scientific reports

OPEN

Novel pruning and truncating of the mixture of vine copula clustering models

Fadhah Amer Alanazi

The mixture of the vine copula densities allows selecting the vine structure, the most appropriate type of parametric marginal distributions, and the pair-copulas individually for each cluster. Therefore, complex hidden dependence structures can be fully uncovered and captured by the mixture of vine copula models without restriction to the parametric shape of margins or dependency patterns. However, this flexibility comes with the cost of dramatic increases in the number of model parameters as the dimension increases. Pruning and truncating each cluster of the mixture model will dramatically reduce the number of model parameters. This paper, therefore, introduced the first pruning and truncating techniques for the model-based clustering algorithm using the vine copula model, providing a significant contribution to the state-of-the-art. We apply the proposed methods to a number of well-known data sets with different dimensions. The results show that the performance of the individual pruning and truncation for each model cluster is superior to an existing vine copula clustering model.

Model-based clustering for unsupervised learning using finite mixture models has received growing interest for decades. Finite mixture models assume that the data are generated from a mixture of g components. Each observation has a probability of belonging to one of these components. In the literature, finite mixture models are commonly used in many areas (see, for example[1–3]). Recently, the mixture of vine copula models received increasing interest in the literature for several reasons. First, the vine copula is a multivariate extension of the copula model using conditional densities. Therefore, copula models allow one to model the marginal distributions independently from the dependence patterns. Hence, one can fit different parametric shapes of the marginal distributions for each variable. Second, the vine copula models work on two variables at a time; hence, no restriction on the type of the bivariate copulas for each pair of variables. Thus, different types of bivariate copulas can be fitted to capture a wide range of complex dependence structures, including symmetric and asymmetric dependence shapes. Therefore, each mixture component has its flexible density. In the literature, the first attempt to incorporate the vine copula models into the finite mixture model is the work of[4]. Kim et al.[4] introduce the mixture of (Drawable vine copula) D-vine copula densities, where the vine structure is fixed for all mixture components, and one type of the bivariate copula was fitted to all pairs of variables. Roy and Parui[5] established a mixture of the vine copula models using a small number of the bivariate copula types and restricted their work to a sub-class of the vine copula model. Alanazi[6] extended the work of[4] into two-folds. First, the author extends the model from a mixture of D-vine to a mixture of regular vine (R-vine) copula model. The R-vine copula model is a general class of vine copula models that allow for a free vine structure. Second, the author fits a wide range of bivariate copula types. However, the author keeps the vine structure fixed among all the mixture components. Recently[7], introduced a model-based clustering algorithm with a vine copula model that allows the vine structure to vary from one mixture component to another. Their method contains five main steps. In the first step, the fast clustering such as `k-means` of[8] is used for the initial data clustering. In the second step, the truncated (at the first tree (level)) vine copula model is fitted and estimated for each cluster data. The n-dimensional vine copula model is called truncated at level $\mathcal{T}$ if all conditional bivariate copulas after level $\mathcal{T}$ are set to the independent copulas. Truncated the vine copula at the first level yields a Markov tree model. The aim of truncating the vine copula model is to reduce the computation complexity in high-dimensional cases. In the third step, the model parameters are estimated using the Expectational Conditional Maximization algorithm (ECM algorithm) of[9], keeping the marginal distribution, bivariate copulas, and vine structures fixed based on the selection in the second step. Hence, the iteration steps of the ECM work on the Markov tree instead of the full vine copula (no-truncation level) model to reduce the model computational complexity. In the fourth step, the data is regenerated

Department of Mathematics and Sciences, Prince Sultan University, 11586 Riyadh, Saudi Arabia. email: fanazi@psu.edu.sa

Variable	First cluster (ϕ)	Second cluster (ϕ)
Var1	Normal (1, 0.4)	Normal (1.5, 0.2)
Var2	Normal (10, 4)	Gamma (1.5, 0.5)
Var3	Normal (1.2, 0.2)	Normal (1, 0.3)
Var4	Gamma (0.9, 0.9)	Gamma (1.5, 0.25)
Var5	Normal (1.2, 0.45)	Normal (1.3, 0.3)
Var6	Normal (0.8, 0.8)	Log-normal (1.2, 0.25)

Table 1. Summary of the fitted univariate marginal distribution for each cluster. The numbers in the bracket refer to the marginals' parameters.

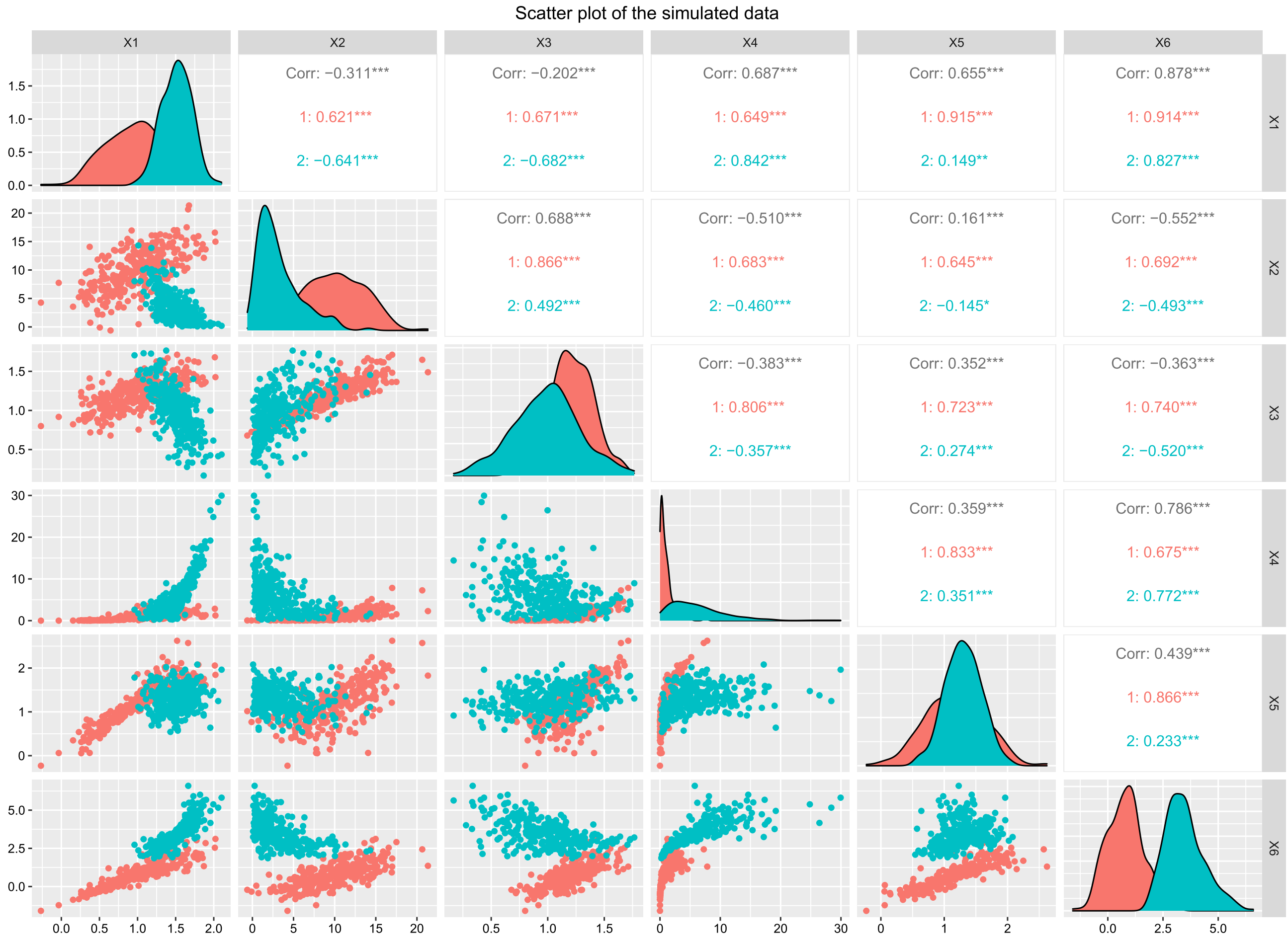

Figure 1. Scatter plot of the simulated data.

based on the successive steps of the ECM algorithm. In the final step, a full vine copula model is fitted to the final clustered data, where the marginal distribution, bivariate copulas, and vine structure of each cluster are updated. Regardless of the flexibility of their method, a mixture of Markov trees does not provide a starting value for the model's parameters at the remaining vine trees. Therefore, important dependence may be ignored in the estimation process. Therefore, we think the full vine copula model should be fitted to the clustered data in all steps with an individual estimation of the truncation level for each cluster. Hence, the truncation level is estimated based on the cluster data instead of the fixed prior truncation level. Alanazi[10] incorporate the truncation method of[11], using selection criteria such as Akaike Information Criteria (AIC) of[12]) and Bayesian Information Criteria (BIC) of[13], into the R-vine copula mixture models, where the bivariate copulas are the mixture components. However, in the mixture of R-vine densities, the R-vine densities are the mixture components (this paper). Therefore, for the mixture of R-vine densities, the truncation level should be determined individually for each cluster. In addition, AIC is known to select a complex model (see[14–16]). BIC has two drawbacks. It can select the true model if the number of the possible parameters increases sufficiently slowly with the sample size, and it assumes that all the models are equally likely[17]. Therefore, identifying the optimal truncation level for each cluster is needed. It can provide numerous flexibility to the mixture vine copula models. In addition, for the nun-truncated levels, pruning each cluster will add extra reduction to the mixture of the vine copula densities, especially in high-dimensional applications. The pruning method aims to fit independent copulas to all pairs of variables with weak/independent dependence structures. To the best of the author's knowledge, individual pruning and truncating vine copula model of each mixture component do not exist in the literature. Therefore, this present research provides a novel method and a great contribution to state-of-the-art. For the pruning vine copula model, we apply the independent test using Kendall's tau of[18]. For the truncation, we adopt the truncation technique of the[17] into the mixture content. We conducted a comprehensive real-data study to illustrate the performance of the proposed method. The results show a dramatic reduction in the number of model parameters. Furthermore, the proposed method outperforms the existing vine copula clustering model.

The remainder of the paper is divided as follows. Section introduces copula, vine copula, and model-based clustering algorithm using the vine copula model, the pruning and truncation approaches. Section provides the result of the simulation and real-data applications. Section discusses the founding results of the studies in this paper.

Results

In this section, we illustrate the performance of the proposed method for simulation and real data applications.

Simulation study. We simulate two mixture components from a 6-dimensional R-vine vine copula model (truncated at tree 2) with 300, and 500 observations, respectively for each cluster. The simulated data is repeated 100 times. We simulate the data using `vineclust` Git-hub repository of[19]. Table 1 shows the summary of the univariate marginal distributions with their corresponding parameters for each cluster. Figure 1 presents the scatter plot of the simulated data (300 observations). Listing (1) and Listing (2) present the summary of the two-component mixture of the vine copula model, where par and tau refer to copula parameter(s) and the corresponding Kendall's tau value (the detail of the fitted models is given by `RvineMatrix` function of the R-program's[20] package `VineCopula`[21]).

```
Tree 1:
4,5  Gumbel (par = 4, tau = 0.75)
1,2  Gaussian (par = 0.6, tau = 0.41)
1,3  Gaussian (par = 0.7, tau = 0.49)
4,1  Clayton (par = 4, tau = 0.67)
6,4  Survival Joe (par = 6, tau = 0.72)

Tree 2:
1,5;4  Gaussian (par = 0.9, tau = 0.71)
3,2;1  Clayton (par = 3, tau = 0.6)
4,3;1  Survival Gumbel (par = 4, tau = 0.75)
6,1;4   Frank (par = 6, tau = 0.51)
Tree 3:
3,5;1,4  Independence
4,2;3,1  Independence
6,3;4,1 Independence
Tree 4:
2,5;3,1,4  Independence
6,2;4,3,1  Independence
Tree 5:
6,5;2,3,1,4  Independence
```

Listing 1. First cluster

```
Tree 1:
4,5   Clayton (par = 0.9, tau = 0.31)
1,2   Gaussian (par = -0.6, tau = -0.41)
1,3   Gaussian (par = -0.7, tau = -0.49)
4,1   Survival Clayton (par = 4, tau = 0.67)
6,4   Clayton (par = 6, tau = 0.75)
Tree 2:
1,5;4   Gaussian (par = -0.5, tau = -0.33)
3,2;1   Clayton (par = 0.3, tau = 0.13)
4,3;1   Joe (par = 4, tau = 0.61)
6,1;4   Frank (par = 6, tau = 0.51)
Tree 3:
3,5;1,4   Independence
4,2;3,1   Independence
6,3;4,1   Independence
Tree 4:
2,5;3,1,4    Independence
6,2;4,3,1    Independence
Tree 5:
6,5;2,3,1,4   Independence
```

Listing 2. Second Cluster

Listing (1) and Listing (2) shows the two-components 6—dimensional vine copula mixture model. The listings show that the vine copula model for each cluster is truncated at the second tree. All the trees after the second tree are specified with independent bivariate copulas. We generated two simulated data sets from this model with 300 and 500 observations, respectively. For the sake of comparison, we fit the Gaussian finite mixture model (from[22] package using the default setting of the package), Tvcmm, Fvcmm, and k-means. Tables 2 and 3 summarize the performance of each fitted model for the simulated data set with 300 observations and 500 observations , respectively. The best-fit model is shown in bold text. Figure 2 shows the box plots of the fitted models for each simulated data set.

Real-data application. To test the performance of the proposed method, we applied it to several real data sets, namely diabetes, Banknotes, Flea and Sonar data sets. Table 3 summarizes the results of the truncated mixture of vine copula models and the full models. The better performance is shown in bold text.

Discussion

This paper incorporates the pruning and truncation methods with the vine copula model-based clustering algorithm. The pruning pairs and truncation levels are determined individually for each cluster. To illustrate the performance of the proposed method, we apply it to a simulation and real data sets. We evaluate the performance of the newly proposed method (Tvcmm), the Fcvmm algorithm of[7], the Gaussian mixture model (GMM), and k-means. Figure 2 shows the Box plots of the misclassification rate of each algorithm per simulation replication data set. Tables 2 and 3 summarize the performance of each algorithm per simulated replication. The performance evaluation of the fitted model for the real data setes is summarized in Table 4 Lower BIC value or misclassification rate are used as a selection criterion for better clustering assignment.

For the simulated data sets, Fig. 2 shows that the Tvcmm, Fvcmm, and GMM provide a better fit than the k-means algorithm. Also, the figure shows that the performance of the Tvcmm and Fvcmm algorithms is noticeably close to each other, and both models provide a better fit than the GMM algorithm. Regarding the misclassification rate, the Tvcmm model provides better clustering assignment, while k-means is the worst. One can notice that although the misclassification rate of the Tvcmm is lower than the Fvcmm model, overall, the accuracy rate of both models is close to each other. The main reason of almost similar performance of Tvcmm and Fvcmm models, is that the data is only truncated at the second tree level. Hence, the performance of the Fvcmm algorithm, with the vine copula model truncated at the first tree level at the initial step, is close to the Tvcmm (truncated at the second tree level). Therefore, this illustrates that the truncation tree level of the data influences the final result, which is illustrated in the real data studies. From Table 4, for the small dimensional data sets, namely, Diabetes, Banknote, and Flea, the performance of Tvcmm, Fcvmm, and GMM are either identical or almost the same. However, GMM outperforms all the other algorithms for BIC value and accuracy rate for the Diabetes and Flea data sets. For Diabetes, the GMM model results in 86.21% misclassification accuracy and with BIC of – 4751.316. In the case of the Flea data set, the GMM model's performance results in a 100% accuracy

Criteria	GMM	Tvcmm	Fvcmm	k-means
Miscassification rate	0.00765	**0.00168**	0.00217	0.08821667
Average BIC	**– 5372.103**	3686.947	3589.848	–

Table 2. The summary of the performance of the Tvcmm and Fvcmm, GMM, and k-means methods fitted to the simulated data with 300 observations. Tvcmm, Fvcmm, GMM, and k-means refer to the truncated vine copula mixture model, full vine copula mixture model, Gaussian mixture model and the k-means method, respectively.

Criteria	GMM	Tvcmm	Fvcmm	k-means
Miscassification Rate	0.00529	**0.00185**	0.00202	0.08621
Average BIC	**– 8818.664**	6013.238	5877.62	–

Table 3. The summary of the performance of the Tvcmm and Fvcmm, GMM, and k-means methods fitted to the simulated data with 500 observations. Tvcmm, Fvcmm, GMM, and k-means refer to the truncated vine copula mixture model, full vine copula mixture model, Gaussian mixture model and the k-means method, respectively.

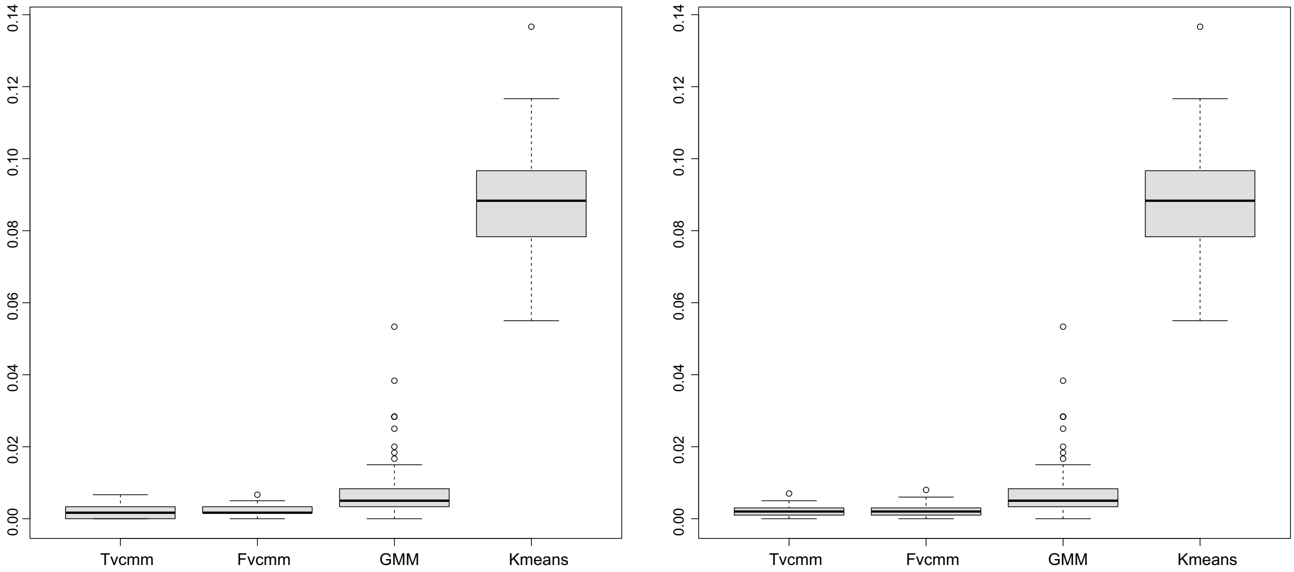

Figure 2. The box plot of the clustering performance of the fitted models for the simulated data (300) observations (left panel) and for the simulated data (500) observations (Right panel).

rate with a BIC of – 2785.572. As a result, the accuracy classification of the Tvcmm and Fvcmm algorithms are identical for Diabetes, Banknote, and Flea data sets. The result is hardly surprising, as the truncation tree level for all latter data sets is at the first tree. Therefore, the performance of Tvcmm is identical to the one of the Fvcmm model, as both treat the data at the initial steps as Markov tree structure. However, for the Sonar data set with individual truncation level for each cluster, the Tvcmm model outperforms all the fitted models with 80.3% accuracy classification and BIC of – 40711.15, while the accuracy rate of the Fvcmm, GMM, and k-mean are 54.3%, 64.9%, and 79.327%, respectively. In addition, the Tvcmm provides a substantial model parameter reduction, resulting in 522 model parameters instead of 4052 parameters for the Fvcmm model. The result of the real data applications strongly supports this paper's contribution and goal. From the result, the conclusion can be summarized into two main points based on the strength of the dependency among variables as follows:

- If the data exhibit weak/independent conditional dependency structure among variables after the first tree, then truncating the vine copula model at the first tree level will not affect the final result. Therefore, the misclassification rate is identical to the one of the Fvcmm model. However, due to the pruning method, Tvcmm result in less number of the estimated model parameters (this is noticed in the result of all the real data sets). In most cases, the BIC criterion selects the model with lower parameters. Comparing the result of the Fvcmm and GMM algorithm for the Sonar data set, BIC criterion selects the Fvcmm model, while its

Data (d)	Model	Misclassification rate	Average BIC	$\mathcal{T}_1(\mathcal{T}_2)$ $(\mathcal{T}_3)$ (δ)
Diabetes (3)	Tvcmm	0.179	4756.75	**1 (1)(1)(32)**
	Fvcmm	0.179	4773.05	NA(34)
	GMM	**0.137931**	**– 4751.316**	–
	k-means	0.179	–	–
Banknote (6)	Tvcmm	**0.005**	1674.75	**1 (1)(-)(40)**
	Fvcmm	**0.005**	1696.63	NA(63)
	GMM	**0.005**	**– 1717.445**	–
	k-means	0.04	–	–
Flea (6)	Tvcmm	0.0405	2791.96	**1(1)(1)(51)**
	Fvcmm	0.0405	2893.71	NA(90)
	GMM	**0**	**– 2785.572**	–
	k-means	0.02702	–	–
Sonar (60)	Tvcmm	**0.197**	**– 40711.15**	**1(3)(-)(522)**
	Fvcmm	0.457	– 30501.02	NA(4052)
	GMM	0.35096	33500.22	–
	k-means	0.20673	–	–

Table 4. The summary of the performance of the Tvcmm and Fvcmm, GMM, and k-means methods fitted to the simulated data with 500 observations. Tvcmm, Fvcmm, GMM, and k-means refer to the truncated vine copula mixture model, full vine copula mixture model, Gaussian mixture model and the k-means method, respectively. d, $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$, and δ refer to the data dimension (without the class variable), the truncation level of the first cluster, the truncation level of the second cluster, the truncation level of the third cluster, and the total number of the estimated model parameters, respectively. Significant values are in bold.

accuracy classification is lower than that of the GMM model. Therefore, our findings support the one of[7], that a better selection criterion than BIC value is needed for the vine copula model, which can be considered as possible future work.

- If the data exhibit a strong conditional dependency structure, with a truncation tree level that can vary from one cluster to another, then the performance of the Tvcmm is superior to other fitted models. Moreover, Tvcmm results in a dramatic reduction in the number of the estimated parameters of the model.

Methods

Copula models have been an interesting research area for decades in several areas (see, for example[23–26]), due to Sklar's theorem[27].

Theorem 1 (Sklar's theorem) *For any an n-dimensional distribution function*, H, *with marginal distributions* $H_1 = H_1(x_1), \dots, H_n(x_n)$, *then there exists an n-dimensional copula function*, $C : [0,1]^n \rightarrow [0,1]$, *such that*:

$$H(x_1, \dots, x_n) = C(H_1(x_1), \dots, H_n(x_n)), \tag{1}$$

where $\boldsymbol{X} = (X_1, \dots, X_n)'$ *is an n-dimensional random vector. Then the joint density function can be given by*:

$$h(\boldsymbol{x}) = \prod_{i=1}^{n} h_n(x_n) \cdot c(H_1(x_1), \dots, H_n(x_n)), \quad \boldsymbol{x} \in \mathcal{R}^n \tag{2}$$

Where c is the copula density function. If all margins are continuous, then copula is unique.

Sklar's theorem states that one can model the joint density function as a product of the marginal's densities and the copula density. However, multivariate copulas impose the same dependence structures among variables, and only elliptical (Gaussian and t-student) copula models can be extended to multivariate cases. Vine copula incorporates the benefit of the copula models into a multivariate context. The vine copula models back to the idea of[28] and then received more interest development in[29]. The density of n-dimensional copula model can be expressed, using vine copula model, as $n(n-1)/2$ bivariate copulas (pair-copulas) densities. Bedford and Cooke[30] represent the vine copula as an unconnected graph structure known as a regular vine copula (R-vine). Due to the decomposition of the bivariate copulas, the vine copula models allow modeling two variables at a time. Each pair of variables can be modeled with a different choice of bivariate copulas; thereby, there is no restriction on the type of dependence among variables. Following the definition of the vine copula structure in[30], the formal definition of the vine copula structure can be given as follows:

Definition 1 The structure $\mathcal{V}$ is a regular vine on n variables if it meets the following conditions:

1. $\mathcal{T}_1$ is a tree with node set $V_1 = \{1, \dots, n\}$, and edge set $E_1 = n - 1$.

2. For $i = 2, \ldots, n-1$, $\mathcal{T}_i$ is a tree with node set $V_i = E_{i-1}$.
3. Two edges in $\mathcal{T}_i$ become a node in $\mathcal{T}_{i+1}$, if and only if they shared a common node in $\mathcal{T}_i$. This condition is known as `proximity condition`.

The structure $\mathcal{V} = (\mathcal{T}_1, \ldots, \mathcal{T}_i)$ is called a vine structure. If each edge in $\mathcal{V}$ is associated with a bivariate copula, then $\mathcal{V}$ is called a vine copula model or a pair-copula construction. The general class of the vine copula model is known as regular vine copula (R-vine copula). In the R-vine copula, there is no restriction on the way of connecting the variables. Variables can be connected by any possible shape following the three conditions given in Definition 1. There are two other sub-classes of the vine copula model, known as Canonical vine (C-vine) and Drawable vine (D-vine). These two sub-classes require a specific structure of the $\mathcal{V}$. For the C-vine, the variables at the first tree are connected concerning a particular variable; hence, it has a star shape. In the D-vine copula, the variables are connected sequentially, one variable after the other, taking a path shape. An example of a mixture of C-vine and D-vine copula is given in Example 2. For full details of the two sub-classes of the R-vine copula, we refer the reader to[31]. In Example 1 we introduce a simple 3-dimensional C-vine copula models (for 3-dimensional data set, the C-vine and D-vine copula models have the same vine structure).

Example 1 (Example of 3-dimensional C-vine copula model). Suppose a 3-dimensional random vector $\boldsymbol{X} = (X_1, X_2, X_3)'$ is given, where all the variables are continuous. Suppose further that H_1, H_2, H_3 are the corresponding univariate marginal distributions with their marginal density functions, h_1, h_2, h_3, and corresponding parameters $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\phi}_3$, respectively. Then, according to Sklar's theorem[27]. The joint density function, h, can be given as follows:

$$\begin{aligned} h(\boldsymbol{x}; \boldsymbol{\alpha}) = {} & c_{3,2}(H_3(x_3; \boldsymbol{\phi}_3), H_2(x_2; \boldsymbol{\phi}_2); \boldsymbol{\theta}_{3,2}) \cdot c_{2,1}(H_2(x_2; \boldsymbol{\phi}_2), H_1(x_1; \boldsymbol{\phi}_1); \boldsymbol{\theta}_{2,1}) \\ & \cdot c_{3,1|2}(H_{3|2}(x_3|x_2; \boldsymbol{\phi}_3, \boldsymbol{\phi}_2, \boldsymbol{\theta}_{3,2}), H_{1|2}(x_1|x_2; \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\theta}_{1,2}); x_2, \boldsymbol{\theta}_{3,1|2}), \end{aligned} \tag{3}$$

where $c_{3,2}$ is the density function of the bivariate copula c associated with the variables 3, and 2, and $\boldsymbol{\theta}_{3,2}$ its corresponding parameters. $c_{3,1|2}$ is the conditional density function of the conditional bivariate copula between the third and first variables conditioning on the second variable. We can see that the conditional copula, $c_{3,1|2}$ depends on the conditioning x_2. In most of the vine copula applications, and to reduce the model complexity, the $c_{3,1|2}$ assumed to be independent of the value of the x_2, and hence, called `simplified vine copula`. Then, the joint density in Eq. (3) can be rewritten as follows:

$$\begin{aligned} h(\boldsymbol{x}; \boldsymbol{\alpha}) = {} & c_{3,2}(H_3(x_3; \boldsymbol{\phi}_3), H_2(x_2; \boldsymbol{\phi}_2)) \cdot c_{2,1}(H_2(x_2; \boldsymbol{\phi}_2), H_1(x_1; \boldsymbol{\phi}_1)) \\ & \cdot c_{3,1|2}(H_{3|2}(x_3|x_2; \boldsymbol{\phi}_3, \boldsymbol{\phi}_2, \boldsymbol{\theta}_{3,2}), H_{1|2}(x_1|x_2; \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \boldsymbol{\theta}_{1,2})) \end{aligned} \tag{4}$$

The structure of the 3-dimensional C-vine copula model for this example is presented in Fig. 3.

For a n-dimensional vine copula model, the joint density function, h is given as follows:

$$\begin{aligned} h(\boldsymbol{x}; \boldsymbol{\alpha}) = {} & \prod_{j=1}^{n} h_j(x_j; \phi_j) \\ & \cdot \prod_{i=1}^{n-1} \prod_{e \in E_i} c_{e_m,e_k|D_e}(H_{e_m|D_e}(x_{e_m}|\boldsymbol{x}_{D_e}; \phi_{e_m|D_e}, \boldsymbol{\theta}_{e_m|D_e}), H_{e_k|D_e}(x_{e_k}|\boldsymbol{x}_{D_e}; \phi_{e_k|D_e}, \boldsymbol{\theta}_{e_k|D_e}); \boldsymbol{\theta}_{e_m,e_k|D_e}), \end{aligned} \tag{5}$$

where $\boldsymbol{\alpha}$ is the parametric vector of all the model parameters, $\mathbf{x}_{D_e}$ is a sub-vector of $\boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathcal{R}^n$ and D_e is the set of the conditioning variables. At the first tree, there are no conditioning variables; hence, D_e is an empty set in the first vine copula model. For $T_i, i = 1, \ldots, n-1, D_e = i - 1$. $H_{e_m|D_e}$ is the conditional distribution function of $X_{e_m}|\boldsymbol{X}_{\boldsymbol{D_d}}$, with the corresponding marginal $\boldsymbol{\phi}_{e_m|D_e}$, and the conditional bivariate copula parameters $\boldsymbol{\theta}_{e_m|D_e}$.

Finite mixture with vine copula model.

This section will briefly introduce the model-based clustering algorithm with the vine copula model using the ECM algorithm following the work of[7]. For more details, we refer to the latter reference. In addition, we will discuss the pruning and truncation technique for the mixture of the vine copula models proposed in this paper.

mixture of the vine copula model.

Finite mixture models assume that the data are generated from a mixture of g components, $g = 1, .., G$. Using an iterative algorithm, such as `ECM`, each observation is assigned to one of the mixture components with a probability. Incorporating the vine copula models into a mixture context adds numerous flexibility to the finite mixture models. The mixture of the vine copula models uncovers complex hidden bivariate dependence patterns among the variables. To define the mixture of the vine copula model formally, suppose that an n-dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_n)'$ is given. Suppose further that we draw t independent realization $= \boldsymbol{x_t} = (x_{t,1}, \ldots, x_{T,n})$, $t = 1, .., T$, from $\boldsymbol{X}$. Then we said that $\boldsymbol{X}$ is generated from a mixture of g-components R-vine copula densities, if its density is given as follows:

$$h(x; \boldsymbol{\delta}) = \sum_{g=1}^{G} \pi_g \cdot h_g(x; \boldsymbol{\alpha}_g), \tag{6}$$

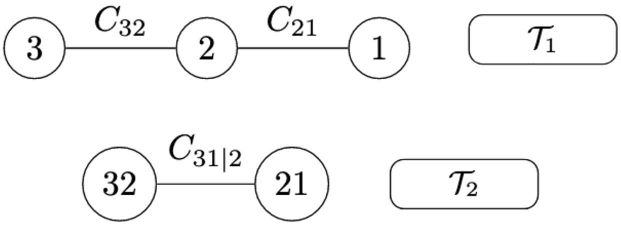

Figure 3. 3-dimensional C-vine copula.

where $\boldsymbol{\delta}$ is the parameters vector that contains all the mixture model parameters, and $\boldsymbol{\delta}_g = (\pi_g, \boldsymbol{\phi}_g, \boldsymbol{\theta}_g)$ the parameters vector of all the parameters of the gth component. $h_g(x; \boldsymbol{\alpha}_g)$ is the density of the gth component and π_g is the mixing proportion (mixture weight) that satisfies the following two conditions:

1. $\sum_{g=1}^{G} \pi_g = 1$
2. $0 < \pi_g < 1$

In this paper, we will use the Inference for margins (IFM) method of[32]. The IFM is a two-stage approach. In the first step, the marginal distribution is estimated parametrically. Then, the estimated margins parameters are used to estimate the copula parameters.

The flexibility of the mixture of the vine copula models comes with the cost of the complex computational process. However, pruning and truncating the mixture vine copula models recover this limitation and provide a great parameter reduction. In this paper, we incorporate the truncation method of[17]. In[17], the authors apply a new modified Bayesian Information Criteria (*mBICV*) of the traditional Bayesian Information Criteria (*BIC*) of[13] to select the optimal truncation level of the vine copula model. Determining the optimal truncation level in their method start by fitting a low truncation level and calculating the mBICV. Then, gradually add more vine copula trees until there is no improvement in the mBICV value. The *BIC*, and *mBICV* can be given as follows:

$$BIC = -2\ln l(\hat{\boldsymbol{\alpha}}) + p\ln(T) \tag{7}$$

$$mBICV = -2\ln(\hat{\boldsymbol{\theta}}) + \vartheta \ln(T) - 2\sum_{i=1}^{n-1}(q_i \ln(\varphi_0^i) - (n-i-q_i)\ln(1-\varphi_0^i)), \tag{8}$$

where $\hat{\boldsymbol{\delta}}$, is the estimated parameters of the model, T is the total number of observations, p is the total number of the model parameters, $\hat{\boldsymbol{\theta}}$ the estimated parameters of the bivariate copulas, ϑ, is the (effective) number of the model parameters, i is the tree level of the vine copula model, φ_0 is the prior probability that the bivariate copula is a non-independent copula, and q_i is the total number of non-independent bivariate copulas in tree i. For the pruning method, we use the independent test based on Kendall's tau introduced in[18]. In the following example, we will explain the idea of the mixture of the vine copula model with the pruning and truncation technique.

Example 2 (A two-components mixture of 4-dimensional vine copula mixture model). Assume that a data set is generated from two components 4-dimensional random vectors $\boldsymbol{X}_1 = (X_{11}, X_{21}, X_{31}, X_{41})$, and $\boldsymbol{X}_2 = (X_{12}, X_{22}, X_{32}, X_{42})$ and are given. Suppose further that t independent realization, $\boldsymbol{x_t} = (x_{t,1}, \ldots, x_{T,n})$, $t = 1, .., T$, are drawn from $\boldsymbol{X_1}, \boldsymbol{X_2}$, respectively. Figure 4 represents a two-component mixture of vine copula. From the figure, one can see that each component has its own vine structure. The first component follows the D-vine copula structure, while the second component is a C-vine copula structure. $\mathcal{T}_{11}$, $\mathcal{T}_{12}$, and $\mathcal{T}_{13}$ represent the trees of the first vine structure, whereas $\mathcal{T}_{12}$, $\mathcal{T}_{22}$, and $\mathcal{T}_{32}$ refer to the trees of the second vine structure. For each vine copula model, only the first two trees are fitted with (different) bivariate copulas. For the third tree of each component, independent bivariate copulas are specified. Hence, each component is truncated at the second tree. Moreover, at the first two trees of each component, some pairs are associated with independent copulas, representing the pruning method. Then the density of the two-component mixture of the vine copula model can be given as:

$$h(x; \boldsymbol{\delta}) = \pi_1 \cdot h_1(x; \boldsymbol{\alpha}_1) + \pi_2 \cdot h_2(x; \boldsymbol{\alpha}_2) \tag{9}$$

$h_1(x; \boldsymbol{\alpha}_1), h_2(x; \boldsymbol{\alpha}_2)$ can be given in a similar way as in Eq. (5).

ECM algorithm. The parameters of the mixture model are usually estimated using iterative methods, such as Expectation Maximization (EM) algorithm[33] and Expected Conditional Maximum (ECM) algorithm[9]. In the general situation, estimating the model parameters can be given by computing the the value that maximizes the log-likelihood of the given data is as follows:

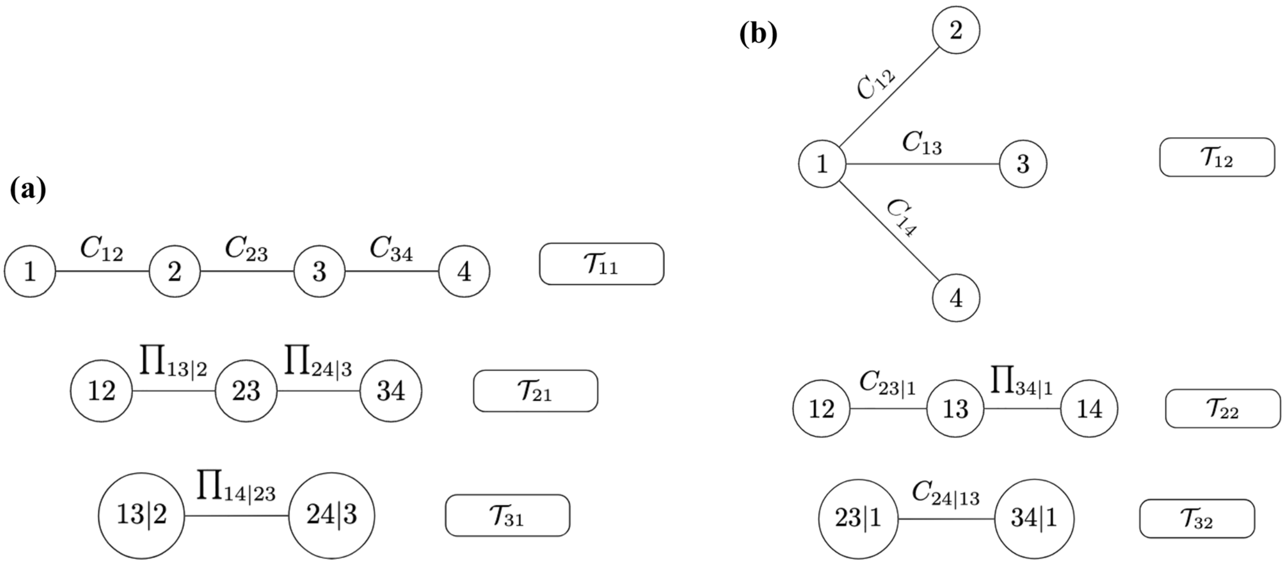

Figure 4. Two-component mixture of R-vine copula densities. (**a**) The left panel represents a 4-dimensional D-vine copula as the first cluster of the mixture model. (**b**) The right panel represents a 4-dimensional C-vine copula as the second cluster of the model.

$$l(\boldsymbol{\delta}; x) = \ln \prod_{t=1}^{T} h(x_t; \boldsymbol{\alpha}) = \ln \prod_{t=1}^{T} \sum_{g=1}^{G} \pi_g \cdot h_g(x_t; \boldsymbol{\alpha}_g). \tag{10}$$

However, since the true label of the data is unknown, the EM algorithm treats the data as incomplete data and introduces latent variables $\boldsymbol{z}_t = (z_{t,1}, \ldots, z_{T,g})'$. $z_{t,g} = 1$ if the x_t belongs to the g^{th} component and $z_{t,g} = 0$ otherwise, and the random vector $\boldsymbol{Z}_t$ follows multinomial distribution, such that: $\boldsymbol{Z}_t \sim Mult(1, (\pi_1, \ldots, \pi_g))$. Therefore, we can define the complete data as $x_c = (x_t, \boldsymbol{z}_t)'$. Hence, the log-likelihood of the complete data can be given by:

$$l_c(\boldsymbol{\delta}; \boldsymbol{z}, x) = \ln \prod_{t=1}^{T} \prod_{g=1}^{G} \left[\pi_g \cdot h_g(x_t; \boldsymbol{\alpha}_g)\right]^{z_{t,g}} = \sum_{t=1}^{T} \sum_{g=1}^{G} z_{t,g} \cdot \ln \pi_g + \sum_{t=1}^{T} \sum_{g=1}^{G} z_{t,g} \cdot \ln h_g(x_t; \boldsymbol{\alpha}_g), \tag{11}$$

where $h_g(\boldsymbol{x}_t; \boldsymbol{\alpha}_g)$ is given in Eq. (5). EM-algorithm is commonly used in the mixture literature. The E-step computes the conditional expectation of the log-likelihood of the complete data, given the observed data at the current estimation of the model parameters. The M-step, then, maximizes the expected log-likelihood from the E-step over all the model parameters. The iterations are continuous till the model converges. However, in the vine copula model, the joint estimation of the marginal parameters, bivariate copula parameters, and mixture weight parameters of the g^{th} component is not tractable and efficient[7]. Therefore, the[7] adapted the ECM algorithm with the mixture of the vine copula models. `ECM` algorithm divided the `M`-step of the *EM* algorithm into three lower dimensional steps called CM-steps. A brief introduction, following[7], of the `CM`-steps in the mixture of the vine copula models can be given as follows:

- `E-step`: This step calculates the posterior probability that an observation $\mathbf{x_i}$ belongs to the g^{th} mixture component given the current value of the mixture weight, π_g^s, and $\boldsymbol{\alpha}_g^s$, where s indicates the first iteration:

$$r_{t,g}^{(s+1)} = \frac{\pi_g^{(s)}\ h_g(\mathbf{x}_t; \boldsymbol{\alpha}_g^{(s)})}{\sum_{g'=1}^{G} \pi_{g'}^{(s)}\ h_g(\mathbf{x}_t; \boldsymbol{\alpha}_{g'}^{(s)})} \tag{12}$$

  for $t = 1, \ldots, T$, and $g = 1, \ldots, G$.
- `CM-step 1`: (update the mixture weights): Maximize $l_c(\boldsymbol{\delta}; \mathbf{z}, \mathbf{x})$ over the mixture weights π_g given $r_{t,g}^{(s+1)}$, such that:

$$\pi_g^{(s+1)} = arg\ max_{\pi_g} \sum_{t=1}^{T} r_{t,g}^{(s+1)}\ \cdot\ \ln \pi_g \tag{13}$$

  A closed form solution of $\pi_g^{(s+1)}$ exists and can be given as:

$$\pi_g^{(s+1)} = \frac{\sum_{t=1}^{T} r_{t,g}^{(s+1)}}{T}, \quad g = 1, \ldots, G. \tag{14}$$

- `CM-step 2` (update the marginal parameters): Maximize $l_c(\boldsymbol{\delta}; \mathbf{z}, \mathbf{x})$ over the marginal parameters $\boldsymbol{\phi}_g$ given the current value of the bivariate copula parameters $\boldsymbol{\theta}_g^{(s)}$, and $r_{i,g}^{(s+1)}$:

$$\boldsymbol{\phi}_g^* = arg\ max_{\boldsymbol{\phi}_g} \sum_{t=1}^{T} r_{t,g}^{(s+1)} \cdot \ln h_g(\mathbf{x}_t; \boldsymbol{\phi}_g, \boldsymbol{\theta}_g^{(s)}) \tag{15}$$

for $g = 1, \ldots, G$. $\boldsymbol{\phi}_h^*$ is the optimal marginal parameter estimate of the g^{th} component. Since a closed-form solution does not exist, the $l_c(\boldsymbol{\delta}; \mathbf{z}, \mathbf{x})$ is maximized numerically over $\boldsymbol{\phi}_{\mathbf{g}}$ such that:

$$\boldsymbol{\phi}_g^{(s+1)} = max_{\boldsymbol{\phi}_g} \sum_{t=1}^{T} r_{t,g}^{(s+1)} \cdot \ln h_g(\mathbf{x}_t; \boldsymbol{\phi}_g, \boldsymbol{\theta}_g^{(s)}) \tag{16}$$

- `CM-step 3` (update the bivariate copula parameters): Similar to the marginal parameters, a closed-form solution that maximizes $l_c(\boldsymbol{\delta}; \mathbf{z}, \mathbf{x})$ 6over $\boldsymbol{\theta}_g$ given $\boldsymbol{\phi}_g^{(s+1)}$ and $r_{t,g}^{(s+1)}$ does not exist. Thus, $l_c(\boldsymbol{\delta}; \mathbf{z}, \mathbf{x})$ will be maximized numerically over $\boldsymbol{\theta}_g$, such that:

$$\boldsymbol{\theta}_g^{(s+1)} = max_{\boldsymbol{\theta}_g} \sum_{t=1}^{T} r_{t,g}^{(s+1)} \cdot \ln h_g(\mathbf{x}_t; \boldsymbol{\phi}_g^{(s+1)}, \boldsymbol{\theta}_g) \tag{17}$$

for $g = 1, \ldots, G$.

In this paper, the truncation and pruning techniques are applied to each model cluster individually. Therefore, the steps of the present work differ from the work of[7] in the second and final step. Unlike the work of[7], at the second step, no prior truncation level is determined. For the last step, truncation and pruning techniques are applied individually for each cluster. The steps of the proposed pruning and truncation method of this paper can be divided into the following steps:

1. Cluster the original data using `k-means` (other clustering methods are possible).
2. Obtains the copula data for each cluster from step 1.
3. Fit vine copula model for each cluster and determine the truncation and pruning pairs. For the vine structure and pair-copula selection, we use the Akaike Information Criteria (AIC) of[12]. AIC can be given as follows:

$$AIC = -2l(\hat{\boldsymbol{\theta}}) + 2p, \tag{18}$$

   where $\hat{\boldsymbol{\theta}}$, and p are the estimated parameters of the bivariate copulas and the number of the model parameters, respectively.
4. Run the ECM algorithm using the cluster data from step 1 and the vine copula model from step 2.
5. Re-clustering the data based on the ECM successive steps.
6. Fit vine copula model and determine the truncation and pruning pairs for each cluster.

To test the model performance, we use the BIC and misclassification rate. The best-fitted model is selected based on the lower BIC or misclassification rate.

Data availability

R software version 4.2.1 (R Development Core Team 2022) was used to implement the proposed methods. The R-package "vineclust" (https://github.com/oezgesahin/vineclust) and "rvinecopulib" were mainly used in this paper. Moreover, several dependent key packages were used, such as "mclust" and "VineCopula" packages. The Diabetes, Banknote data sets are available in the "mclust" package (https://cran.r-project.org/web/packages/mclust/vignettes/mclust.html). The flea data set is available in the "fdm2id" (https://cran.r-project.org/web/packages/fdm2id/index.html) package. The Sonar data set is available in the "mlbench" package from (https://rdrr.io/cran/mlbench/man/Sonar.html).

Received: 30 August 2022; Accepted: 14 November 2022
Published online: 17 November 2022

References

1. Dias, J. G., Vermunt, J. K. & Ramos, S. Mixture hidden Markov models in finance research. In *Advances in Data Analysis, Data Handling and Business Intelligence*, 451–459 (Springer, 2009).
2. Mateen, M., Wen, J., Song, S. & Huang, Z. Fundus image classification using VGG-19 architecture with PCA and SVD. *Symmetry* https://doi.org/10.3390/sym11010001 (2019).
3. Maliuk, A. S., Prosvirin, A. E., Ahmad, Z., Kim, C. H. & Kim, J.-M. Novel bearing fault diagnosis using gaussian mixture model-based fault band selection. *Sensors* https://doi.org/10.3390/s21196579 (2021).
4. Kim, D., Kim, J.-M., Liao, S.-M. & Jung, Y.-S. Mixture of d-vine copulas for modeling dependence. *Comput. Stat. Data Anal.* **64**, 1–19 (2013).
5. Roy, A. & Parui, S. K. Pair-copula based mixture models and their application in clustering. *Pattern Recogn.* **47**, 1689–1697 (2014).
6. Alanazi, F. A. A mixture of regular vines for multiple dependencies. *J. Probab. Stat.* **2021**, 1–15 (2021).

7. Sahin, Ö. & Czado, C. Vine copula mixture models and clustering for non-gaussian data. *Econom. Stat.* **22**, 136–158 (2022).
8. Hartigan, J. A. & Wong, M. A. Algorithm as 136: A k-means clustering algorithm. *J. R. Stat. Soc. Ser. (Appl. Stat.)* **28**, 100–108 (1979).
9. Meng, X.-L. & Rubin, D. B. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* **80**, 267–278 (1993).
10. Alanazi, F. A. International Journal of Mathematics and Mathematical Sciences; New York Vol. 2021 https://doi.org/10.1155/2021/3214262 (2021).
11. Brechmann, E. C. & Joe, H. Truncation of vine copulas using fit indices. *J. Multivar. Anal.* **138**, 19–33 (2015).
12. Akaike, H. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, 199–213 (Springer, 1998).
13. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **6**(2), 461–464. http://www.jstor.org/stable/2958889 (1978).
14. Bozdogan, H. Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika* **52**, 345–370 (1987).
15. Celeux, G. & Soromenho, G. An entropy criterion for assessing the number of clusters in a mixture model. *J. Classif.* **13**, 195–212 (1996).
16. Claeskens, G., Hjort, N. L. *et al. Model Selection and Model Averaging*. Cambridge Books (2008).
17. Nagler, T., Bumann, C. & Czado, C. Model selection in sparse high-dimensional vine copula models with an application to portfolio risk. *J. Multivar. Anal.* **172**, 180–192 (2019).
18. Dissmann, J., Brechmann, E. C., Czado, C. & Kurowicka, D. Selecting and estimating regular vine copulae and application to financial returns. *Comput. Stat. Data Anal.* **59**, 52–69 (2013).
19. Sahin, Ö. vineclust: Model-based clustering with vine copulas. https://github.com/oezgesahin/vineclust (2022).
20. R Core Team. *R: A Language and Environment for Statistical Computing* (R Foundation for Statistical Computing, 2013).
21. Nagler, T. et al. VineCopula: Statistical Inference of Vine Copulas (2022). R package version 2.4.4.
22. Scrucca, L., Fop, M., Murphy, T. B. & Raftery, A. E. mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *R J.* **8**, 289–317 (2016).
23. Billio, M., Frattarolo, L. & Guégan, D. High-dimensional radial symmetry of copula functions: Multiplier bootstrap versus randomization. *Symmetry* https://doi.org/10.3390/sym14010097 (2022).
24. Kollo, T., Käärik, M. & Selart, A. Multivariate skew t-distribution: Asymptotics for parameter estimators and extension to skew t-copula. *Symmetry* https://doi.org/10.3390/sym13061059 (2021).
25. Li, Q. & Zhang, T. Research on the reliability of bridge structure construction process system based on copula theory. *Appl. Sci.* https://doi.org/10.3390/app12168137 (2022).
26. Nonvignon, T. Z., Boucif, A. B. & Mhamed, M. A copula-based attack prediction model for vehicle-to-grid networks. *Appl. Sci.* https://doi.org/10.3390/app12083830 (2022).
27. Sklar, M. Fonctions de repartition an dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris* **8**, 229–231 (1959).
28. Joe, H. Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters. *Lect. Notes-Monogr. Ser.* **28**, 120–141. http://www.jstor.org/stable/4355888 (1996).
29. Bedford, T. & Cooke, R. M. Probability density decomposition for conditionally dependent random variables modeled by vines. *Ann. Math. Artif. Intell.* **32**, 245–268 (2001).
30. Bedford, T. & Cooke, R. M. Vines-a new graphical model for dependent random variables. *Ann. Stat.* **30**, 1031–1068 (2002).
31. Aas, K., Czado, C., Frigessi, A. & Bakken, H. Pair-copula constructions of multiple dependence. *Insur.: Math. Econ.* **44**, 182–198 (2009).
32. Joe, H. & Xu, J. J. The estimation method of inference functions for margins for multivariate models. *R. Faculty Research and Publications* https://doi.org/10.14288/1.0225985 (1996).
33. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.: Ser. B (Methodol.)* **39**, 1–22 (1977).

Acknowledgements

The author acknowledges that the APC was funded by Prince Sultan University through the TAS lab.

Author contributions

This is a sole authorship manuscript.

Competing interests

The author declares no competing interests.

Additional information

Correspondence and requests for materials should be addressed to F.A.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.