STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **11**, September 2023, pp 853–864.
Published online in International Academic Press (www.IAPress.org)

Copula based learning for directed acyclic graphs

Rusul Mohsin Moharib Alsarray and Vahid Rezaeitabar*

Department of statistics, Faculty of Statistics, Matematics & Computer, Allameh Tabataba'i University, Tehran, Iran

Abstract We provide the learning of a DAG model arising from high dimensional random variables following both normal and non-normal assumptions. To this end, the copula function utilized connecting dependent variables. Moreover to normal copula, the three most applicable copulas have been investigated modeling all three dependence structures negative, positive, and weak kinds. The copula functions, FGM, Clayton, and Gumbel are considered coving these situations and their detailed calculations are also presented. In addition, the structure function has been exactly determined due to choosing a good copula model based on statistical software R with respect to any assumed direction among all nodes. The direction with the maximum structure function has been preferred. The corresponding algorithms finding these directions and the maximization procedures are also provided. Finally, some extensive tabulations and simulation studies are provided, and in the following to have a clear thought of provided strategies, a real world application has been analyzed.

Keywords Continuous models, Copula functions, Directed acyclic graph, Direction selection, Structure function.

AMS 2010 subject classifications

DOI: 10.19139/soic-2310-5070-1634

1. Introduction

multivariate analysis has arisen in lots of real-world applications. It is also can be used in many aspects of data sciences and new statistical areas such as graphs, quality controls, network analysis, and so on. Up to now, lots of improvements have been made in multivariate statistical models. Understanding all the complex connections and multivariate dependencies that exist in a data set, developing suitable models, and improving inference procedures are imperative challenges in modern statistical fields. In this regard, the graphical models that recently providing an innovative field can be served as powerful instruments utilizing to discovering data structures. In another point of view, the graphical models are one of the multivariate models relating to the joint densities of a family of variables confined by some conditional independence assumptions and in addition, holding the conditional relation between corresponding random variables which are denoted by a graph. For more information about the definition, structure, performance, and usefulness of a graph see [7,8].

Among the numerous kinds of all graphs, directed acyclic graph (DAG) attracted powerful literatures from probabilistic models. This term was firstly applied in [18,19] and known by the term DAG in statistical areas [11,15] and bayesian network in artifical intelligence disiplines [11,20]. Data based learning for DAG models are investigated by several researchers [4,15]. It is also worth mentioning that all of the afromentioned investigations have been done in accordance with two concepts including score based learning and search procedure. In the case of multinomial descrete random variables, there were provided a metrics for domains conditioning samples in [10]. The scoring metrics for continous variables following the multivariate normal distribution are given in [6]. A

*Correspondence to: Vahid Rezaeitabar (Email: vhrezaei@atu.ac.ir).
Department of statistics, Faculty of Statistics, Matematics & Computer, Allameh Tabataba'i University, Tehran, Iran.
School of biological sciences, IPM, Tehran, Iran.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

combination learning for two previous works using a general metric for both of discrete and continous variables was studied in [9].

Learning DAGs are a type of machine learning algorithm that is used for causal inference. DAGs are graphs that represent the causal relationships between different variables in a dataset. In a DAG, the nodes represent the variables, and the edges represent the causal relationships between them. The goal of learning DAGs is to identify the causal relationships between the variables in a dataset. This is important because it allows us to determine which variables are causing changes in other variables in the dataset, and which variables are being influenced by other variables. There are several methods for learning DAGs, including constraint based methods, score-based methods, and hybrid methods. Constraint-based methods involve testing conditional independence relationships between variables, while score-based methods involve optimizing a score function that measures the goodness-of-fit of the DAG to the data. Hybrid methods combine elements of both constraint-based and score-based methods. Overall, learning DAGs is a powerful tool for understanding the causal relationships between variables in a dataset, and is used in a wide range of applications, including healthcare, finance, and social sciences. The process of learning DAGs involves using observational or experimental data to infer the causal relationships between variables. This is done by using statistical algorithms to search through all possible DAGs that could explain the data and identify the one that best fits the observed relationships between variables. Learning DAGs is useful in a variety of applications, such as predicting the effect of interventions in complex systems, identifying risk factors for diseases, and optimizing decision-making processes. However, it can be challenging because it requires a deep understanding of statistical concepts and the ability to interpret complex graphs. Overall, learning DAGs is a powerful tool for understanding complex systems and making informed decisions based on causal relationships between variables. In this paper, we develop the learning method using copula functions that is more efficient and sufficient to previous and mentioned studies.

One of the most interesting quantities in the modeling of DAGs is determining their structure. The problem is we can not calculate the structure function for a DAG before we know its model. As it is clear, on one hand, the value of structure function is so important and can play a useful role in determining the model of a DAG, and in another hand, we must identify the model and afterward calculate the mentioned function. To this end, we provide a strategy that chooses the best model of a DAG, based on maximizing structure function calculating according to some possible dependency joint densities. Precisely, at first, the good copula function is fitted on the variables or some suitable copula functions have been selected for suitable model presenting of the corresponding random variables. Secondly, with the assumption of a fixed copula that is selected in the previous step, we maximize structure function among all the possible models. Precisely, with a fixed copula function, we can calculate the exact value of structure function, and then, the model with maximized structure function can be selected.

To the best of our knowledge, score based learning has been extensively discussed in[13,2]. According to the nonconvex and combinational features in that way, it is rarely useful in the existence of high dimensional data, particularly when the considering parameters are greater than the sample size. The problem has remained as a gap in literature until providing some investigation in [1,3]. Another applicable strategy for learning DAG, called Markov Chain Monte Carlo (MCMC) procedures has been deeply discussed in [14,12,8,11,7,5]. For all the above methods, the ordering variables does not consider and it is clear that this issue can be regarded in many real world application disciplines like genetics, finance and so on. Learning DAGs for such fields is limited to determining the parent random variables and extensively provided in [16,3,20]. In this manner, we present innovative learning for DAGs in accordance with maximizing all of the possible structure functions through a search algorithm.

The rest of the paper is organized as follows. The copula function aiming to model dependency among random variables is provided in section 2. The fitting procedures of a copula function in some variables and investigation of its performances are also provided. In section 3, we review the structure function of a DAG model. In addition, the calculating method for this function is described in detail. The structure function is derived in accordance with the fitted copula function and for any possible model, the values of this function can be exactly calculated. Selecting the best model for a mentioned DAG is the main task of section 4. The search algorithm and maximizing structure function are also provided in this section. Extensive simulation studies finding a DAG model in lots of cases are presented in section 5. Their performances have been deeply taken under investigation and finally, the conclusion of our study is given in section 6.

2. Copula function

Here, we proposed some preliminary notions that utilized in the study. First of all, consider a $d-$dimensional $X = (X_1, X_2, \ldots, X_d)$, representing the DAG we are going to assess. The observed or realization values of random vector $X = (X_1, X_2, \ldots, X_d)$ is considered with notions $x = (x_1, x_2, \ldots, x_d)$. It is worth mentioning that n represent the number of sampling and consequently each $x_i, i = 1, 2, \ldots, d$ is a $n-$dimensional column vector collected during the sampling process. More precisely, the collected data consist of a $n \times d$ matrix denoted by $\underline{x}$ as the form of

$$\underline{x} = \begin{bmatrix} x_{11} & x_{21} & \ldots & x_{d1} \\ x_{12} & x_{22} & \ldots & x_{d2} \\ . & . & \ldots & . \\ x_{1n} & x_{2n} & \ldots & x_{dn} \end{bmatrix}$$

Suppose that $X \sim P_X(.)$, including the factorizing distribution regarding the DAG structure G:

$$\begin{aligned} P_X(x_1, x_2, \ldots, x_d; G) &= \prod_{j=1}^{d} p(x_j | x_1, x_2, \ldots, x_{j-1}) \\ &= \prod_{j=1}^{d} P(x_j | x_{pa(j)}), \end{aligned} \tag{1}$$

where $pa(j)$ is the set of parent nodes of the $j-$th node in G ([21]). In fact, $pa(X_i) \subseteq \sqcap_i \in \{X_1, X_2, \ldots, X_{i-1}\}$ is a set of random variables rendering X_i and $\{X_1, X_2, \ldots, X_{i-1}\}$ that are conditionally independent. The hidden assumption that exists in relation (1), is the statistical independence properties of the nodes.

In all previous studies, the joint model of these variables is normally considered. The whole discussions and investigations are also constructed around these strong assumptions. It is obvious that in the real world analysis, the joint model is not exactly determined and moreover, there are many real data sets that did not follow the normal assumption. Now, we set ourselves to generalize this assumption into a general continuous family of joint distributions. The dependent structure is also generally considered. The dependency among variables is mostly held in real data analysis and additionally, it has several kinds of structures. To this end, we utilize the copula functions that are so so useful in modeling dependent variables. In the present study, the copula function is provided modeling the dependence structure among the nodes. Up to now, this assumption is innovative. At first, we present the basic definitions and preliminary relations due to this function. To continue, the definition of copula function is presented and we are going to represent alternative of relation (1), in such a dependent situation.

Definition 1. Copula function: A copula function $C(.)$ satisfies in the following relations:

I: $C(u_1, u_2, \ldots, u_d) = 0$ if at least one $u_j = 0$.
II: $C(1, 1, \ldots, 1, u_j, 1, \ldots, 1, 1) = u_j$ if at most one $u_j \neq 1$.
III:

$$\int_B dC(u) = \sum_{\mathbf{z} \in \times_{i=1}^{d} \{x_i, y_i\}} (-1)^{N(\mathbf{z})} C(\mathbf{z}) \geq 0,$$

There exist many different types of copula functions and extensively covered all dependency models. For more pieces of information see [10]. The joint density of variables $X_1, X_2, \ldots, X_d$ using the copula function can be represented as

$$P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_n \leq x_n) = C(F_{X_1}(x_1), F_{X_2}(x_2), \ldots, F_{X_n}(x_n)),$$

where $F_{X_i}(.), i = 1, 2, \ldots, d$ stands for the marginal cumulative distribution function and it is easy to check that the independent model can be modeled using independent copula meaning that

$$C(u_1, u_2, \ldots, u_d) = \prod_{j=1}^{d} u_j.$$

The relation (1) may be noticed by Bayesian networks in artificial intelligence literature and DAG in statistical studies. Learning a Bayesian network requires to understand the structure of the model. Similarly, the conditional probability that captures $P(x_j|x_{pa(j)})$ should be learned. Accordingly, before dealing with the learning problems, we should choose a suitable copula function. Afterward, the corresponding conditional independence structure must be determined. Finally, the direction among nodes should be identified. Choosing a good copula function that can provide suitable goodness of fit for a given data set is investigated in the future section.

3. Multivariate analysis

In this section, we are going to provide our strategy to find a suitable joint model throughout a copula function due to the corresponding samples. The copula model can model dependency among variables and dependency between random variables can be as three positive, weak, and negative forms.

Here, we don't want to focus on the copula model and the main goal is to use this useful function to reach the best learning of a DAG. Therefore, we present 3 famous copulas covering three mentioned dependencies. It is easy to check that Gumbel, FGM, and Clayton copulas can include all dependency situations. In addition, we also consider the normal copula as an important model that is extensively utilized in the Bayesian network.

Accordingly, we should assess the performances of these copulas based on a given data set, and then a copula model is determined as a suitable model for the corresponding variables. Taking a good approach for our selection, we can use the helpful command "gof" in useful package "gofCopula" from statistical software R [17], computing for a given dataset and according to the choices of the user different tests for different copulae. The corresponding relations for the aforementioned copulae are as follows:

Definition 2. Gumbel copula: If $C(.)$ in (1) has the following form:

$$C_{\mathrm{G}}(u_1, u_2, \ldots, u_n) = \exp[-(\sum_{i=1}^{n}(-\log(u_i))^{\theta})^{\frac{1}{\theta}}].$$

We say it Gumbel copula with parameters $\theta \geq 1$. When $\theta = 1$, it is reduced to an independent copula.

Definition 3. Clayton copula: If $C(.)$ in (1) has the following form:

$$C_{\mathrm{C}}(u_1, u_2, \ldots, u_n) = [1 + \sum_{i=1}^{n}(u_i^{-\theta} - 1)]^{-\frac{1}{\theta}}.$$

We say it Clayton copula with parameters $\theta \geq 1, \theta \neq 0$. When $\theta \to 0$, it is reduced to an independent copula.

Definition 4. FGM copula: If $C(.)$ in (1) has the following form:

$$C_{\mathrm{FGM}}(u_1, u_2, \ldots, u_n) = \prod_{j=1}^{n} u_j(1 + \theta \prod_{j=1}^{n}(1 - u_j)).$$

We say it FGM copula with parameters $-1 \leq \theta \leq 1, \theta$. When $\theta = 0$, it is reduced to an independent copula.

Definition 5. Normal copula: If $C(.)$ in (1) has the following form:

$$C_{\mathrm{N}}(u_1, u_2, \ldots, u_n) = \Phi_R^n(\Phi_{U_1}^{-1}(u_1), \Phi_{U_2}^{-1}(u_2), \ldots, \Phi_{U_n}^{-1}(u_n)).$$

We say it Normal copula of n dimension with coefficient correlation matrix

$$R = \begin{bmatrix} Cov(U_1, U_2) & Cov(U_1, U_3) & \ldots & Cov(U_1, U_n) \\ Cov(U_2, U_1) & Cov(U_2, U_2) & \ldots & Cov(U_2, U_n) \\ . & . & \ldots & . \\ Cov(U_n, U_1) & Cov(U_n, U_2) & \ldots & Cov(U_n, U_n) \end{bmatrix},$$

where Φ^{-1} is the inverse cumulative distribution function of a standard normal or precisely $\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{u} e^{-\frac{x^2}{2}} dx$ and Φ_R^n is the joint cumulative distribution function of a multivariate normal distribution with mean vector zero and covariance matrix equal to R. It is worth mentioning that when $R = 0$, it is reduced to an independent copula.

To continue, utilizing the useful package "gofCopula" we can satisfy the dependency model of observed data with the four mentioned copula. These functions, generally covered most of the applicable real data dependencies, and furthermore using the command "gof" can also improve the determination of selected copulas. The selection copula is not our attention, and we are going to provide general structures that can include not only the normal dependency but also most of the dependency models like negative and positive ones. For all of the aforementioned copulas, we present corresponding learning strategies and both theoretical and applicational backgrounds are also investigated.

Using the copula functions, the three most used kinds of these functions including the Farlie-Gumbel-Morgenstern (FGM) copula, the Gumbel copula, and the Clayton copula are provided and corresponding learning methods for dependence structures following these models are investigated. The comparison results of these copulas are explained through Kendall's τ correlation coefficient (rank dependent correlation coefficient). We will present learning a DAG not only for different models following these non-normal dependence structures but also for the models with the assumption of the multivariate normal model. Choosing the normal model is clear and it is discussed previously by many scholars since the assumption is trivial and applicable. We also consider three copula functions, the FGM copula, the Gumbel copula, and the Clayton copula, which have been very useful independence modeling. The FGM can accommodate relatively weak dependence between nodes and the Gumbel copula is well suited for the case when there is strong right tail dependence and the Clayton copula can be used when the correlation between marginal variables exhibits a strong left tail dependence.

4. Learning DAGs

Accordingly, to the previous section, the dependence structure of variables is known. In the following, we are going to determine a learning concept for a pre-specified DAG. Learning will be explained for all of the mentioned copulas and their details will be extensively discussed. Here assume that the copula function with a good fitting to the variables set $X_1, X_2, \ldots, X_d$ is demonstrated by C^*.

The graph structure G described in (1) is determined as C^*. Consequently, this relation can be exactly calculated in any direction among the nodes. To this end, the direction should be firstly assumed and the structure-function should be exactly determined. Since, there are many possible directions, the direction with maximum structure function can be selected as our goal. The process is given as follows.

I: Fit data set to copula functions normal, Clayton, Gumbel, and FGM and select the best one according to their test power called by C^*.
II: Calculate the structure values assuming C^* for all possible DAGs considering the whole direction.
III: Select a graph with the largest structure values demonstrated by our selected DAG.

Assuming $X_1, X_2, \ldots, X_d$ following the selected copula function C^* meaning that

$$P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_d \leq x_d) = C^*(F_{X_1}(x_1), F_{X_2}(x_2), \ldots, F_{X_d}(x_d)),$$

and consequently, the corresponding joint density is given by:

$$f_{X_1,X_2,\ldots,X_d}(x_1, x_2, \ldots, x_d) = \prod_{j=1}^{d} f_{X_j}(x_j) c^*(F_{X_1}(x_1), F_{X_2}(x_2), \ldots, F_{X_d}(x_d)),$$

Using the chain rule of probability, the other form of joint density can be derived as:

$$f_{X_1,X_2,\ldots,X_d}(x_1,x_2,\ldots,x_d)=\prod_{j=1}^{d} f_{X_j|X_1,X_2,\ldots,X_{j-1},X_{j+1},\ldots,X_d}(x_j).$$

But, X_j may conditionally be independent of some nodes. Let $I_{ij}, j, i = 1, 2, \ldots, d.$ be a bivariate quantity equal to zero when X_j is not of parents for node X_i or precisely

$$I_{ij}=\begin{cases}1 & X_j \textit{ is a parents for the node } X_i;\\ 0 & X_j \textit{ is not of parents for the node } X_i.\end{cases}$$

Hence, the joint density can be rewritten as the following form:

$$\begin{aligned}f_{X_1,X_2,\ldots,X_d}(x_1,x_2,\ldots,x_d)&=\prod_{j=1}^{d} f_{X_j|X_1,X_2,\ldots,X_{j-1}}(x_j)\\&=\prod_{j=1}^{d} f_{X_j|\Pi_j}(x_j),\end{aligned}$$

where Π_j be a set of all variables conditionally dependent to node X_j or

$$\Pi_j=\{X_i\in\{X_1,X_2,\ldots,X_j\}|I_{ij}=1\}.$$

To the best of your knowledge about our strategy, we provide a wide and general examples below.

Example 1. Dual nodes: This example deals with the variables priority i.e, in the first stage, we should determine the variables we want to start or learning method. Consider two nodes X_1 and X_2 with corresponding joint density $f_{X_1,X_2}(x_1,x_2)=f_{X_1}(x_1)f_{X_2}(x_2)c_2^*(F_{X_1}(x_1),F_{X_2}(x_2))$, and we are going to prefer among all possible directions demonstrated in figures (1), and (2).

The priority of node X_1 to the other nodes (P_1) are shown in figure (1) and conversely the priority of X_2 to the other nodes (P_2) are shown in figure (2). The initial problem without any special assumptions of the model is a decision about these cases. Accordingly, the quantities P_1, and P_2 can be respectively defined as:

$$P_1=\int_{\mathbb{R}}\frac{f_{X_1}(x)}{f_{X_2}(x)}dx,$$

and

$$P_2=\int_{\mathbb{R}}\frac{f_{X_2}(x)}{f_{X_1}(x)}dx.$$

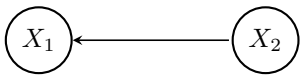

Figure 1. Case I

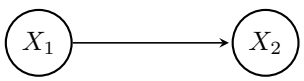

Figure 2. Case II

If $P_1 \geq P_2$ the case I, is preferred and vice versa if $P_1 \leq P_2$, the case II is selected. In fact, P_1 is a measure validating the whole affection that can be received by node X_1 from all other nodes, and can be extended for

d-dimensional as the following form:

$$\begin{aligned} P_i &= \int_{\mathbb{R}} \frac{f_{X_i}(x)}{f_{X_1,X_2,\ldots,X_{i-1},X_{i+1},\ldots,X_d|X_i}(x)}dx \\ &= \int_{\mathbb{R}} \frac{f^2_{X_i}(x)}{f_{X_1,X_2,\ldots,X_d}(x)}dx \\ &= \int_{\mathbb{R}} \frac{f^2_{X_i}(x)}{c^*_d(x,x,\ldots,x)}dx \\ &= \int_{\mathbb{R}} \frac{f^2_{X_i}(x)}{\frac{\partial^d}{\partial x^d}C^*_d(x,x,\ldots,x)}dx, d = 3,4,\ldots. \end{aligned}$$

The node with a maximum value of P_is is our learning priority and in the same manner, we can continue learning the concept. The next example deals with the problem of finding suitable variables that can directly connect with the first node, which is selected in the first step.

Example 2. Triple nodes: In this example, there are three variables that have been considered regarding any dependence assumption and we are going to investigate an applicable learning issue. Firstly, as can be seen in Figures (3), and (4), we should determine one variable in accordance with the start of a learning algorithm. In previous pieces of literature not only there only assumed variables following the normal assumptions but also they start with an initial number 1 meaning that their algorithms start with the first random variable such as [20,14]. In the beginning, one may be interested investigate what variable is suitable for starting the algorithm instead of the fixed choice of the first variable? The reason may return to the fact that we didn't have any further knowledge about the nodes model, but here, we maximize the whole structure-function or main likelihood aiming to find the best model. For these variables assume that we determine C^*_3 such that:

$$P(X_1 \leq x_1, X_2 \leq x_2, X_3 \leq x_3) = C^*_3(F_{X_1}(x_1), F_{X_2}(x_2), F_{X_3}(x_3)),$$

and also it is clear that:

$$f_{X_1,X_2,X_3}(x_1,x_2,x_3) = \prod_{j=1}^{3} f_{X_j}(x_j) c^*_3(F_{X_1}(x_1), F_{X_2}(x_2), F_{X_3}(x_3)).$$

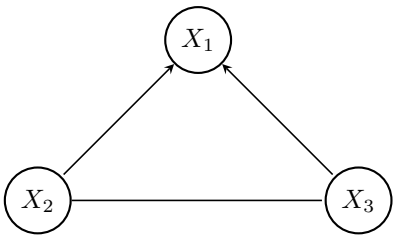

Figure 3. Case I

In accordance with these relations, it is available to calculate marginal densities or partial structure values. First of all, without losing any generality, assume that $P_3 \leq P_2 \leq P_1$. Hence the priorities respectively are P_3, P_2, and P_1 and the learning algorithm should start with nodes X_1. The next question arise now. What is the number of connected variables to X_1? Are two nodes connected to X_1 like as figure (3), or just only a variable is connected to X_1 like as figures (4), and (5)? The other issue is, if there is only one variable connected to X_1, what is that node? or what model should be preferred between figures (4), and (5)?

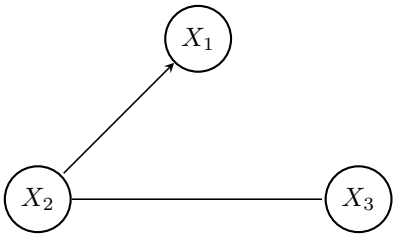

Figure 4. Case II

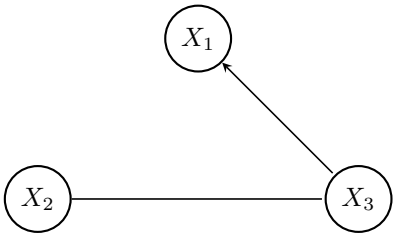

Figure 5. Case III

Discussing the above question, we can extend P_i for two or more directions. For instance, the affection received by node X_i from the other two nodes like as X_{j_1} and X_{j_1}, can be extended as:

$$\begin{aligned} P^i_{j_1,j_2} &= \int_{\mathbb{R}} \frac{f_{X_i}(x)}{f_{X_{j_1},X_{j_2}|X_i}(x)} dx \\ &= \int_{\mathbb{R}} \frac{f^2_{X_i}(x)}{f_{X_{j_1},X_{j_2}}(x,x)} dx \end{aligned}$$

Accordingly, choosing two suitable nodes can be done by maximizing the above relation just like the previous algorithms. Returning our attention to the main questions, we can calculate $P^1_{2,3}$, P^1_3, and P^1_2. In what follows, we should choose among 2 and 1 directions connected to the pre-selected node X_1. Here, we can define the maximum affection received by node X_i from k directions as follows:

$$P^{i-k} = \max\left(P^i_{j_1,j_2,\ldots,j_k} | (j_1, j_2, \ldots, j_k) \in \pi_k\right), \tag{2}$$

where π_k stands for all corresponding possible permutations.

Finally, we have:

$$P^{1-2} = P^1_{2,3},$$

and

$$P^{1-1} = \max\left(P^1_2, P^1_3\right).$$

Now, if $P^{1-2} \geq P^{1-1}$, we prefer Figure (3), while if $P^{1-2} \leq P^{1-1}$, in accordance with the previous steps, we should determine what the figures are suitable between figure (4), and (5).

5. Simulation studies

In this section, five heterogeneous nodes with different types of dependency are considered and the corresponding simulation studies aiming to find related DAG's are present. To this end, the Gumbel (1.5), Clayton (1), and

normal (0.5), dependence structures are considered with the same Kendalls tau $\frac{1}{3}$, representing a positive strong dependency. In addition, five models were also taken into account including normal, Weibull, gamma, exponential, and betta densities that are mostly utilized in real-world applications. In the beginning, we should determine the variable priorities. This code for normal copula is available in Appendix and the calculation for other copulas is similar and also can be transformed. In this regard, the P_i values are available in Table (1), and under any assumption, we can determine the main node. The same task is also provided for P_i^k in Table (2).

	Normal copula (0.5)	Clayton copula (1)	Gumbel copula (1.5)
P_1	7.048	22.593	6.255
P_2	3.768	8.773	2.821
P_3	0.615	1.469	0.466
P_4	8.082	38.159	9.028
P_5	92.913	279.033	80.172
Priorities	$P_3 \leq P_2 \leq P_1 \leq P_4 \leq P_5$	$P_3 \leq P_2 \leq P_1 \leq P_4 \leq P_5$	$P_3 \leq P_2 \leq P_1 \leq P_4 \leq P_5$

Table 1. The P_i values for the Clayton, Gumbel, and normal copulas with assumed variable models

Hence the priorities are the same in the first step and the algorithm should be started from node X_5. In continue, the partial priorities have been calculated in Table (1), and in the same manner, the learning contexts are shown in Figures (6), (7), and (8), respectively for the normal, Clayton, and Gumbel structures.

	Normal copula (0.5)	Clayton copula (1)	Gumbel copula (1.5)
$P^5_{1,2,3,4}$	14.265	24.368	63.254
P^{5-4}	14.265	24.368	63.254
$P^5_{1,2,3}$	13.285	12.654	34.957
$P^5_{1,2,4}$	11.395	21.478	54.298
$P^5_{1,3,4}$	17.426	32.147	24.298
$P^5_{2,3,4}$	21.289	25.962	40.147
P^{5-3}	21.289	32.147	54.298
$P^5_{1,2}$	29.821	30.126	32.258
$P^5_{1,3}$	9.357	19.895	43.159
$P^5_{1,4}$	11.237	13.592	21.328
$P^5_{2,3}$	32.856	35.728	17.983
$P^5_{2,4}$	38.921	15.628	29.864
$P^5_{3,4}$	33.294	21.328	11.958
P^{5-2}	38.921	35.728	43.159
P^5_1	7.284	41.925	11.278
P^5_2	4.329	34.956	29.862
P^5_3	9.289	84.962	21.327
P^5_4	11.249	67.389	27.925
P^{5-1}	11.249	84.962	29.862

Table 2. The corresponding priorities after choosing the node X_5.

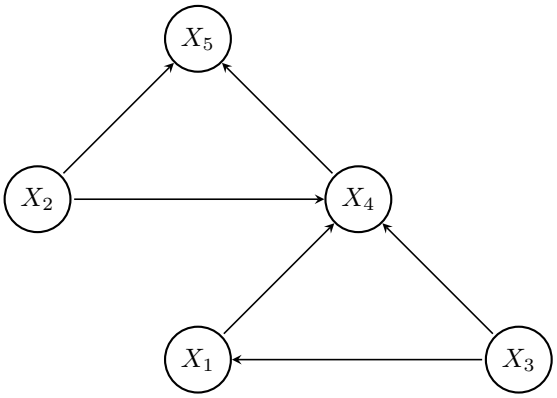

Figure 6. Dag learning for the normal copula

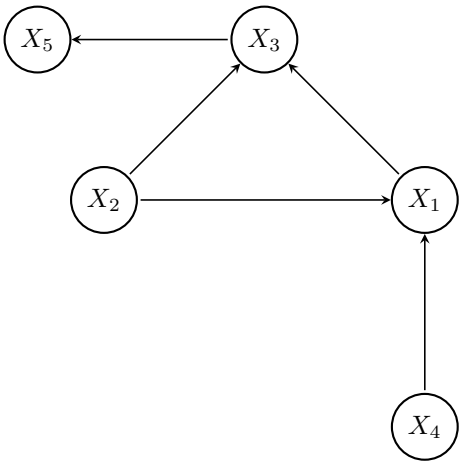

Figure 7. Dag learning for the Clayton copula

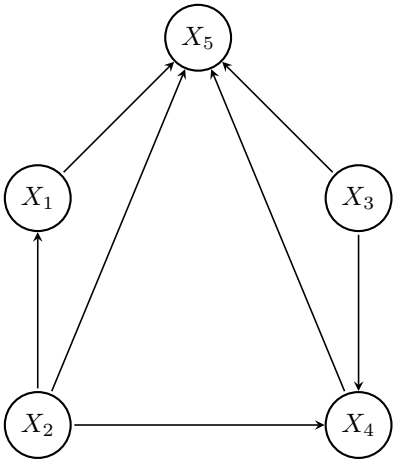

Figure 8. Dag learning for the Gumbel copula

6. Conclusion

In this study, a multivariate data set coming from an unknown directed acyclic graph has been considered and we are going to determine the learning topic around this graph. In contrast to the previous studies, not only normal assumptions for these variables are considered but also non-normal assumptions have been widely investigated.

To this end, the copula function is utilized for constructing a suitable dependency structure between the considered variables. The copula function is not our attention here and it has been used just for fitting the variables for a good model as it is possible that can be normal or even not normal.

The task is done by introducing three non-normal models including FGM, Clayton, and Gumbel, and moreover the normal one itself. The fitting process is devoted to an excellent package "gofCopula" and its very useful command "gof" and we regret the further topics around finding a suitable model, but the corresponding algorithm is provided in detail. Using the selected model, we are going to determine structure-function due to all possible directions.

The direction with maximum value is selected as an excellent DAG model and the corresponding algorithms are also given with comprehensive discussion. In continue, through simulation studies, these algorithms are examed and their performances are also calculated. In addition, some tables and figures are presented as a good showcase of what we discussed.

REFERENCES

[1] ARAGAM, B., AMINI, A. A., AND ZHOU, Q. Learning directed acyclic graphs with penalized neighbourhood regression. *arXiv preprint arXiv:1511.08963* (2015).

[2] ARAGAM, B., AND ZHOU, Q. Concave penalized estimation of sparse gaussian bayesian networks. *The Journal of Machine Learning Research 16*, 1 (2015), 2273–2328.

[3] CAO, X., KHARE, K., AND GHOSH, M. Posterior graph selection and estimation consistency for high-dimensional bayesian dag models. *The Annals of Statistics 47*, 1 (2019), 319–348.

[4] COOPER, G. F., AND HERSKOVITS, E. A bayesian method for constructing bayesian belief networks from databases. In *Uncertainty Proceedings 1991*. Elsevier, 1991, pp. 86–94.

[5] CÓRDOBA, I., BIELZA, C., AND LARRAÑAGA, P. A review of gaussian markov models for conditional independence. *Journal of Statistical Planning and Inference 206* (2020), 127–144.

[6] GEIGER, D., AND HECKERMAN, D. Learning gaussian networks. In *Uncertainty Proceedings 1994*. Elsevier, 1994, pp. 235–243.

[7] GIUDICI, P., AND CASTELO, R. Improving markov chain monte carlo model search for data mining. *Machine learning 50*, 1 (2003), 127–158.

[8] GOUDIE, R., AND MUKHERJEE, S. A gibbs sampler for learning dags.

[9] HECKERMAN, D., AND GEIGER, D. Learning bayesian networks: a unification for discrete and gaussian domains. *arXiv preprint arXiv:1302.4957* (2013).

[10] HECKERMAN, D., GEIGER, D., AND CHICKERING, D. M. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning 20*, 3 (1995), 197–243.

[11] KRATZER, G., AND FURRER, R. mcmcabn: a structural mcmc sampler for dags learned from observed systemic datasets. *R package version*, 3 (2019).

[12] NIINIMAKI, T., PARVIAINEN, P., AND KOIVISTO, M. Partial order mcmc for structure discovery in bayesian networks. *arXiv preprint arXiv:1202.3753* (2012).

[13] RAMSEY, J., GLYMOUR, M., SANCHEZ-ROMERO, R., AND GLYMOUR, C. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International journal of data science and analytics 3*, 2 (2017), 121–129.

[14] REZAEI TABAR, V., ZAREIFARD, H., SALIMI, S., AND PLEWCZYNSKI, D. An empirical bayes approach for learning directed acyclic graph using mcmc algorithm. *Statistical Analysis and Data Mining: The ASA Data Science Journal 12*, 5 (2019), 394–403.

[15] SPIEGELHALTER, D. J., DAWID, A. P., LAURITZEN, S. L., AND COWELL, R. G. Bayesian analysis in expert systems. *Statistical science* (1993), 219–247.

[16] TABAR, V. R., ESKANDARI, F., SALIMI, S., AND ZAREIFARD, H. Finding a set of candidate parents using dependency criterion for the k2 algorithm. *Pattern Recognition Letters 111* (2018), 23–29.

[17] TEAM, R. C., ET AL. R: A language and environment for statistical computing.

[18] WRIGHT, S. *Principles of livestock breeding*. No. 905. US Department of Agriculture, 1920.

[19] WRIGHT, S. The relative importance of heredity and environment in determining the piebald pattern of guinea-pigs. *Proceedings of the National Academy of Sciences of the United States of America 6*, 6 (1920), 320.

[20] ZAREIFARD, H., REZAEI TABAR, V., AND PLEWCZYNSKI, D. A gibbs sampler for learning dag: a unification for discrete and gaussian domains. *Journal of Statistical Computation and Simulation* (2021), 1–21.

[21] ZHENG, X. *Learning DAGs with Continuous Optimization*. PhD thesis, PhD thesis, Carnegie Mellon University, 2020.