\theta}_m^{(l)})}. \tag{26}$$

- *M-step 1*: Set

$$\pi_j^{(l+1)} = \frac{\sum_{i=1}^{n} z_{ij}^{(l+1)}}{n}.$$

- *CM-step 1* (*M-step 2*): For each component, $j = 1, \ldots, K$, maximize

$$\sum_{i=1}^{n} z_{ij} \Big\{ \log \phi_p(\Phi^{-1}\{G_1(x_{i1}; \boldsymbol{\gamma}_{j1})\}, \ldots, \Phi^{-1}\{G_p(x_{ip}; \boldsymbol{\gamma}_{jp})\}; \boldsymbol{\psi}_j) + \sum_{i=1}^{p} \log \frac{g_t(x_{it}; \boldsymbol{\gamma}_{jt})}{\phi_1(\Phi^{-1}\{G_t(x_{it}; \boldsymbol{\gamma}_{jt})\})} \Big\} \tag{27}$$

  w.r.t to $\boldsymbol{\gamma}_{j1}, \ldots, \boldsymbol{\gamma}_{jp}$ to obtain updated values $\boldsymbol{\gamma}_{j1}^{(l+1)}, \ldots, \boldsymbol{\gamma}_{jp}^{(l+1)}$ for the marginal parameters ($j = 1, \ldots, K$).

- *CM-step 2* (*M-step 2*): For each component, $j = 1, \ldots, K$, maximize

$$\Big\{ \sum_{i=1}^{n} z_{ij} \log \phi_p(\Phi^{-1}\{G_1(x_{i1}; \boldsymbol{\gamma}_{j1})\}, \ldots, \Phi^{-1}\{G_p(x_{ip}; \boldsymbol{\gamma}_{jp})\}; \boldsymbol{\psi}_j) \Big\} - \lambda \sum_{\omega \in \boldsymbol{\Theta}_j} (\omega - \frac{\pi}{2})^2, \tag{28}$$

  w.r.t to $\boldsymbol{\psi}_j$ to obtain an updated value $\boldsymbol{\psi}_j^{(l+1)}$ of the copula parameter ($j = 1, \ldots, K$), where $\boldsymbol{\Theta}_j = f_{\boldsymbol{\Theta}}(\boldsymbol{\psi}_j)$ is the $(p-1) \times (p-1)$ matrix of angles associated with $\boldsymbol{\psi}_j$.

The expression to maximize varies in each of CM-1 and CM-2, however, they still aim to maximize the conditional expectation of the same penalized complete data likelihood function (25). The second summation term in CM-1 is omitted from CM-2 since it is independent of the copula parameter. Similarly, the L_2 penalization term is independent of the marginal parameters and thus not required in CM-1.

The algorithm implemented iterates between the E-step and M-step (M-step 1, CM-step 1 & CM-step 2) until a termination criterion is satisfied. That is, either a pre-defined maximum number of iterations is reached, or in most cases, the relative difference of the incomplete data penalized log-likelihood (24) in two successive iterations is less than 10^{-6}.

5.4 Model selection

Given the observed data, the model selection process aims to automatically select the number of components and adaptively select the best correlation structure for the component distributions using shrinkage methods. This problem reduces to fitting a whole family of mixture models, and selecting the most appropriate using some criterion. The procedure is considered equivalent to `mclust`'s model selection implementation, where instead of finding optimal eigenvalue decomposition of the component distributions' covariance matrices, an optimal value of the tuning parameter is determined.

Suppose $\boldsymbol{K} = (K_1, \ldots, K_e)$, such that $K_s \in \{2, \ldots, 9\}$ $(s = 1, \ldots, e)$, and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_f)$ with $0 = \lambda_1 < \cdots < \lambda_f$. The model selection procedure adopts a grid search methodology, where a RCBMM with density (22) is fitted for each pair $(K_s, \lambda) \in \boldsymbol{K} \times \boldsymbol{\lambda}$ creating a family of $e \times f$ models.

For fixed $K_s \in \boldsymbol{K}$, the most appropriate correlation structure is determined by finding the value of $\lambda \in \boldsymbol{\lambda}$ that corresponds to the model maximizing the mean silhouette width of the observations. The silhouette value associated with an observation is a measure of how similar the data point is to its own cluster compared to other clusters. For a more detailed explanation, see Appendix C. As a result, the mean of the silhouette widths over the entire dataset is a metric describing how appropriately the data is clustered, and choosing the model that maximizes the value of the mean is equivalent to selecting the model that maximizes the separation between the clusters. Hence, the model selection procedure identifies, for each $K_s \in \boldsymbol{K}$, the value of λ that maximizes cluster separation.

This results in a list of pairs $(K_1, \lambda_1^*), \ldots, (K_e, \lambda_e^*)$, such that $\lambda_s^* \in \boldsymbol{\lambda}$ $(s = 1, \ldots, e)$ represents the optimal correlation structure for a given number of mixture components. The final step in the process is to select the pair that optimizes the number of the components in the mixture, which is performed using BIC. The model selection algorithm is therefore:

1. For $s = 1, \ldots, e$:

   (a) Consider the models defined by the pairs $(K_s, \lambda_1), \ldots, (K_s, \lambda_f)$ and select $\lambda_s^* \in \boldsymbol{\lambda}$ as the turning parameter corresponding to the model that achieves the larget average silhouette width.

   (b) Compute $BIC_{(K,M)} = (2 \times loglik_{(K,M)}) - (q \times \log n)$ for the model defined by the pair $(K, M) = (K_s, \lambda_s^*)$, $loglik_{(K,M)}$ is the unpenalized log-likelihood (6) of the model, and q and n are the number of independent parameters in the model and observations in the data, respectively.

2. Select the pair $(K^*, \lambda^*) \in \{(K_1, \lambda_1^*), \ldots, (K_e, \lambda_e^*)\}$ that maximizes $BIC_{(K^*,\lambda^*)}$.

6 Computational aspects

6.1 Initialization

The initialization procedure adopts a semi-stochastic semi-deterministic approach by identifying multiple initial partitions of the data and assessing which set of corresponding parameter estimates should be used as starting values for ECM. The parameter estimates corresponding to each partitioning are obtained by applying the deterministic method offered by Kosmidis and Karlis (2016), *Section 3.2*, which is an application of the IFM method (Joe 1997, *Section 10*). The strategy used also makes use of the practicality offered by `mclust` in enhancing separation amongst groups prior to identifying the partitioning, which aims to ensure reasonable starting values are identified. The strategy for finding starting values corresponding to a given transformation `T`, where `T` corresponds to `SPH`, `PCS`, `PCR` or `SVD` in Sect. 2.5.2, is as follows for the case that no shrinkage is applied to the model ($\lambda = 0$).

1. Retrieve (if already computed for a previous value of K) or determine the results of MBHAC via `hc()`, such that the separation amongst the data is enhanced prior to performing MBHAC using transformation `T`.

2. Identify an initial classification vector by parsing the results of MBHAC and K to `hclass()`, which is another function offered by `mclust`. This partitions the observation indices $\{1, \ldots, n\}$ into K mutually exclusive subsets $S_1, \ldots, S_K$, such that $\cup_{j=1}^{K} S_j = \{1, \ldots, n\}$ and $\left|S_j\right| = N_j$ is the cardinality of the jth subset.

3. Set the starting values for the mixing proportions $\boldsymbol{\pi}$ as $\pi_j^{(0)} = N_j/n$ $(j = 1, \ldots, K)$.

4. Set the starting value $\boldsymbol{\gamma}_{jt}^{(0)}$ of the jth component's tth marginal parameter vector by using maximum likelihood to fit the marginal g_t on data x_{it} for $i \in S_j$ and $t = 1, \ldots, p$ $(j = 1, \ldots, K)$.

5. Set the starting value $\boldsymbol{\psi}_j^{(0)}$ of the jth component's copula parameter by using maximum likelihood to fit the copula $C_j(u_1, \ldots, u_p; \boldsymbol{\psi}_j)$ on observations $u_{it} = G_t(x_{it}; \boldsymbol{\gamma}_{jt}^{(0)})$ for $i \in S_j$ and $t = 1, \ldots, p$ $(j = 1, \ldots, K)$.

6. Compute the corresponding likelihood (24) of the model where the L_2 penalty is insignificant since $\lambda = 0$.

This provides four sets of starting values corresponding to the different transformations applied before performing MBHAC. The starting values associated with the model achieving maximal likelihood are chosen as the starting values for ECM.

For the case $\lambda > 0$, the problem of identifying starting values for the parameter estimates is straightforward. The implementation ensures that given a grid of lambda values $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_f)$, where $0 = \lambda_1 < \cdots < \lambda_f$ without loss of generality, the family of mixture models is fitted sequentially: first on the pair (K, λ_1) through to (K, λ_f). As λ is increased, the previous solution $\boldsymbol{\pi}^*$ and $\boldsymbol{\theta}^*$ is parsed as a "warm start" to ECM for the new value of λ. This approach is intuitive since the parameter estimates that achieve convergence for a previous value of λ are considered the best possible starting values for new λ, since the model corresponding to previous λ is very close the new model in the model space if the interval between the values of λ is not large.

The solution is efficient as it makes use of the convenient MBHAC procedure available in `mclust`, which allows for the identification of a hierarchical clustering that is independent of the number of mixture components. It removes the requirement for performing MBHAC for each value of K when fitting a family of models, which is especially practical in the scenario that the number of observations n or dimension p is large. The transformations are used in an attempt to achieve convergence to a global maximum using ECM, although this may not always be the case.

The resulting procedure is a combination of stochastic *emECM* (a previously undefined procedure considered the extension of *emEM* to ECM) and the deterministic approach used in Kosmidis and Karlis (2016). As Biernacki et al. (2003) describes, there doesn't exist an initialisation strategy that works uniformly well in all cases, however, by defining a procedure that takes advantage of the benefits of both stochastic and deterministic methods, one would hope the initialization procedure would perform well. The performance of the initialization method is not investigated in great detail here since the primary focus of this report is the definition of regularized copula-based mixture models, but is suggested as a topic of interest in future research.

6.2 Conditional-maximization steps

The separable form of the penalized complete data log-likelihood (25) is convenient, since it allows the decomposition of the maximization task into K independent maximizations of weighted likelihoods,

$$\boldsymbol{\theta}_j^{(l+1)} = \underset{\boldsymbol{\theta_j}}{\operatorname{argmax}} \Bigg\{ \Big\{ \sum_{i=1}^{n} z_{ij} \{ \log \pi_j + \log f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) \} \Big\} - \lambda \sum_{\omega \in \boldsymbol{\Theta}_j} (\omega - \frac{\pi}{2})^2 \Bigg\}, \qquad (j = 1, \ldots, K) \tag{29}$$

where the update from $\boldsymbol{\theta}_j^{(l)} = (\boldsymbol{\gamma}_j^{(l)T}, \boldsymbol{\psi}_j^{(l)T})^T$ to $\boldsymbol{\theta}_j^{(l+1)} = (\boldsymbol{\gamma}_j^{(l+1)T}, \boldsymbol{\psi}_j^{(l+1)T})^T$ is achieved by

- *CM-step 1*: updating from $\boldsymbol{\theta}_j^{(l)} = (\boldsymbol{\gamma}_j^{(l)T}, \boldsymbol{\psi}_j^{(l)T})^T$ to $\boldsymbol{\theta}_j^{(\frac{l+1}{2})} = (\boldsymbol{\gamma}_j^{(l+1)T}, \boldsymbol{\psi}_j^{(l)T})^T$ by maximizing (27) w.r.t the marginal parameters $\boldsymbol{\gamma}_j = (\boldsymbol{\gamma}_{j1}^T, \ldots, \boldsymbol{\gamma}_{jp}^T)^T$, followed by,
- *CM-step 2*: updating from $\boldsymbol{\theta}_j^{(\frac{l+1}{2})} = (\boldsymbol{\gamma}_j^{(l+1)T}, \boldsymbol{\psi}_j^{(l)T})^T$ to $\boldsymbol{\theta}_j^{(l+1)} = (\boldsymbol{\gamma}_j^{(l+1)T}, \boldsymbol{\psi}_j^{(l+1)T})^T$ by maximizing (28) w.r.t the copula parameter $\boldsymbol{\psi}_j$.

Thus, the pair CM-1 and CM-2 can be maximized independently across the mixture components to allow for parallel optimization, which reduces computation time significantly in multi-core systems.

6.3 Transformation of parameters

Both of the conditional-maximization steps require numerical methods to optimize some objective function. Unconstrained optimization is achieved by performing a suitable transformation to the parameters prior to optimization, and performing an inverse transformation to the results before updating the parameters of the model.

When performing CM-1, the transformations used are the identity, logarithmic and exponential functions. Prior to optimization, the current estimate of a mixture component's marginal parameter vector $\boldsymbol{\gamma}_j^{(l)} = (\boldsymbol{\gamma}_{j1}^{(l)T}, \ldots, \boldsymbol{\gamma}_{jp}^{(l)T})^T$ is transformed to $\boldsymbol{\gamma}_{*j}^{(l)} = (\boldsymbol{\gamma}_{*j1}^{(l)\ T}, \ldots, \boldsymbol{\gamma}_{*jp}^{(l)\ T})^T$ defined on the whole real line. Expressing this swap formally is hard to follow so the intuition is described instead. All scalar parameters within the marginal parameter vector that are defined on $(0, \infty)$ (for example, α and β parameterizing a $Beta(\alpha, \beta)$ marginal distribution) are exchanged with their corresponding logarithmic value. Scalar parameters defined on the whole real line (for example, μ parameterizing a $N(\mu, \sigma^2)$ marginal) are exchanged with their corresponding identity mapping since they are already considered unconstrained. The component's transformed marginal parameter vector $\boldsymbol{\gamma}_{*j}^{(l)}$ is parsed to the optimization function which outputs an updated value $\boldsymbol{\gamma}_{*j}^{(l+1)}$ for the transformed marginal parameters. The inverse transformations are then applied to the output to ensure the marginal distributions are well-defined. This results in the updated parameter vector $\boldsymbol{\gamma}_j^{(l+1)T}$ which is used by CM-2 to update the implied dependence structure of the component distribution.

When performing CM-2, accommodating for unconstrained optimization is more challenging since the copula parameter $\boldsymbol{\psi}_j$ takes the form of a $p \times p$ correlation matrix constrained on $[-1, 1]$. The problem is simplified by using the reparameterization approach discussed in Sect 4.2. The current value of the

copula parameter is exchanged with an upper-triangular $p \times p$ matrix of angles $\boldsymbol{\Theta}_j^{(l)} = (\theta_{ij}) = f_{\boldsymbol{\Theta}}(\boldsymbol{\psi}_j^{(l)})$ unconstrained on $(0, \pi]$. Finally, unconstrained optimization on the whole real line is managed by creating a temporary matrix $\boldsymbol{\Theta}_{*j}^{(l)} = (\theta_{*ij})$, such that $\theta_{*ij} = \tan((\theta_{ij} + \pi)/2)$, which is parsed to the optimizer to obtain the updated value of the transformed copula parameter $\boldsymbol{\Theta}_{*j}^{(l+1)}$. Similarly to CM-1, the output is transformed back by its inverse function to attain an updated parameter $\boldsymbol{\Theta}_j^{(l+1)}$. Finally, the upper-triangular matrix of angles is transformed to a correlation matrix by performing the update $\boldsymbol{\psi}_j^{(l+1)} = f_{\boldsymbol{\Theta}}^{-1}(\boldsymbol{\Theta}_j^{(l+1)})$.

6.4 Numerical issues

When performing the conditional-maximization steps of ECM, it is possible to encounter numerical issues that result in likelihood values taking unbounded negative values.

Suppose an observation $\boldsymbol{x}_i$ is extremely unlikely to belong to the jth cluster, then the density $f(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ takes value close to 0. Hence, $\log f(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ and the summations (27) and (28) can take unbounded negative values. This can produce unstable results during optimization, resulting in the inability to determine a solution to the optimization problem at hand. To overcome this issue, observations with a small probability of belonging to the jth cluster, precisely $z_{ij} < 10^{-16}$, are omitted from the summations (27) and (28).

Moreover, evaluating the density $\phi_p(...)$ in (27) and (28) results in a likelihood value equal to 0, which causes $\log \phi_p(...)$ to take unbounded values, if $u_{it} = G_t(x_{it}; \boldsymbol{\gamma}_{jt}) \in \{0, 1\}$ for any $t = 1, \ldots, p$. This problem is alleviated by setting

$$u_{it} = \begin{cases} 0.999, u_{it} > 0.999 \\ 0.001, u_{it} < 0.001 \end{cases}$$

prior to computing the density $\phi_p(...)$. This modification ensures the stability of the optimization procedures that perform the conditional-maximization steps of ECM.

Finally, a threshold is applied to *Beta* marginal distributions to avoid unbounded likelihood values resulting from clusters containing a single observation or clusters containing multiple observations with equal value. The variance for a *Beta* marginal distribution of a mixture component is forced to be greater than 10^{-5}.

7 Application

7.1 Simulated data

The foregoing theory is illustrated by the analysis of a simulated dataset. Since the shrinkage methods (see Sect. 5.2) adopted during model fitting encourage the copula parameter parameterizing the component distributions to approach zero correlation, one would expect the geometric features of the clusters to change as λ is increased. It is well known that the k-means algorithm is a special case of Gaussian mixture models with all component distributions parameterized by covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{I}_p$, where p is the dimension of the data, resulting in K equal-volume spherical clusters. The intuitive correspondence between the identity covariance matrix and identity correlation matrix implies components in a mixture of regularized Gaussian-copula will occupy a more spherical region of the sample space as λ is increased and the copula parameter approaches zero correlation.

This hypothesis is investigated by generating data from $K = 2$ distinct clusters, with dimension $p = 2$ for the purposes of visualization. The mixing proportions π_1 and π_2 are equal to $2/3$ and $1/3$, respectively, and the data is produced for $n = 20, 50, 150, 375, 2000$ observations. The marginal distributions G_1 and G_2 respectively take $Normal(\mu, \sigma^2)$ and $Beta(\alpha, \beta)$ distribution. Hence, the parameters of the jth component in the mixture are $\boldsymbol{\gamma}_j = (\mu_j, \sigma_j, \alpha_j, \beta_j)^T$ for the marginal distributions and $\boldsymbol{\psi}_j = \rho_j$ for the copula parameter. A model taking density (22) is estimated, for each simulated dataset corresponding to $n = 20, 50, 150, 375, 2000$, via ECM with tuning parameter $\lambda = 0, 100, 200$ and $K = 2$ constrained to the true number of clusters in the data.

The contours in Fig. 1 demonstrate clusters become more spherical as λ increases, as expected. Moreover, the clusters' orientation becomes bias towards the coordinate axis as λ increases as a consequence of the copula parameter approaching zero correlation. The effect on parameter estimates and classification error of this changing shape and orientation is investigated.

Define,

$$RMSE(\boldsymbol{\gamma}) = \sqrt{\frac{1}{2}\sum_{j=1}^{2}\frac{1}{4}\sum_{t=1}^{4}\Big(\frac{(\boldsymbol{\gamma}_j)_t - (\boldsymbol{\gamma}_j^*)_t}{(\boldsymbol{\gamma}_j^*)_t}\Big)^2}, \qquad RMSE(\boldsymbol{\psi}) = \sqrt{\frac{1}{2}\sum_{j=1}^{2}\Big(\frac{\rho_j - \rho_j^*}{\rho_j^*}\Big)^2} \tag{30}$$

as a measure of the relative error associated with the estimate of the marginal and copula parameters, respectively, where $\boldsymbol{\gamma}_j^*$ and ρ_j^* represent the true value of their associated parameters for the jth component.

Table 1 demonstrates a relationship between $RMSE(\boldsymbol{\gamma})$ and $RMSE(\boldsymbol{\psi})$. The $RMSE(\boldsymbol{\psi})$ is listed first for each λ, as it is considered the primary dependent variable of the study due to the application of the shrinkage directly on $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$. As expected, $RMSE(\boldsymbol{\psi})$ increases as the magnitude of shrinkage is increased and $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ approach zero correlation. More interestingly, the secondary dependent variable $RMSE(\boldsymbol{\gamma})$ also increases with $RMSE(\boldsymbol{\psi})$ and λ. The report so far discusses the ability when using a copula function to model multivariate data to first select the marginal properties and then capture the distribution's implied dependence with the copula. This suggests that the value of a component distribution's copula parameter $\boldsymbol{\psi}_j$ depends on its marginal parameters $\boldsymbol{\gamma}_{j1}, \ldots, \boldsymbol{\gamma}_{jp}$, however, the contrary is not considered. The proposition introduced by Table 1 that increasing λ can increase the error in the estimate of the marginal parameters is explained by analysing the likelihood function (27) maximized by CM-1. The updated values of the marginal parameters are dependent on the copula function C_j parameterized by $\boldsymbol{\psi}_j$, which means applying shrinkage to the copula parameter in CM-2 during the lth iteration of ECM affects the estimate of the marginal parameters in the $(l+1)$th iteration.

A more in-depth analysis of Table 1 finds the number of observations affects the error in parameter estimates. For $\lambda = 200$, $RMSE(\boldsymbol{\psi})$ is equal to 0.9999 and 0.1565 respectively for $n = 20$ and $n = 2000$. The effect of n on the errors is explained by considering the model's penalized log-likelihood (24). The L_2 penalty in the second parenthesis is independent of n, therefore, increasing n results in the non-penalized likelihood given by the first parenthesis dominating the likelihood value. Considering the results, it is concluded the number of observations in the sample data should be considered when constructing a grid of λ values. If the number of observations in the data is large, a grid of larger λ values is suggested, and vice versa. A suggestion for future research is to investigate this relationship further, and identify a method of constructing a reasonable tuning grid prior to constructing the family of models.

It can be expected that an increasing error in the estimates of the parameters and changing cluster shapes has an effect on the classification implied by the mixture model. The maximum of the posterior probabilities of membership $z_{i1}, \ldots, z_{iK}$ at the last iteration of the ECM algorithm (see E-step in Sect. 5.3) is used to determine the cluster membership of each observation. Figure 1 shows the classification error is affected by the changing dependence structure caused by increasing λ.

This simulated example is included primarily to demonstrate the effect of shrinkage methods on resulting cluster shapes. An increasing value of the tuning parameter λ will consistently encourage the mixture's clusters to become more spherical and bias their orientation towards the coordinate axis. However, the large increase in misclassification error in Fig. 1 is not associated with an increasing value of λ, rather with the values of λ used. The shrinkage applied to the model's dependence structure is considered extreme due to the large values of λ with respect to the number of observations $n = 375$.

When the same dataset ($n = 375$) is modelled with $K = 2$ components using a new family of regularized copula-based mixture models constructed using a more appropriate grid of lambda values $\lambda = 0, 1, \ldots, 50$, the model selection procedure (see Sect 5.4) determines that $\lambda = 17$ produces the best correlation structure of the component distributions. The corresponding model results in a misclassification error of only 17.6%, which is less than the error produced in the scenario that no shrinkage is applied to the model, as shown by the case $\lambda = 0$ in Fig 1. Hence, it is concluded the RCBMM outperforms a non-regularized mixture of copula in terms of clustering performance in this example.

Fitting a Gaussian mixture model using `mclust`, with $K = 2$ restricted to the true number of clusters, to the simulated dataset ($n = 375$), yields a BIC value of -1447.8. This model is also constrained to a "VVV" parameterization of the component distribution's covariance matrix, allowing for comparison to the optimal RCBMM $(K, \lambda) = (2, 17)$ whose components are also free to vary in volume, shape and orientation across groups. Interestingly, the model produced by `mclust` results in a misclassification error of 18.7% meaning the RCBMM outperforms `mclust` in terms of clustering performance and BIC, since it achieves a BIC value -1063.7, which is not a guaranteed phenomenon in model-based clustering. The classification table of each model can be viewed in Tab. 2. In summary, in this example the RCBMM outperforms a non-regularized copula-based mixture and Gaussian mixture model in terms of clustering performance and BIC.

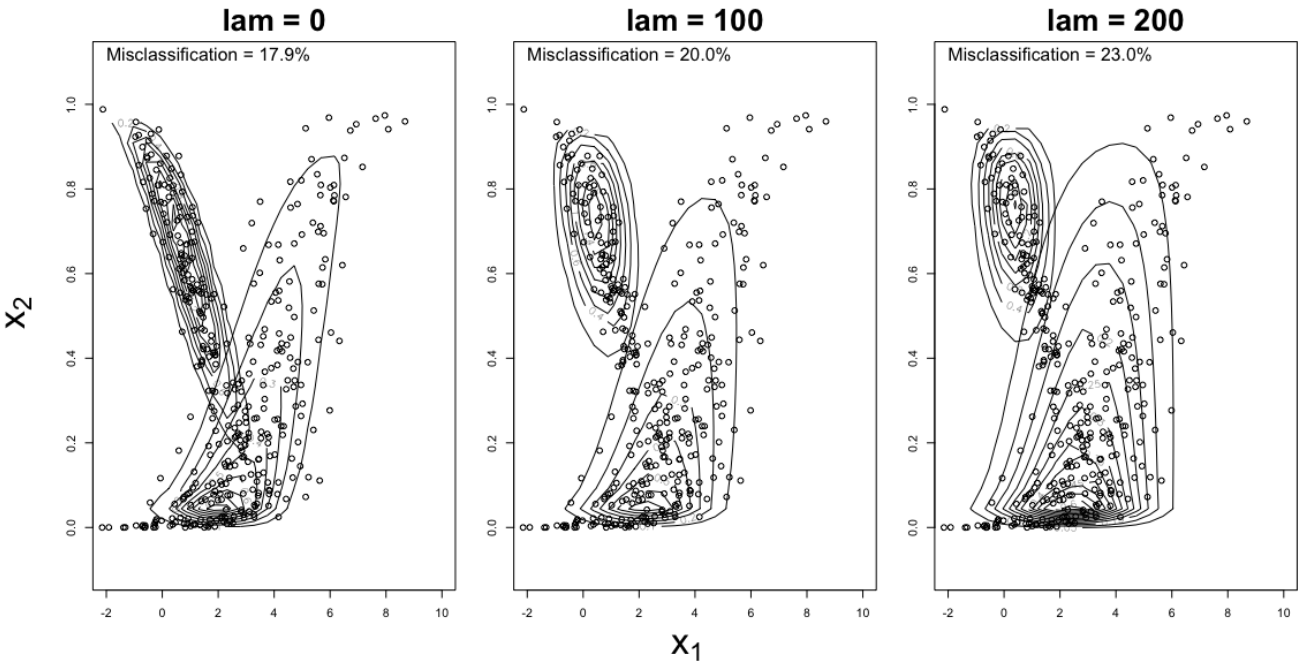

Figure 1: The contours of the mixture components corresponding to the models $(K, \lambda) = (2, 0), (2, 100), (2, 200)$ for the simulated dataset with $n = 375$.

	$\lambda = 0$		$\lambda = 100$		$\lambda = 200$	
n	$RMSE(\boldsymbol{\psi})$	$RMSE(\boldsymbol{\gamma})$	$RMSE(\boldsymbol{\psi})$	$RMSE(\boldsymbol{\gamma})$	$RMSE(\boldsymbol{\psi})$	$RMSE(\boldsymbol{\gamma})$
20	0.0920	0.5449	0.9999	26.3010	0.9999	26.3011
50	0.0401	0.3395	0.9110	0.7247	0.9538	0.7397
150	0.0050	0.1051	0.8155	1.7114	0.8973	1.8893
375	0.0200	0.0844	0.4672	0.4332	0.7021	0.4778
2000	0.0120	0.0385	0.0556	0.1398	0.1565	0.3459

Table 1: Error in parameter estimates corresponding to the models $(K, \lambda) = (2, 0), (2, 100), (2, 200)$ for the simulated dataset with varying numbers of observations n.

The investigation thus far is in a supervised setting, where the number of components in the mixture is known. In the unsupervised setting $K = 2, \ldots, 9$, a family of RCBMMs with a tuning grid $\lambda = 0, 1, \ldots, 50$ and a Gaussian mixture produced by `mclust` with parameterization "VVV" is constructed, to investigate the ability of both approaches to select the optimal number of mixture components. The RCBMM made correct inference regarding the number of components while `mclust` yielded 4 mixture components. It is concluded that for the simulated data example the RCBMM outperforms `mclust` in terms of clustering performance, BIC and making correct inference regarding number of clusters.

7.2 Real data

The exposition of the methodology is applied to a real dataset by extending the analysis of Kosmidis and Karlis (2016) on a sport-related dataset from `Hoopdata.com`, which is a website that provides an extensive database for NBA statistics. The data contains statistics on 493 NBA players that played more than 24 minutes (the equivalent of half a game) in the 2011-2012 season. The paper's investigation saw the grouping of players in terms of their performance by fitting a family of Gaussian copula-based mixtures with copula parameters represented as exchangeable and unstructured correlation matrices. With the introduction of regularization, this analysis extends the ideology by creating a family of mixtures ranging from an unstructured dependence structure (when $\lambda = 0$) to near exchangeable (since a matrix with zero correlation can also be described as exchangeable) as λ becomes large, and a range of intermediate fits by increasing λ gradually.

The data contains 6 performance-related variables in $(0, 1)$, each representing a percentage attribute,

	`mclust` classification	
True cluster	1	2
1	197	67
2	3	108

	rcbmm classification	
True cluster	1	2
1	194	60
2	6	115

Table 2: Classification tables for the Gaussian mixture model with "VVV" parameterization fitted via `Mclust()` and optimal RCBMM given by $\lambda = 17$ when applied to the simulated data with $n = 375$ observations. The number of components $K = 2$ is restricted in both examples.

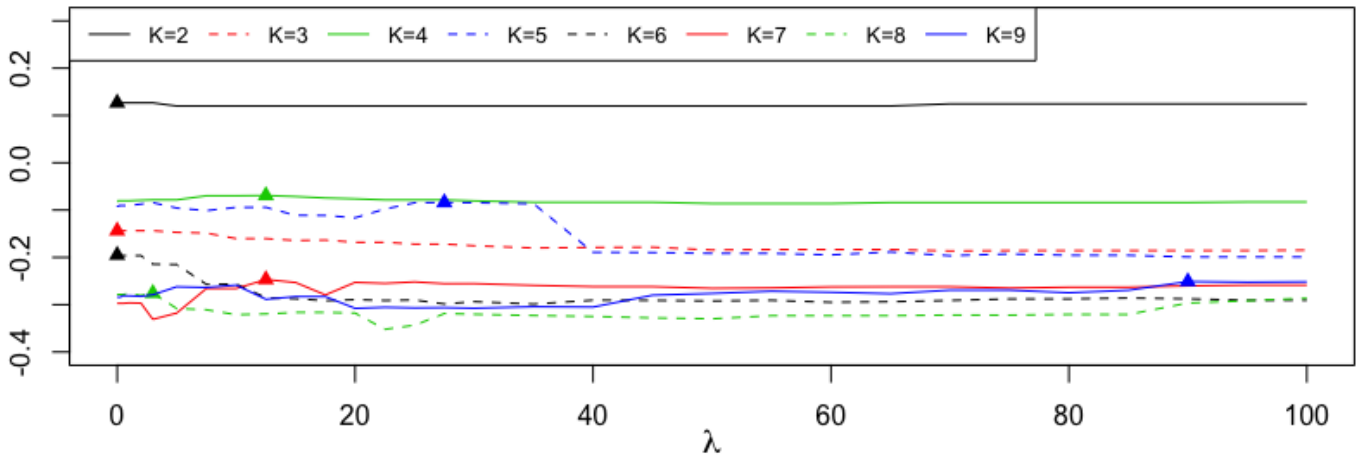

Figure 2: For $K = 2, \ldots, 9$, the mean silhouette width of model fitted to the NBA data according to density (22) varies depending on the magnitude of shrinkage controlled by λ. The maximum silhouette width, which corresponds to optimal correlation structure, is indicated for each value of K.

and a final variable representing the total points scored with all values positive. Appropriate marginals are the *Beta* and *Gamma* distribution for representing the percentage attributes and points scored, respectively. For a detailed description of the data see Kosmidis and Karlis (2016), *Example 4.2*.

For $K \in \boldsymbol{K} = \{2, \ldots, 9\}$ and $\lambda \in \boldsymbol{\lambda} = \{0, 0.1, 0.25, 0.5, 1, 2, 3, 5, 7.5, 10, \ldots, 30, 35, 40, 50, \ldots, 100\}$, density (22) is fitted to the data by applying ECM to maximize the penalized likelihood (24). This results in $|\boldsymbol{\lambda}| = 32$ different parameterizations of the copula parameters for each $K \in \boldsymbol{K}$, thereby constructing a family of 256 models. Restrictions are applied to constrain the variance of each *Beta* marginal to be greater than 10^{-5} to avoid unbounded likelihood values (see, Sect. 6.4).

For a fixed K, the model with optimal correlation structure is found by picking $\lambda \in \boldsymbol{\lambda}$ that maximizes the average silhouette width of the observations. In the case that adjacent values of λ in the tuning grid result in an equal mean silhouette width, which occurs when both models predict an identical classification of the observations, the smallest such value of λ is chosen. Fig 2 shows the mean silhouette values for increasing λ, and highlights the chosen value of λ for each K.

After finding the best correlation structure for each number of mixture components, the model selection problem is reduced to finding the optimal value of K using BIC. Table 3 outlines that $K = 4$ provides the largest value of BIC, indicating the optimal model from the family of mixtures fitted to the NBA data is given by $(K^*, \lambda^*) = (4, 12.5)$.

To visualize the effect of the shrinkage-driven method on the copula parameters, the $\frac{1}{2}p(p-1) = 21$ values in [-1,1] parameterizing a mixture component's copula parameter are plotted against λ. Fig 3 demonstrates the elements of each $\boldsymbol{\psi}_j$ approach 0 as the magnitude of shrinkage increases for the case $K = 4$, and the dashed vertical line located at $\lambda = 12.5$ highlights the optimal correlation structure identified previously. For the case of the first component in the mixture, the same data can be observed in matrix form in Fig 4 for $\lambda = 0$ (unstructured) and $\lambda = 100$ (maximal shrinkage applied). The results are as expected since the magnitude of L_2 regularization applied to each $\boldsymbol{\psi}_j$ in (24) becomes larger as λ increases, thereby encouraging the $\frac{1}{2}p(p-1)$ parameters of each matrix to approach 0.

Further investigation into the effect of the shrinkage on the copula parameters is performed by investigating the determinant of the correlation matrix parameterizing a particular mixture component

K	Optimal λ	q	Log-lik	BIC	q	Log-lik	BIC	q	Log-lik	BIC
		Shrinkage-driven correlation			Exchangeable correlation			Unstructured correlation		
2	0	71	3521.49	6602.75	31	3146.45	6100.68	71	3491.12	6542.00
3	0	107	4007.53	7351.61	47	3734.31	7177.2	107	3990.00	7316.55
4	12.5	143	4142.58	7398.49 (***)	63	3900.49	7410.35	143	4119.54	7352.41(*)
5	27.5	179	4157.81	7205.73	79	4002.73	7515.62	179	4218.24	7326.59
6	0	215	4261.15	7189.20	95	4053.61	7518.17(**)	215	4267.97	7202.83
7	12.5	251	4332.84	7109.35	111	4073.36	7458.46	251	4270.47	6984.61
8	3	287	4348.23	6916.91	127	4146.57	7505.68	287	4365.99	6952.43
9	90	323	4337.50	6672.24	143	4159.01	7431.35	323	4387.43	6772.1

Table 3: Maximum likelihood fits of the density (22) on the NBA dataset with best shrinkage-driven correlation structure for $K = 2, \dots, 9$ (left) and exchangeable and unstructured correlation structure presented by Kosmidis and Karlis (2016) (right). A (*) denotes the best BIC for each correlation specification, (**) denotes the best BIC found by Kosmidis and Karlis (2016) and (***) denotes the best BIC found by the shrinkage-driven investigation. Log-lik represents the non-penalized likelihood (6) in all cases. Optimal λ is understood by analysing Fig. 2.

as λ increases. Since the copula parameters approach zero correlation, the determinant of the correlation matrix should approach $\det(\boldsymbol{I}_p) = 1$. Fig 5 shows the result is as expected with the exception of 1 mixture component in the case $K = 9$ where the shrinkage is not yet apparent.

The results of the investigation can be compared to those of Kosmidis and Karlis (2016) by analysing Tab. 3. The case such that the optimal shrinkage-driven correlation structure is given by the model experiencing no shrinkage ($K = 2, 3, 6$), the BIC value can be compared directly to the value achieved by the model with unstructured correlation. A stochastic approach to determining starting values and a more stringent criterion for assessing convergence causes the unstructured BIC results in Kosmidis and Karlis (2016) to be similar but not equal to the results of this investigation. Interestingly, the introduction of shrinkage methods to maximize separation between clusters results in different inference regarding the optimal number of mixture components K. However, the result that optimal $K = 4$ in this study is also achieved for unstructured correlation in Kosmidis and Karlis (2016), and hence the BIC values of these results are discussed. The variance of maximum likelihood parameter estimates caused by varying magnitudes of shrinkage, as demonstrated in Section 7.1, causes the unpenalized log-likelihood of any regularized copula-based mixture to decrease in value as λ increases. Hence, the greater BIC value 7398.49 for optimal shrinkage-driven model in this investigation compared with value 7352.41 for $K = 4$ with no shrinkage arises from the differing starting values rather than the shrinkage-method. However, this result does not imply the semi-deterministic semi-stochastic initialization approach used here is more successful than the random approach adopted by Kosmidis and Karlis (2016), as the model defined by (K, λ)=$(6, 0)$ achieved a lower BIC than the unstructured results of the previous study. This is evidence of the previous discussion that there is no "optimal" approach to identifying starting values prior to performing EM or ECM (see Sect. 2.3).

To conclude the investigation, the bivariate marginals of the best model $(K, \lambda) = (4, 12.5)$ are studied. It was remarked earlier that the desirable closure under marginalization property for Gaussian mixture models, which ensures that the marginal of any dimension of the component distributions arises from the same family of distribution as the full component distribution itself, contributed to the popularity of modelling component distributions according to the multivariate normal distribution. Fang et al. (2002) demonstrates this property is satisfied for the Gaussian copula, and hence, the contours of the bivariate marginals of optimal model $(K, \lambda) = (4, 12.5)$ are shown by Fig. 6.

8 Software: `rcbmm`

This chapter aims to describe the implementation that facilities the newly defined model and methods discussed in Sect. 5 & 6. It includes a description of how various readily available tools aided development, a flowchart to explain the relationship between components within the system and a comprehensive documentation of each function associated with the implementation. A description of the methodology adopted towards software development is omitted as it is included in Sect. 9.1. The software is available at `https://github.com/ben-j-barlow/rcbmm`.

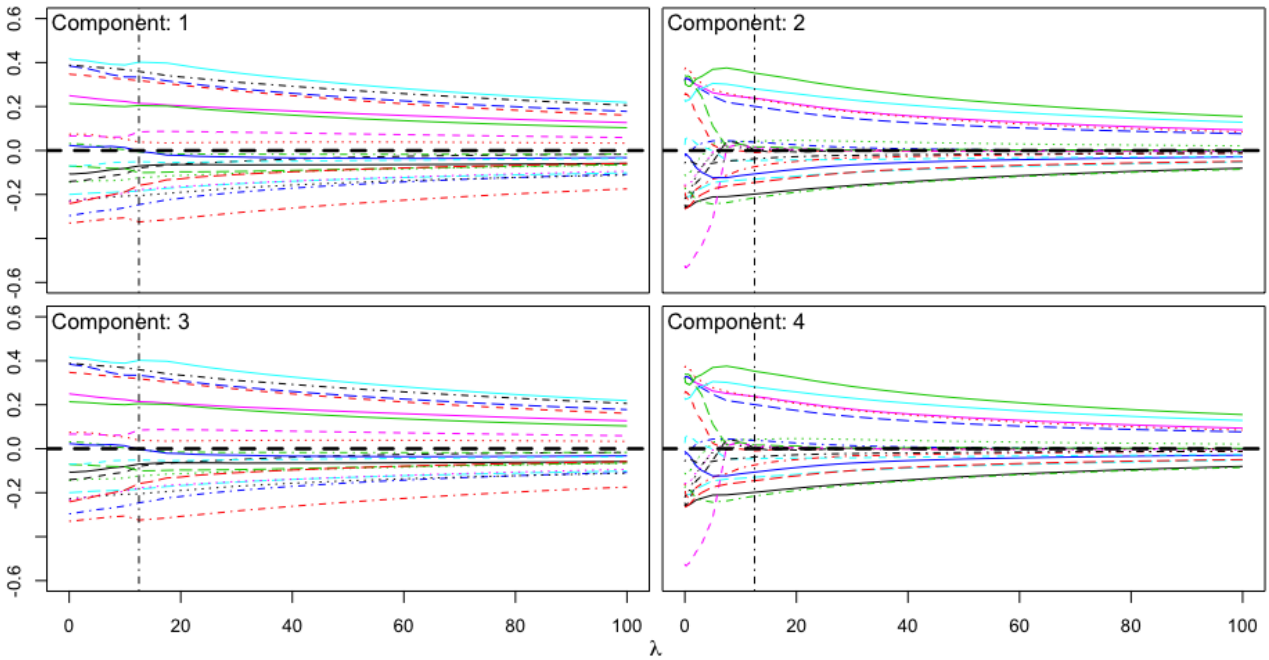

Figure 3: Differing correlation structure arising from an increasing magnitude of shrinkage for model with density (22) and $K = 4$ mixture components. A dashed vertical line is located at $\lambda = 12.5$ indicating the optimal correlation structure of the model.

$$\begin{pmatrix} -0.106 & 0.348 & -0.144 & -0.296 & -0.200 & 0.250 \\ & -0.141 & 0.074 & 0.031 & 0.385 & 0.417 \\ & & 0.065 & -0.228 & -0.33 & -0.07 \\ & & & 0.023 & -0.071 & -0.232 \\ & & & & 0.388 & -0.24 \\ & & & & & 0.214 \end{pmatrix} \quad \begin{pmatrix} -0.059 & 0.161 & -0.067 & -0.108 & -0.102 & 0.127 \\ & -0.017 & 0.033 & -0.017 & 0.178 & 0.219 \\ & & 0.059 & -0.112 & -0.175 & -0.064 \\ & & & -0.032 & -0.037 & -0.094 \\ & & & & 0.206 & -0.059 \\ & & & & & 0.103 \end{pmatrix}$$

Figure 4: Value of copula parameter ψ_1 of mixture component with index 1 for the case $K = 4$ in the setting no shrinkage is applied (left) and shrinkage is applied with $\lambda = 100$ (right).

8.1 Discussion of tools and frameworks used

The intuitive choice for implementation was R or Python since they are state-of-the-art in terms of programming language oriented towards data science. The choice of R was logical, due to the project leader's familiarity with the programming language amongst other immediate benefits.

The prominent advantage offered by R is the comprehensive integrated development environment (IDE) RStudio (RStudio Team, 2019). The IDE offers smart indentation, integrated R help and documentation, integrated capabilities for visualizing plots, and extensive package development tools which aided the production and documentation of the project's resulting package.

Moreover, R was reinforced as the intuitive choice since a number of pre-existing R packages that aided the implementation were identified in the early stages of the project. The two packages that proved most beneficial were `mclust` (Scrucca et al., 2016), for obtaining starting values for estimation of copula-based models via ECM, and `copula` (Hofert et al., 2014), for modelling mixture components with a multivariate distribution defined using a copula function.

The `copula` package offered two classes that were important to modelling mixture components. The `copula` class within the package allowed for modelling the dependence of mixture components using an unstructured correlation matrix taking values in $[-1.1]$, whilst the `mvdc` class allowed the modelling of a multivariate distribution with proper marginal distributions with implied dependence captured by an associated `copula` object. Aside from the convenience offered in modelling mixture components, the package provided the ability to visualize density contours of mixture models as well as a formal method for parameter estimation of copulas when obtaining starting values from observation data.

Despite the limitations of `mclust` as a means for constructing flexible mixture models, the package

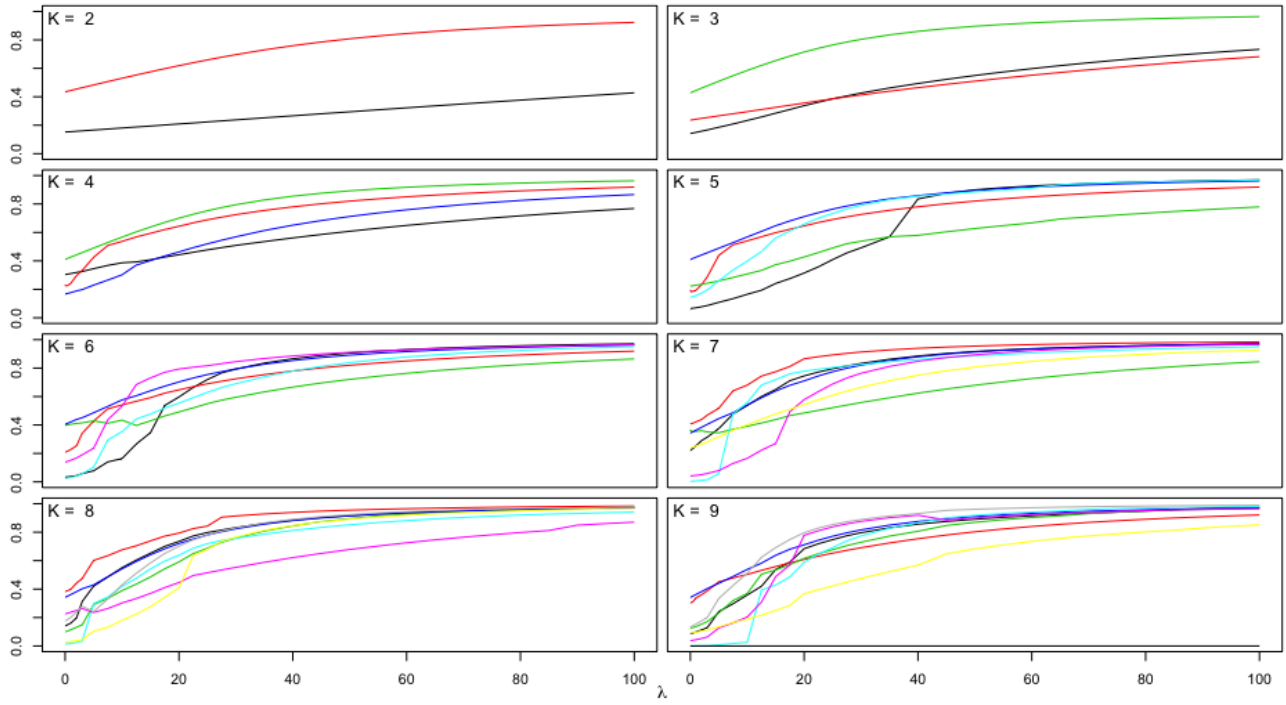

Figure 5: For $K = 2, \ldots, 9$, model defined by density (22) with optimal correlation structure is considered. The determinant of the copula parameters of each model ($\det(\boldsymbol{\psi}_j)$ for $j = 1, \ldots, K$) is plotted and shown to approach $\det(\boldsymbol{I}_p) = 1$, where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix.

contains a number of other functions that were extremely helpful during the project's development. The package's function `hc()` is used to achieve MBHAC results on the observation data that are independent of the value of K. This function also facilitates the transformations previously discussed to enhance the separation amongst groups in the data. The output is stored when fitting a family of models for varying K, and the results are parsed alongside a new value of K to `mclust`'s function `hclass()` to determine an initial partitioning of the data into K groups.

Without the availability of the methods and classes described in `mclust` and `copula`, the project's initialization procedure would have suffered from the shortcomings discussed in Sect. 2.5.2 and the modelling of mixture components would have required more attention. The latter could have increased the workload of the project considerably, which may have resulted in some functionality of the project's output `rcbmm` being lost due to the project's fixed time scope.

Other packages that were less fundamental yet still beneficial to the project's progress were `cluster` (Maechler et al., 2013), `fitdistrplus` (Delignette-Muller et al., 2019) and `parallel`. Firstly, `cluster` offered a framework for computing the silhouette width of each observation after finding a clustering complex with ECM. This allows the best correlation structure to be determined for a fixed number of components by comparing the average silhouette width of the fits arising from varying the value of the tuning parameter. The remaining packages `parallel` and `fitdistrplus` were used to allow for parallel computing and for obtaining estimates for marginal parameters using maximum likelihood, respectively. In the absence of the three packages discussed here, a method for computing silhouette widths could have been written by the project leader in a reasonable time frame, another approach could have been adopted to initializing marginal parameters and conditional-maximization steps could have been performed serially. All of these changes would have been undesirable but not catastrophic to the project's aims, hence, it is concluded that these frameworks were helpful but not essential to the project's progress.

8.2 Explanation of system

The output of the development is an R package named `rcbmm`, which is the first general ECM implementation for copula-based mixture models in the R software. The software can be found at...

The fundamental functions in the system are `fit.rcbmm()`, `ECM.Algorithm()` and `initialize.ECM()`,

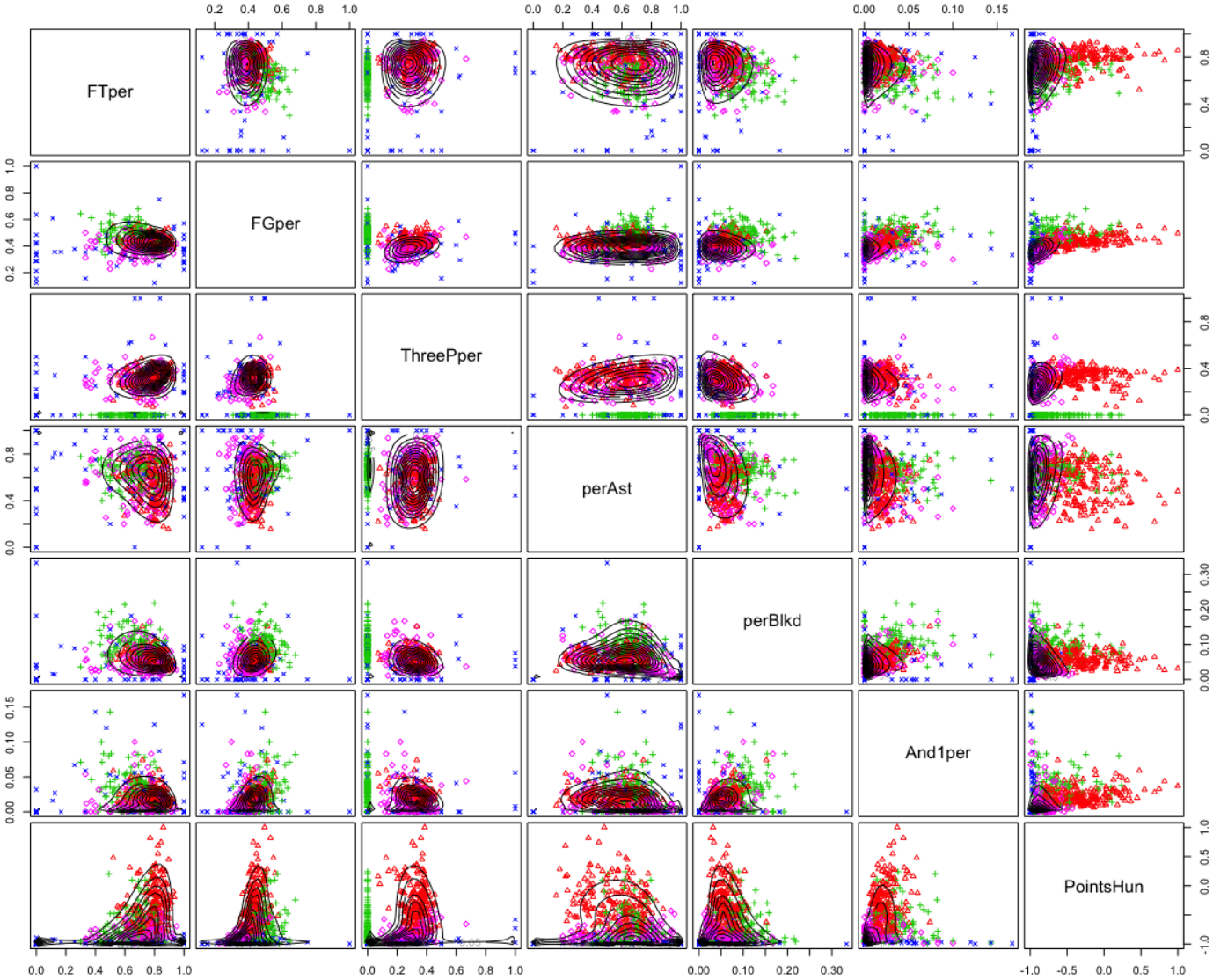

Figure 6: The contours of the bivariate marginal densities of the model with density (22) defined by $(K, \lambda) = (4, 12.5)$ in Sect. 7.2. The observations are coloured according to their cluster assignments.

which respectively estimates a family of models and performs model selection; estimates a regularized mixture of Gaussian copula for a fixed value of K and λ given starting values; and computes starting values given a fixed value of K. All of these functions are also dependent on some observation data and prescribed marginal distributions supplied by the user. Fig 7 provides a flowchart to demonstrate the approach `fit.rcbmm()` takes to model selection and the relationship between the functions.

A detailed explanation of the relationships between the non-fundamental functions in the package is omitted, however, a description of each function, its arguments and its output can be found in the software documentation (see Appendix D).

8.3 Evaluation

The software produced facilitates a new flexible family of mixture models, thereby contributing to the field of model-based clustering. Moreover, the implementation of ECM is the first general ECM implementation for copula-based mixtures. The development of the software is therefore considered a successful project. Any sub-standard programming practices were identified and corrected during the early stages of the project with the assistance of the project supervisor, which ensured the final software incorporated good coding practices throughout. Also, the part agile methodology (see Sect. 9.1) was beneficial as it

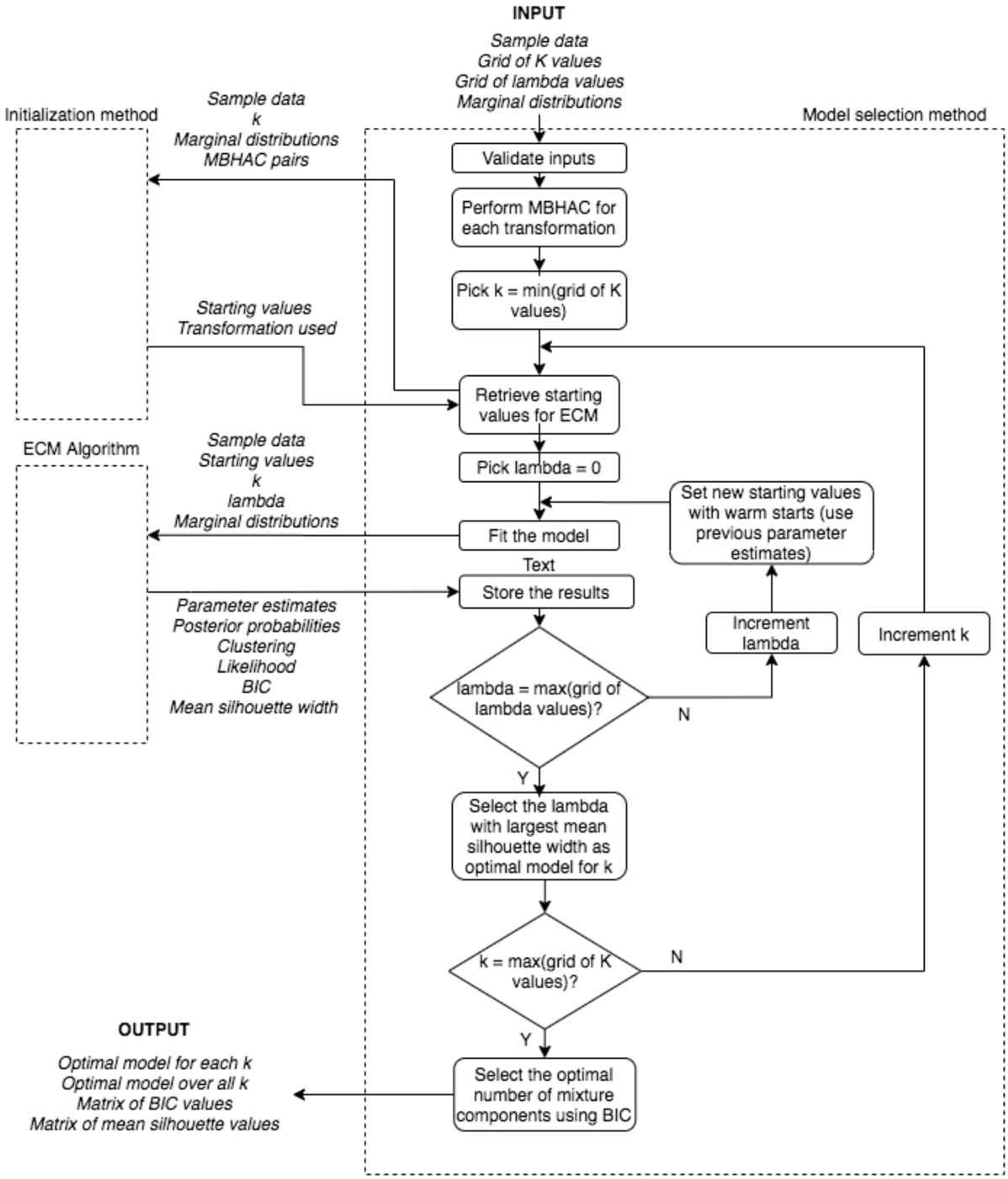

Figure 7: A flowchart to demonstrate the approach taken by `fit.rcbmm()`.

allowed for changing requirements. Any limitations and suggestions for future development are included in Section 10.4.

9 Project management

In order to research and understand challenging statistical concepts, achieve the goal of development and overcome unexpected challenges, the project was broken down into multiple tasks. This section discusses the milestones of the project, how it was managed and professional issues arising from the work completed.

9.1 Methodology

Initially, the project was broken down into four phases: *research*, *development*, *comparison & testing* and *report writing*.

The approach towards the *research* stage was simple yet efficient. A small bibliography was identified with the assistance of the project supervisor, and progress was made by examining the literature followed by creating written notes on the topics. This allowed the project leader to consider the material in great detail, ensuring the fundamental concepts of the project were interpreted and understood at this early stage. In the case that a subject found in the literature was challenging, personal notes were highlighted with a green question mark allowing the answer to enquiries to be found during the next supervision session. This approach allowed statistically challenging concepts to be understood relatively quickly.

A sub-objective of the *research* stage was the development of the full EM algorithm from scratch. Whilst the implementation served no purpose to the long term project goals, the process of 'learning by doing' ensured theory related to the EM algorithm and GMMs was fully understood. This was effective, since the exercise highlighted some misunderstanding of the convergence of the EM algorithm which had not been identified by simply reading the literature. The assistance of the project supervisor helped overcome this issue, which ensured the same problem did not arise during the implementation of the ECM algorithm.

A mixture of plan-driven and agile was adopted during the *development* stage of the project. The project deadline meant a long-term plan with various development goals had to be used to ensure target completion date was met, whilst a part agile methodology was needed to adapt to changing requirements and varying workload from other modules. An underestimation of the problems faced by numerical issues during the fitting of various mixture models to the NBA dataset meant development was extended, however, this did not have a significant effect on the project's progress since the fundamental concepts of the ECM algorithm were implemented by week 5 of term 2. Moreover, a 2 week buffer built into the plan meant there was sufficient time to extend the development efforts whilst still preparing adequately for the 2nd deliverable (project presentation) in week 10 of term 2.

A part agile approach was implemented by using a methodology similar to that of SCRUM. Weekly meetings with the project supervisor, which are considered equivalent to a sprint planning meeting if using SCRUM methodology, allowed the review of goals from the previous week, the identification of goals for the forthcoming week and continuous feedback on coding strategies. It must be noted that due to the responsibility of the project's progress falling to an individual rather than a team, the requirement for a daily SCRUM meeting was dropped. The weekly in-person assistance, coupled with intermediate email communications when necessary, allowed the development of the project's algorithms at a rapid rate during the first half of term 1. This eased the pressure from week 6 onwards, ensuring the remaining goals of the project could be overcome in a manageable way.

The concluding stages of the project *tesing & comparison* and *report writing* were completed mostly independently. The first of which was completed by storing the results in `.RData` and `R` files allowing results to be reproduced. The full *report writing* process was aided by a detailed review of two chapters of the report by the project supervisor, thus enabling the development of a writing style that was maintained for the remaining chapters. This was advantageous as it allowed the project leader to learn from a writer with experience in terms of producing statistical literature, thereby improving the quality of the report. Supervision sessions were relaxed to a fortnightly basis as less assistance was required at this stage due to a decrease in new challenges encountered. The benefits of these meetings, which were held via Skype due to the challenges presented by the COVID-19 pandemic, were maximised by preparing questions in advance and using the share screen capabilities of Skype for the project supervisor to read written questions and corresponding code when necessary.

A method used at more than one stage of the project was the strategy behind development of the EM algorithm in the *resarch* stage and ECM algorithm in the *development* stage. For both tasks, a prototype was produced that could cater for $K = 2$ mixture components and data with dimension $p = 2$. In the case of ECM, the prototype was designed to facilitate only normal marginal distributions when modelling mixture components using copulas. The prototypes were extended after seeking advice from the project supervisor upon completion, ensuring any poor coding practices were identified and corrected before writing the fully functional software.

9.2 Timeline

The project was formalised by a detailed specification on October 9, 2019, and concluded by the completion of the report on January 13, 2025. This 7 month period can be divided into Term 1 and the

Christmas break, which were used for research purposes, Term 2, which had a strong focus towards software development, and the Easter break, which aimed to conclude the software development and saw the production of the final report.

It is common for a long-term project of this nature to have associated with it a Gantt chart. However, due to the lack of understanding of the concepts behind development prior to completing the *research* stage, a Gantt chart could not be included in the specification. Instead, the project was divided into the four stages as previously discussed. A goal of the *research* stage, which was planned under the advice of the project supervisor, was to spend weeks 3-7 of term 1 understanding the initial bibliography (mclust 5, Kosmodis, Meng & Rubin). The allocation of approximately a month to this task was sensible, since it was realistic and provided a buffer of a week before the deadline of the progress report in the case that research progress was hindered. During this period, intense workloads from other modules; commitments as the 1st team captain of The University of Warwick's football club; and a small period of illness, meant the planned buffer was needed and an additional week of work was necessary to complete the goal. Good time management at this stage ensured the 1st deliverable (the progress report) was completed by the deadline.

The final goal of the *research* stage was to understand the method of reparameterizing correlation matrices, discussed in Tsay and Pourahmadi (2017), during the Christmas break. This task was less demanding in terms of time constraints, and was completed prior to term 2 without any delays. Throughout the *research* stage, time was balanced on a weekly basis between reading new literature and implementing algorithms (methods used by EM), so that the supervision sessions could be used to answer questions on both topics. The monitoring of the *research* stage with small goals proved helpful, as it ensured motivation was maintained at all times.

At the time of writing the progress report in week 8 of term 1, a deeper understanding of the project had been learned, and hence, the progress report highlighted a plan for development during term 2. The part agile methodology ensured the project could cater for changing requirements. In fact, the requirement for the implementation of the project to handle data with continuous and non-continuous features was dropped under the guidance of the supervisor in week 6 of term 2. This change was sensible as it was more important to finalise the methods already implemented (for example, overcome numerical issues) at that stage than it was to add new functionality. A structured plan, effective time management and high levels of motivation during the *development* period ensured the 2nd deliverable (project presentation) was completed to a high standard. On reflection, a higher mark would have been achieved in the presentation if a proportion of time in term 2 had been spent applying the software to a dataset, which would have produced results that could have been included.

The report was written over a 2 month period from the start of March to the start of May. During this phase, time was balanced between revision demands from other modules and report writing. Effective use of supervision meetings ensured any problems during this period were overcome and the 3rd deliverable (written report) was completed significantly before the deadline.

9.3 Tools

The following table outlines the tools used during the project and a description of how, when or why they were used.

R & Rstudio	See Sect. 8.1.
Dropbox	The online file hosting service Dropbox was used partly as a means of file transfer between project leader and supervisor, but predominately as a framework for accessing the project's associated files via the cloud.
Email	Emails were used as a means of communication between project leader and supervisor for the purpose of seeking feedback, giving updates and providing guidance on the next steps of the project. Although the majority of discussions regarding the project were held during meetings, emails proved helpful when communication was required between times, especially when weekly meetings changed to fortnightly meetings.
Skype	The advantage of Skype during the project is it allowed real-time communication regardless of geographical location. This was helpful prior to the deadline of the module specification, where feedback was given despite the project supervisor being off campus. Moreover, Skype proved even more helpful when university rules demanded that all meetings were held remotely as a result of the Covid-19 pandemic. Skype's screen share capabilities allowed code specific feedback as well as questions in written English to be answered.
Overleaf & LaTeX	The Overleaf editor was used to produce the written deliverables of the project using LaTeX, as well as providing a framework for producing the equations that had to be displayed in the project presentation. LaTeX was chosen due to its professional appearance, whilst Overleaf was chosen due to the convenience it offers in visualising written reports as they are produced, as well as acting as an online file store. This was helpful during the production of the final report, as it enabled computers with large monitors in the university's library to be used.
natbib & BibTex	BibTex allowed the management of the project's bibliography, whilst natbib allowed the customization of the format of citations.
Google Slides	The presentation program Google Slides offered a free software for producing the project's presentation. Google slides' built-in themes and easy-to-use interface allowed the effortless construction of a professional looking presentation.
Pen & Paper	Although the most simple tool used in the project, pen & paper was invaluable. The ability to write structured notes is dependent on understanding the concepts first. Hence, making handwritten notes forced the ideas behind the project to be understood early on in the project.

Table 4: A table to show the tools used and a description of the effect they had on the project.

9.4 Risks

Risk: The project leader or supervisor falling ill and causing a backlog of work.
Mitigation: Due to the project being spread over such a long period, illness was a likely concern. Skype communications were set up at the start of the project which meant meetings could still be held in the case of illness. However, illness of the project leader posed a serious threat as they were entirely responsible for the completion of all work related to the project. Hence, buffers were built in to the project's plan which were, in fact, needed.

Risk: Loss of work due to human or computer error.
Mitigation: This issue can be separated into a digital risk and physical risk. Firstly, loss of files related to the project (for example, presentation slides, the bibliography, R code) was prevented by using online-based services such as Dropbox, Google Slides and Overleaf. Secondly, photos were taken of the most important handwritten notes as they were produced during term 1. It was not necessary to take photos of all notes, as some were simple enough to be memorized as they were produced.

Risk: Software development resulting in new errors that compromise the functionality of the project's software.
Mitigation: Separate versions of the project's associated code were saved as milestones were reached.

This ensured that new errors could be overcome by tracing back to the previous version of the code. For example, the prototype of the EM implementation during term 1 was copied before it was extended to the full working version.

Risk: Underestimation of the time required to complete phases of the project.
Mitigation: Firstly, the dividing of the project into four stages, with an allocated time period for each, provided a structured and measurable scale to assess the ongoing progress of the project. This process allowed the impact of the project leader's illness during the *research* stage to be easily assessed. Secondly, continuous communication with the project supervisor, who has experience of supervising similar projects with identical time constraints, enabled advice regarding time pressures throughout the project. This advice, alongside a part agile approach to development meant requirements of the project could be updated to ensure the project did not overrun.

Risk: Alternative project direction unintentionally followed.
Mitigation: Due to the fact that concepts were being learned with little prior knowledge during the *research* phase of the project, it was likely too much time would be spent by the project leader unconsciously trying to understand concepts that had little benefit to the project's *development* stage. This risk was mitigated partially by the project supervisor's advice during weekly meetings. If excessive notes were made on a topic that was related to finite mixture models but not the project specifically, the project supervisor would recommend no more time be spent researching that topic.

9.5 Legal, social, ethical & professional issues

The algorithms associated with the project were applied to a real dataset in Sect 7.2. The data was readily available and hence there was not a requirement for appropriate permissions to be obtained before using the dataset. Moreover, it was verified that there was no sensitive information in the data or the results presented in the report. Finally, the regulations outlined by the CRAN Repository Policy will be obeyed when the software is released as an R package.

10 Discussion

10.1 Contribution to the field

A new family of regularized copula-based mixture models has been proposed, and a method of estimating such models for continuous data has been described and implemented in the R software. The implementation, which is to be released in the form of an R package, is the first general ECM implementation available. The implementation permits unconstrained optimization during estimation by performing an appropriate transformation to the parameters of the model. The software allows *Normal*, *Beta* and *Gamma* marginal distributions, for data defined on the whole real line, $(0, \infty)$ and $(0, 1)$, respectively. An optimal correlation structure of the component distributions is adaptively selected for each new modelling problem, and the optimal number of components is identified using BIC. The application of the average silhouette width of observations for the purposes of identifying optimal correlation structure ensures the clusters are well separated and the resulting model has good generalization properties. The application to simulated data and comparison with `mclust` saw the software outperformed existing methods in terms of BIC, clustering performance and inference regarding mixture components.

Copulas are significant in the context of model-based clustering since they allow for the construction of flexible families of models with clusters taking a range of exotic shapes. The ability to first select the marginal properties of the distribution and then capture their implied dependence is highly desirable, as it allows one to handle mixed-domain data in a natural way. Moreover, copula-based mixtures encompass all known mixture models implying no flexibility is forfeited by switching from an elliptical structure, or another proposal in the literature, to a copula-based approach.

10.2 Further research

A number of questions remain for further development, such as, the definition of an appropriate grid of λ values given the sample data. When fitting a family of models for $K = 2 \ldots, 9$, an additional parameter in the tuning grid of λ values results in the addition of 8 new models to the model selection problem. The estimation of these models results in additional computational effort, which introduces especially

large demands in terms of computation time for larger values of K. However, a trade-off problem exists since a tuning grid with few values and large intervals between each value may identify a significantly sub-optimal correlation structure.

Further applications may identify a trend in the relationship between a tuning grid and the resulting mean silhouette widths of the observations in the scenario where other factors, such as n and p, are fixed. Establishing such a pattern would allow the construction of a more appropriate tuning of λ values prior to fitting $|\boldsymbol{K}| \times |\boldsymbol{\lambda}|$ models, where $\boldsymbol{K}$ and $\boldsymbol{\lambda}$ are sets containing the range of mixture components and values of the tuning parameter, respectively.

In a different direction, a rigorous study of the proposed method's performance in a supervised setting can assess if a regularized copula-based mixture estimated via ECM overcomes some of the shortcomings of EM and BIC. It is well known that the combination of GMM estimation and model selection via EM and BIC often results in an overestimation of the number of mixture components when the true value is known. Hennig (2010) proposed a merging approach where normally distributed components' union is interpreted as a single cluster. On the other hand, EM and BIC has demonstrated a proclivity for severely underestimating the number of components in higher dimensions, which was overcome by the introduction of a LASSO-penalized BIC in Bhattacharya and McNicholas (2014). The range of exotic shapes taken by component distributions in copula-based mixtures has been shown to outperform other approaches in terms of capturing the dependence structure underlying sample data (see, Kosmidis and Karlis 2016, *Example 1.1*), however, the resulting inference regarding the number of components in the mixture is yet to be investigated in detail. Sect 7.1 demonstrated one example where the regularized copula-based approach correctly inferred the number of mixture components $K = 2$ using the previously discussed model selection procedure (see Sect 5.4), thereby outperforming a Gausian mixture fitted via `mclust` which incorrectly predicted $K = 4$.

10.3 Future improvements to the regularized copula-based mixture model

In the context of the current definition of the model, the number of parameters (23) associated with the model grows linearly with p in terms of marginals and proportional to the square of p in terms of copula parameters. The latter has a significant negative effect in terms of computation time during estimation on the copula parameter in CM-2 in each iteration of ECM, particularly in the case of high-dimensional data modelling. Kosmidis and Karlis (2016) suggested extending parsimonious parameterizations for normal mixture models like the factor analyzers proposed in Mcnicholas and Murphy (2008) to copula-based mixtures, which has since been investigated in Zhang and Baek (2019). Due to the approach described in this dissertation involving the reparameterization of copula parameters using angles, it is suggested that an alternative direction is adopted. Tsay and Pourahmadi (2017) introduced methods of identifying a structured correlation matrix from a flexible matrix of angles and identifying a set of pivotal angles that can be used to parameterize the structured matrix. This process can reduce the number of parameters associated with the optimization to significantly less than $\frac{1}{2}p(p-1)$, thereby reducing estimation time.

The previous discussion regarding the problem of identifying an approximate tuning grid of λ values to be used as a starting point for many model-based clustering problems is challenging due to many factors that affect the results of mixture model estimation. The influence of n on the magnitude of shrinkage portrayed by the analysis of a simulated dataset (see, Sect. 7.1) can be eliminated by adapting the likelihood of the model. Exchanging the tuning parameter λ regulating the shrinkage term in the likelihood function (24) with $\lambda f(n)$, where $f(n)$ is some function of n, would stabilise the dominance of the first parenthesis in the equation when n is large. Determining an appropriate function $f(n)$ should be considered in future research.

10.4 Future improvements to `rcbmm`

In terms of performance, the regularized copula-based mixture significantly outperformed `mclust` in Sect. 7.1, but the computation time of `mclust` is unrivalled. The fitting time of the optimal model for both applications in Sect. 7 is given in Tab. 5. All timings took place on an iMac (Mid 2011) with a 3.1 GHz Intel Core i5, and 20 GB of RAM memory, running R version 3.6.2. Parallelization across components was used as described in Sect. 6.2. Whilst the timings can be significantly reduced by a less severe termination criterion (see, Sect. 5.3), an attempt to optimize the computation should be made using the `Rccp` package (Eddelbuettel and Balamuta, 2018) in R, which is a clean API allowing the execution of high-performance code by connecting C++ to R. The package helps address bottlenecks in code, such

Application	n	K	λ	q	Iterations	Time	Additional time ($\lambda = 0$)
Simulated dataset	375	2	17	51	23	0.27	0.21
NBA dataset	493	4	12.5	143	28	3.96	12.12

Table 5: The computation time (in minutes) of the optimal models determined in Sect. 7.1 & 7.2 estimated with the new grid of tuning parameters $\lambda = 0, 17$ (for simulated dataset) and $\lambda = 0, 12.5$ (for NBA dataset).

as loops that cannot be easily vectorized because subsequent iterations depend on previous ones. This solution applies directly to the warm starts method for identifying starting values for ECM each time λ is increased.

Aside from favourable efficiency in terms of computation time, a valuable feature of the function call `Mclust()` is the ability to parse a previous result obtained from the function call `mclustBIC()`. The BIC value for models that have been previously computed are not recomputed in such a scenario. Extending `fit.rcbmm()` in the software to adopt a similar methodology would be of great convenience when constructing a family of regularized copula-based mixture models. In the case the user wishes to append a value λ to the tuning grid after observing the results from a previous function call, warm starts could be adopted using the nearest value in the grid rather than recomputing the family of models. This implementation would be especially beneficial as it will relax the trade-off previously described between a large tuning grid and finding a sub-optimal dependence structure, as the user could begin with a tuning grid with relatively large intervals and "zoom-in" on a specific value of the grid after observing the results.

References

Andrews, J. L. and P. D. Mcnicholas (2012). Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions. *Statistics and Computing 22*(5), 1021–1029.

Arakelian, V. and D. Karlis (2014). Clustering dependencies via mixtures of copulas. *Communications in Statistics-Simulation and Computation 43*(7), 1644–1661.

Banfield, J. D. and A. E. Raftery (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, 803–821.

Bhattacharya, S. and P. D. McNicholas (2014). A lasso-penalized bic for mixture model selection. *Advances in Data Analysis and Classification 8*(1), 45–61.

Biernacki, C., G. Celeux, and G. Govaert (2003). Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis 41*(3-4), 561–575.

Celeux, G. and G. Govaert (1995). Gaussian parsimonious clustering models. *Pattern Recognition 28*(5), 781 – 793.

Dean, N. and R. Nugent (2013). Clustering student skill set profiles in a unit hypercube using mixtures of multivariate betas. *Advances in Data Analysis and Classification 7*(3), 339–357.

Delignette-Muller, M. L., C. Dutang, R. Pouillot, J.-B. Denis, and A. Siberchicot (2019). Package 'fitdistrplus'.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological) 39*(1), 1–22.

Di Lascio, F. M. L. and S. Giannerini (2012). A copula-based algorithm for discovering patterns of dependent observations. *Journal of Classification 29*(1), 50–75.

Eddelbuettel, D. and J. J. Balamuta (2018). Extending r with c++: a brief introduction to rcpp. *The American Statistician 72*(1), 28–36.

Fang, H.-B., K.-T. Fang, and S. Kotz (2002). The meta-elliptical distributions with given marginals. *Journal of multivariate analysis 82*(1), 1–16.

Fraley, C. and A. Raftery (1998a). Mclust: Software for model-based cluster and discriminant analysis. *Department of Statistics, University of Washington: Technical Report* (342).

Fraley, C. and A. E. Raftery (1998b). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal 41*(8), 578–588.

Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association 97*(458), 611–631.

Fraley, C. and A. E. Raftery (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of classification 24*(2), 155–181.

Friedman, H. P. and J. Rubin (1967). On some invariant criteria for grouping data. *Journal of the American Statistical Association 62*(320), 1159–1178.

Frühwirth-Schnatter, S. and S. Pyne (2010). Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions. *Biostatistics 11*(2), 317–336.

Hennig, C. (2010). Methods for merging gaussian mixture components. *Advances in data analysis and classification 4*(1), 3–34.

Hofert, M., I. Kojadinovic, M. Maechler, J. Yan, M. M. Maechler, and M. Suggests (2014). Package 'copula'. *URL http://ie. archive. ubuntu. com/disk1/disk1/cran. r-project. org/web/packages/copula/copula. pdf*.

Jajuga, K. and D. Papla (2006). Copula functions in model based clustering. In *From Data and Information Analysis to Knowledge Engineering*, pp. 606–613. Springer.

Jasra, A. (2006). *Bayesian inference for mixture models via Monte Carlo computation*. Ph. D. thesis, Imperial College London (University of London).

Joe, H. (1997). *Multivariate models and multivariate dependence concepts*. CRC Press.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the american statistical association 90*(430), 773–795.

Kosmidis, I. and D. Karlis (2016). Model-based clustering using copulas with applications. *Statistics and computing 26*(5), 1079–1099.

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

Li, D. X. (2000). On default correlation: A copula function approach. *The Journal of Fixed Income 9*(4), 43–54.

Lin, T.-C. and T.-I. Lin (2010). Supervised learning of multivariate skew normal mixture models with missing information. *Computational Statistics 25*(2), 183–201.

Lin, T. I., J. C. Lee, and W. J. Hsieh (2007). Robust mixture modeling using the skew t distribution. *Statistics and computing 17*(2), 81–92.

Lin, T. I., J. C. Lee, and H. F. Ni (2004). Bayesian analysis of mixture modelling using the multivariate t distribution. *Statistics and Computing 14*(2), 119–130.

Lin, T. I., J. C. Lee, and S. Y. Yen (2007). Finite mixture modelling using the skew normal distribution. *Statistica Sinica*, 909–927.

Little, R. J. and D. B. Rubin (1987). Statistical analysis with missing data. *Hoboken, NJ: Wiley*.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Volume 1, pp. 281–297. Oakland, CA, USA.

Maechler, M., P. Rousseeuw, A. Struyf, M. Hubert, K. Hornik, M. Studer, et al. (2013). Package 'cluster'. *Doseglјivo na*.

Maitra, R. (2009). Initializing partition-optimization algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 6*(1), 144–157.

McLachlan, G. J. and K. E. Basford (1988). *Mixture models: Inference and applications to clustering*, Volume 38. M. Dekker New York.

McLachlan, G. J. and T. Krishnan (2007). *The EM algorithm and extensions*, Volume 382. John Wiley & Sons.

McLachlan, G. J. and D. Peel (2004). *Finite mixture models*. John Wiley & Sons.

Mcnicholas, P. D. and T. B. Murphy (2008). Parsimonious gaussian mixture models. *Statistics and Computing 18*(3), 285–296.

Meng, X.-L. and D. B. Rubin (1993). Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika 80*(2), 267–278.

Nelsen, R. B. (2007). *An introduction to copulas*. Springer Science & Business Media.

RCore, T. (2016). R: A language and environment for statistical computing. r foundation for statistical computing, vienna, austria.

RStudio Team (2019). *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, Inc.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics 6*(2), 461–464.

Scrucca, L., M. Fop, T. B. Murphy, and A. E. Raftery (2016). mclust 5: clustering, classification and density estimation using gaussian finite mixture models. *The R journal 8*(1), 289.

Scrucca, L. and A. E. Raftery (2015). Improved initialisation of model-based clustering using gaussian hierarchical partitions. *Advances in data analysis and classification 9*(4), 447–460.

Titterington, D. M., A. F. Smith, and U. E. Makov (1985). *Statistical analysis of finite mixture distributions*. Wiley,.

Tsay, R. S. and M. Pourahmadi (2017). Modelling structured correlation matrices. *Biometrika 104*(1), 237–242.

Vrac, M., L. Billard, E. Diday, and A. Chédin (2012). Copula analysis of mixture models. *Computational Statistics 27*(3), 427–457.

Wickham, H., D. Cook, H. Hofmann, A. Buja, et al. (2011). tourr: An r package for exploring multivariate data with projections. *Journal of Statistical Software 40*(2), 1–18.

Wu, C. J. (1983). On the convergence properties of the em algorithm. *The Annals of statistics*, 95–103.

Zhang, L. and J. Baek (2019). Mixtures of gaussian copula factor analyzers for clustering high dimensional data. *Journal of the Korean Statistical Society 48*(3), 480–492.

A Maximum-likelihood estimation

Suppose observed data $\boldsymbol{x}_{obs} = (\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_n^T)^T$ with n samples assumed to have been independently drawn from the same distribution. If the parameters $\boldsymbol{\Psi}$ govern a density function $f(\boldsymbol{x}_i; \boldsymbol{\Psi})$ $(i = 1 \ldots, n)$, then the resulting density for the data is,

$$f(\boldsymbol{x}_{obs}; \boldsymbol{\Psi}) = \prod_{i=1}^{n} f(\boldsymbol{x}_i; \boldsymbol{\Psi}) = \mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{x}_{obs}) \tag{31}$$

$\mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{X})$ is called the likelihood function, which is a function of the parameters $\boldsymbol{\Psi}$ given the fixed data $\boldsymbol{x}_{obs}$. When performing maximum likelihood estimation, the aim is to maximise $\mathcal{L}$ subject to $\boldsymbol{\Psi}$. In the majority of cases, this is performed by maximising $\log \mathcal{L}$, due to the log-likelihood being analytically simpler to work with.

B EM algorithm's parameter updates

The EM algorithm's M-step 2 requires the maximisation of the conditional expectation of the complete data log-likelihood in (8), which can be separated into K different maximisation problems, such that, for each $j \in \{1, \ldots, K\}$, the expression maximised is

$$\sum_{i=1}^{n} z_{ij} \{\log f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)\}, \tag{32}$$

. where $f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)\}$ takes multivariate normal distribution (3). Hence, the updates for EM are derived as follows. The density

$$f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) = \frac{1}{(\sqrt{2\pi})^p} \frac{1}{\sqrt{\det(\boldsymbol{\Sigma}_j)}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j) \Big\}$$

gives

$$\log f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j) = -\frac{p}{2} \log 2\pi - \frac{1}{2} \log \big|\boldsymbol{\Sigma}_j\big| - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j).$$

Differentiating with respect to $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ gives:

$$\frac{d}{d\boldsymbol{\mu_j}} \log(f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)) = \boldsymbol{x}_i - \boldsymbol{\mu}_j, \qquad \frac{d}{d\boldsymbol{\Sigma_j}} (\log(f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)) = \frac{-1}{2\boldsymbol{\Sigma}_j} + \frac{(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)}{2\boldsymbol{\Sigma}_j^2}.$$

Hence,

$$\sum_{i=1}^{n} z_{ij} \frac{d}{d\boldsymbol{\mu_j}} \log(f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)) = 0 \Rightarrow \sum_{i=1}^{n} z_{ij} \boldsymbol{x}_i = \sum_{i=1}^{n} z_{ij} \boldsymbol{\mu}_j \Rightarrow \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} z_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{n} z_{ij}}$$

$$\sum_{i=1}^{n} z_{ij} \frac{d}{d\boldsymbol{\Sigma j}} \log(f_j(\boldsymbol{x}_i; \boldsymbol{\theta}_j)) = 0 \Rightarrow \sum_{i=1}^{n} \frac{z_{ij}}{2\boldsymbol{\Sigma}_j} = \sum_{i=1}^{n} z_{ij} \frac{(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)}{2\boldsymbol{\Sigma}_j^2} \Rightarrow \boldsymbol{\Sigma}_j^{=} \frac{\sum_{i=1}^{n} z_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T (\boldsymbol{x}_i - \boldsymbol{\mu}_j)}{\sum_{i=1}^{n} z_{ij}}.$$

C Silhouette width

The silhouette width of an observation $\boldsymbol{x}_i$ in a dataset $\boldsymbol{x}_{obs}$ can be computed after a clustering of the data into K components has been identified, with each $\boldsymbol{x}_i$ assigned to exactly one of $1, \ldots, K$. Define $C_j = \{\boldsymbol{x}_i : \boldsymbol{x}_i \text{is assigned to cluster } j\}$ $(j = 1, \ldots, K)$, then define for observation $\boldsymbol{x}_i$ in C_j:

- The measure of similarity of the point $\boldsymbol{x}_i$ with other points in the same cluster as

$$a(\boldsymbol{x}_i) = \frac{1}{\big|C_j\big| - 1} \sum_{\boldsymbol{x}_k \in C_j} d(\boldsymbol{x}_i, \boldsymbol{x}_k).$$

- The measure of dissimalirity of the point $\boldsymbol{x}_i$ to it's neighbouring cluster as

$$b(\boldsymbol{x}_i) = \min_{j' \neq j} \frac{1}{\left|C_{j'}\right|} \sum_{\boldsymbol{x}_k \in C_{j'}} d(\boldsymbol{x}_i, \boldsymbol{x}_k).$$

- The silhouette width of the point $\boldsymbol{x}_i$ as

$$s(\boldsymbol{x}_i) = \frac{a(\boldsymbol{x}_i) - b(\boldsymbol{x}_i)}{\max\{a(\boldsymbol{x}_i) - b(\boldsymbol{x}_i)\}}.$$

Additionally, the distance metric $d(\boldsymbol{x}_i, \boldsymbol{x}_k)$ is taken as the Euclidean distance between the point $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$. It follows that $-1 \leq s(\boldsymbol{x}_i) \leq 1$ for all $i = 1, \ldots, n$. A value closer to 1 suggests the point is appropriately clustered, a value close to -1 implies the point has been assigned to a poor choice of cluster, and a value near 0 suggests the point is on the border of two different clusters.

D Software documentation

The documentation is attached below.

Package 'rcbmm'

May 18, 2020

Type Package

Title Regularized copula-based mixture models

Version 0.1.0

Imports cluster, mclust, copula, fitdistrplus, parallel

Author Ben Barlow [aut, cre],
Ioannis Kosmodis [aut]

Maintainer Ben Barlow <ben.j.barlow.1@gmail.com>

Description Model-based clustering through regularized copula-based mixture models with arbitrary marginal distributions. Estimation via the expectation-conditional-maximization algorithm. Shrinkage driven methods applied to the copula parameters controlled by a tuning parameter to adaptively select the best correlation structure for the model. Model selection based on average silhouette width and BIC.

License MIT

Encoding UTF-8

RdMacros Repack

R topics documented:

angles2rho *A tool for converting a matrix of angles to a correlation matrix*

Usage

```
angles2rho(theta)
```

Arguments

theta	A $(p-1)$ times $(p-1)$ matrix of angles.

Value

A p times p correlation matrix corresponding to `theta`.

See Also

rho2angles

Examples

```
theta <- matrix(rep(0.5, 4), 2, 2)
angles2rho(theta)
```

CM.step.1 *CM-step 1 for regularized copula-based mixture models estimation*

Description

Implements the conditional-maximization step 1 of the ECM algorithm for regularized copula-based mixture model.

Usage

```
CM.step.1(x, K, z, mvdc, margins, trace = T)
```

Arguments

x	A numeric matrix or data frame of observations. Rows correspond to observations and columns correspond to variables.
K	The number of mixture components.
z	A numeric matrix representing the current value of the posterior probabilities of membership of the observations after the expectation step of the ECM algorithm. Columns are associated with a mixture component and rows are associated with observations. The current value of the probabilities should be computed via E.step.
mvdc	A list of objects of class `mvdc`. Each element of the list corresponds to a mixture component and contains the previous estimates for the component distribution's marginal and copula parameters.

margins	A character vector specifying the marginal distributions of the components in the mixture. The vector must have a length equal to the number of columns in x. Each element must be equal to "norm", "beta" or "gamma".
trace	A logical value indicating if an update regarding the step's progress should be displayed.

Value

A list of objects of class mvdc. Each element of the list corresponds to a mixture component and contains the updated estimates for the component distribution's marginal paramters and the same estimates as were parsed for the estimates of the copula parameter.

See Also

CM.step.2, ECM.Algorithm

CM.step.2 *CM-step 2 for regularized copula-based mixture models estimation*

Description

Implements the conditional-maximization step 2 of the ECM algorithm for regularized copula-based mixture model.

Usage

```
CM.step.2(x, K, z, mvdc, margins, lambda trace = T)
```

Arguments

x	A numeric matrix or data frame of observations. Rows correspond to observations and columns correspond to variables.
K	The number of mixture components.
z	A numeric matrix representing the current value of the posterior probabilities of membership of the observations after the expectation step of the last iteration of the ECM algorithm. Columns are associated with a mixture component and rows are associated with observations.
mvdc	A list of objects of class mvdc. Each element of the list corresponds to a mixture component and contains the current estimates for the component distribution's marginal and previous estimates for copula parameters. The estimates for the marginal parameters should have been updated by CM.step.1.
margins	A character vector specifying the marginal distributions of the components in the mixture. The vector must have a length equal to the number of columns in x. Each element must be equal to "norm", "beta" or "gamma".
lambda	A numeric value indicating the value of the tuning parameter for regularization.
trace	A logical value indicating if an update regarding the step's progress should be displayed.

Value

mvdc	A list of objects of class mvdc. Each element of the list corresponds to a mixture component and contains the updated estimates for the component distribution's copula paramter and the same estimates as were parsed for the estimates of the marginal parameters.
penalty	The shrinkage penalty to apply to the log-likelihood of the model, resulting from the estimates of the copula parameters and the value of lambda.

See Also

CM.step.1, ECM.Algorithm

E.step *E-step for regularized copula-based mixture models*

Description

Implements the expectation step of the ECM algorithm for regularized copula-based mixture model.

Usage

```
E.step(x, K, mixing_probs, mvdc, margins)
```

Arguments

x	A numeric matrix or data frame of observations. Rows correspond to observations and columns correspond to variables.
K	The number of mixture components.
mixing_probs	A numeric vector indicating the mixing proportions of the mixture model.
mvdc	A list of objects of class mvdc. Each element of the list corresponds to a mixture component and contains the current estimates for the component distribution's marginal and copula parameters.
margins	A character vector specifying the marginal distributions of the components in the mixture. The vector must have a length equal to the number of columns in x. Each element must be equal to "norm", "beta" or "gamma".

Value

A numeric matrix representing the posterior probabilities of membership of the observations after performing a single expectation step of the ECM algorithm. Columns are associated with a mixture component and rows are associated with observations.

See Also

ECM.Algorithm

`ECM.Algorithm` *Estimation of a regularized copula-based mixture model*

Description

Implements the ECM algorithm for regularized copula-based mixture models, starting with the expectation step.

Usage

```
ECM.Algorithm(x, K, lambda, start, margins, trace = T, maxit = 1000, epsilon = 1e-06)
```

Arguments

`x`	A numeric matrix or data frame of observations. Rows correspond to observations and columns correspond to variables.
`K`	An integer specifying the number of components for which a regularized copula-based mixture model should be fitted.
`lambda`	A numeric value indicating the value of the tuning parameter for regularization.
`start`	A list providing the starting values for ECM. The list fit.rcbmm
`margins`	A character vector specifying the marginal distributions of the components in the mixture. The vector must have a length equal to the number of columns in `x`. Each element must be equal to `"norm"`,`"beta"` or `"gamma"`.
`trace`	A logical value indicating if an update regarding the algorithm's progress should be displayed after each iteration.
`maxit`	A numeric value indicating the maximal number of ECM iterations.
`epsilon`	A numeric value specifying the tolerance associated with determining when convergence of the ECM algorithm has been achieved.

Value

`K`	The number of mixture components.
`lambda`	The value of the tuning parameter
`z`	A numeric matrix representing the posterior probabilities of membership of the observations after the expectation step of the last iteration of the ECM algorithm. Columns are associated with a mixture component and rows are associated with observations.
`clusters`	A classification vector indicating the associated cluster of each observation. The classification corresponds to `z`.
`loglik`	A numeric vector displaying the penalized log-likelihood after each iteration of ECM.
`param_number`	The number of independent parameters associated with the model.
`BIC`	The BIC value of the model. Computed using the unpenalized log-likelihood after the last iteration of the ECM algorithm.
`mixing_probs`	The mixing proportions associated with the model.
`mvdc`	A list of objects of class `mvdc`. Each element of the list corresponds to a mixture component.

transformation	A character value indicating the transformation when identifying starting values. The value is NULL unless lambda=0. See initialize.ECM.
marginal_param	A list containing the marginal parameters of the model as estimated by ECM. Each element corresponds to a mixture component.
copula_param	A list containing the copula parameters of the model as estimated by ECM. Each element corresponds to a mixture component.
copula_param_angles	A list containing the copula parameters of the model re-expressed as angles.
silhouette	See information regarding silhouette package. Add reference here.

See Also

E.step, CM.step.1,CM.step.2, ECM.Algorithm, ECM.Algorithm

Examples

```
iris_x <- as.matrix(datasets::iris[, -5])
margins_iris = rep("norm", 4)
K <- 3
init_iris <- initialize.ECM(iris_x, K, margins_iris, transform = F)
ECM_out <- ECM.Algorithm(iris_x, K = 3, lambda = 2, start = init_irs, margins = margins_iris)
```

---

extract.copula.pars *Extraction of copula parameters of a regularized copula-based mixture model*

Description

A function for extracting the copula parameter parameterizing the mixture components of a regularized copula-based mixture model from a list of mvdc objects. The copula parameter for each mixture component can be extracted in terms of a correlation matrix or a matrix of angles.

Usage

```
extract.copula.pars(mvdc, as_angles = F)
```

Arguments

mvdc	A list of objects of class mvdc. Each element of the list corresponds to a mixture component and contains the estimates for the component distribution's marginal and copula parameters.
as_angles	A logical value indicating whether the copula parameter should be returned as a matrix of angles instead of a correlation matrix.

Value

A list of matrices corresponding to the copula parameter of each mixture component contained in mvdc.

See Also

extract.marginal.pars

Examples

```
iris_x <- as.matrix(datasets::iris[, -5])
margins_iris = rep("norm", 4)
K <- 3
init_iris <- initialize.ECM(iris_x, K, margins_iris, transform = F)
ECM_out <- ECM.Algorithm(iris_x, K = 3, lambda = 2, start = init_irs, margins = margins_iris)
cop_pars <- extract.copula.pars(ECM_out$mvdc)
```

`extract.marginal.pars` *Extraction of marginal parameters of a regularized copula-based mixture model*

Description

A function for extracting the marginal parameters parameterizing the mixture components of a regularized copula-based mixture model from a list of `mvdc` objects.

Usage

```
extract.marginal.pars(mvdc)
```

Arguments

`mvdc`	A list of objects of class `mvdc`. Each element of the list corresponds to a mixture component and contains the estimates for the component distribution's marginal and copula parameters.

Value

A list of lists corresponding to the marginal parameters of each mixture component contained in `mvdc`. Each sub-list in the list has a length corresponding to the number of marginal distributions defined by the model.

See Also

extract.copula.pars

Examples

```
iris_x <- as.matrix(datasets::iris[, -5])
margins_iris = rep("norm", 4)
K <- 3
init_iris <- initialize.ECM(iris_x, K, margins_iris, transform = F)
ECM_out <- ECM.Algorithm(iris_x, K = 3, lambda = 2, start = init_irs, margins = margins_iris)
cop_pars <- extract.copula.pars(ECM_out$mvdc)
```

`fit.rcbmm` *Model selection*

Description

The function selects the most appropriate model from a family of regularized copula-based mixture models arising from a varying number of components and a differing shirnkage parameter.

Usage

```
fit.rcbmm(x, K, lambda_grid, margins, transform, ...)
```

Arguments

`x`	A numeric matrix or data frame of observations. Rows correspond to observations and columns correspond to variables.
`K`	An integer vector specifying the number of components for which a regularized copula-based mixture model should be fitted. The default is `K=2:9`.
`lambda_grid`	An integer vector specifying the the values of the shrinkage parameter for which a regularized copula-based mixture model should be fitted. The default is `lambda_grid = 0`. If the vector parsed does not contain `0`, then `0` is appended as starting values can only be obtained in the case no regularization is applied to the model.
`margins`	A character vector specifying the marginal distributions of the components in the mixture. The vector must have a length equal to the number of columns in `x`. Each element must be equal to `"norm"`,`"beta"` or `"gamma"`.
`transform`	A logical value indicating whether or not starting values for the case `lambda = 0` should be obtained using the transformations SPH, PCS, PCR and SVD. The default is TRUE.

Value

`BIC`	A matrix demonstrating the BIC values achieved by each model in the family. Each column corresponds to a given value of lambda and each row corresponds to a given number of components.
`SIL`	A matrix demonstrating the average silhouette width achieved by each model in the family. Each column corresponds to a given value of lambda and each row corresponds to a given number of components.
`all_models`	A list of lists with each element containing information about a specific model fitted (see the help file for ECM.Algorithm for details).
`selected_models`	A list containing the information about the optimal model for each number of mixture components in the family. The optimal model for each number of componets is selected by picking the lambda that results in the largest average silhouette width. See the help file for silhouette for details.
`final_model`	The model contained in `selected_models` that maximized BIC. (see the help file for ECM.Algorithm for details)

See Also

ECM.Algorithm, ECM.initialize.ECM

Examples

```
n <- 100 ; mix_1 <- cbind(rnorm(n, 2, 1), rgamma(n, 10, 1/5), rbeta(n, 2, 20))
n <- 120 ; mix_2 <- cbind(rnorm(n, -2, 1), rgamma(n, 5, 2), rbeta(n, 6, 5))
sim_x <- rbind(mix_1, mix_2)
margins_sim <- c("norm", "gamma", "beta")
K_grid <- c(2, 3)
lg <- seq(0, 30, 1)
fit_out <- fit.rcbmm(sim_x, K = K_grid, lg, margins_sim)
```

`initialize.ECM` *Starting values for ECM*

Description

Identifies optimal starting values for ECM using the results of MBHAC applied to the data. The data undergoes transformations to enhance separation amongst groups prior to performing MBHAC.

Usage

```
initialize.ECM(x, K, margins, transform = T, hc_pairs = NULL, classification = NULL)
```

Arguments

`x`	A numeric matrix or data frame of observations. Rows correspond to observations and columns correspond to variables.
`K`	The number of mixture componets.
`margins`	A character vector specifying the marginal distributions of the components in the mixture. The vector must have a length equal to the number of columns in x. Each element must be equal to "norm", "beta" or "gamma".
`transform`	A logical value indicating whether or not starting values should be obtained using the transformations SPH, PCS, PCR and SVD. The default is `TRUE`.
`hc_paris`	The results from MBHAC obtained from the function call `hc()` using the mclust package. If `NULL`, the function obtains the results by calling `hc()`.
`classification`	A numeric vector representing a partitioning of the data `x`. If not `NULL`, the clustering identified by the vector takeß preference over the clustering identified by `hc_pairs`.

Value

`mixing_probs`	A numeric vector indicating the starting values of the mixing proportions.
`mvdc`	A list of objects of class `mvdc`. Each element of the list corresponds to a mixture component and contains the starting values for the component distribution's marginal and copula parameters.
`transformation`	A character string indicating the transformation applied prior to performing MBHAC.
`loglik`	The log-likelihood corresponding of the model given the starting values of the parameters and the data.

See Also

hc

Examples

```
iris_x <- as.matrix(datasets::iris[, -5])
margins_iris = rep("norm", 4)
K <- 3
init_iris <- initialize.ECM(iris_x, K, margins_iris, transform = F)
init_iris <- initialize.ECM(iris_x, K, margins_iris, transform = T)
```

P2p.angles *A tool to work with matrices of angles*

Description

Creates a numeric vector of parameters from a matrix of angles.

Usage

```
P2p.angles(theta)
```

Arguments

theta	A matrix of angles to be converted into a parameter vector.

Value

A numeric parameter vector representing angles.

See Also

p2P.angles

Examples

```
theta <- matrix(c(0.9, 0, 1.2, 0.3), 2, 2)
P2p.angles(theta)
```

p2P.angles *A tool to work with matrices of angles*

Description

Creates a matrix from a given vector of parameters representing angles.

Usage

```
p2P.angles(params)
```

Arguments

params	A parameter vector to be converted into a matrix of angles.

Value

A matrix of angles.

See Also

P2p.angles

Examples

```
param <- c(0.3, 0.2, 0.6)
p2P.angles(param)
angles2rho(p2P.angles(param))
```

rho2angles *A tool for converting a correlation matrix to a matrix of angles*

Usage

```
rho2angles(rho_mat)
```

Arguments

rho_mat	A p times p correlation matrix to be converted to a matrix of angles.

Value

A $(p-1)$ times $(p-1)$ matrix of angles corresponding to `rho_mat`.

See Also

ECM.angles2rho

Examples

```
rho_mat <- matrix(c(1, 0.5, 0.5, 1), 2, 2)
rho2angles(rho_mat)
```

Index