Constraint-based learning for non-parametric continuous bayesian networks

Marvin Lasserre, Régis Lebrun, Pierre-Henri Wuillemin

Annals of Mathematics and Artificial Intelligence manuscript No.
(will be inserted by the editor)

Constraint-Based Learning for Non-Parametric Continuous Bayesian Networks

Marvin Lasserre · Régis Lebrun · Pierre-Henri Wuillemin

Received: date / Accepted: date

Abstract Modeling high-dimensional multivariate distributions is a computationally challenging task. In the discrete case, Bayesian networks have been successfully used to reduce the complexity and to simplify the problem. However, they lack of a general model for continuous variables. In order to overcome this problem, [9] proposed the model of copula Bayesian networks that parametrizes Bayesian networks using copula functions. We propose a new learning algorithm for this model based on a PC algorithm and a conditional independence test proposed by [4]. This test being non-parametric, no model assumptions are made allowing it to be as general as possible. This algorithm is compared on generated data with the parametric method proposed by [9] and proves to have better results.

Keywords Continuous Bayesian Networks, Non-parametric Learning, Copula Theory

1 Introduction

Modeling multivariate continuous distributions is an important task in statistics and machine learning with many applications in science and engineering. However, high-dimensional distributions are hard to manipulate and may lead to intractable computations. In addition, apart from simple parametric models such as the Gaussian distribution, univariate distributions usually don't have multivariate equivalents leading to difficulties in building multivariate models.

M. Lasserre
4 place Jussieu, 75005 Paris, France
E-mail: marvin.lasserre@lip6.fr

R. Lebrun
Airbus AI Research
22 rue du Gouverneur Général Eboué
92130 Issy les Moulineaux, France
E-mail: regis.lebrun@airbus.com

P.-H. Wuillemin
4 place Jussieu, 75005 Paris, France
E-mail: pierre-henri.wuillemin@lip6.fr

Probabilistic graphical models are used to compactly represent such multivariate distributions. In particular, Bayesian networks (BN) use a directed acyclic graph (DAG) and a set of conditional probability distributions (CPD) to encode the joint distribution. This representation reduces the complexity by taking advantage of conditional independencies and allows efficient inference and learning algorithms. However, BNs lack of a general model for continuous variables: discretizations or Gaussian models are often used despite no theoretical restrictions on CPD models. On the one hand, discretizations need to be determined and are limited in the number of bins that can be used. Gaussian models on the other hand allow efficient inference and learning algorithms but lack of expressiveness.

According to Sklar (theorem 1), any multivariate distribution is related to its univariate marginals by means of a copula function. Thus, the copula function allows to model the dependence structure between continuous variables by ruling out the marginal behavior of each variable. From a constructive perspective, this allows to dissociate the choice of the marginals and the choice of a dependence structure. In practice however, copulas are limited to only a few variables and constructing or manipulating high-dimensional ones is difficult.

In order to take advantage of these two frameworks, many graphical models for copulas have been proposed such as pair-copula construction [7], Vine model [2] or cumulative distribution networks [13]. One encouraging model is the Copula Bayesian Network (CBN) [9] which parametrizes a BN with a set of local conditional copula functions giving it the same graphical properties. Consequently, this allows to use similar methods than in the classical case to learn them. In this regard, [9] proposed a learning method based on the well known BIC score coupled with a TABU search.

The main contribution of this paper is a new learning algorithm for CBNs. This learning algorithm relies on a PC-algorithm coupled with a continuous conditional independence (CI) test proposed by [4] that uses Bernstein copula estimators. The method is compared with the BIC score method in terms of structural scores and time complexity on generated data sets.

The paper is organized as follows. Section 2 describes copulas and some of their useful properties. Section 3 introduces the CBN framework proposed by [9]. Section 4 presents in details the two learning algorithms for CBNs, that is our algorithm and the method proposed in [9]. Section 5 compares the algorithms on generated data from known structures in terms of structure learning and time complexity. Section 6 concludes the paper.

2 Copulas

Let $\overline{\mathbb{R}}$ be the extended set of real numbers defined as $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ and $\mathbf{I}$ be the unit segment $[0, 1]$. Let $\mathbf{X} = (X_1, \ldots, X_d)$ be an n-dimensional random vector and $\mathbf{x} = (x_1, \ldots, x_d)$ a vector of $\overline{\mathbb{R}}^d$ denoting a realization of $\mathbf{X}$. Before giving the definition of a multivariate (cumulative) distribution function, the notion of H-volume needs to be introduced.

Definition 1 (H-Volume) Let $\mathbf{a}$ and $\mathbf{b}$ be two points of $\overline{\mathbb{R}}^d$ such that for each i, $a_i < b_i$. The d-box $\mathbf{B}$ is the set $\times_{i=1}^{d}[a_i, b_i]$. Giving a function H defined on a

subset of $\overline{\mathbb{R}}^d$ containing $\mathbf{B}$, the H-volume of $\mathbf{B}$ is the quantity:

$$V_H(\mathbf{B}) = \sum_{\mathbf{v}\in\mathcal{V}} (-1)^{N(\mathbf{v})} H(\mathbf{v})$$

where $\mathcal{V} = \times_{i=1}^{d}\{a_i, b_i\}$ is the set of the vertices of the d-box and where $N(\mathbf{v}) = \text{card}\,\{i|v_i = a_i\}$.

As an example, let $H : (u, v) \to uv$ be a function defined on $\mathbb{R}^2$ and let $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = (b_1, b_2)$ be two points of $\mathbb{R}^2$. These points define the 2-box $[a_1, b_1] \times [a_2, b_2]$ whose H-volume is given by

$$\begin{aligned} V_H([a_1, b_1] \times [a_2, b_2]) &= H(a_1, a_2) + H(b_1, b_2) - H(a_1, b_2) - H(a_2, b_1) \\ &= a_1a_2 + b_1b_2 - a_1b_2 - a_2b_1 \\ &= (a_1 - b_1)(a_2 - b_2), \end{aligned}$$

which is the area of the 2-box. In fact, for the particular case of $H : \boldsymbol{u} \to \prod_{i=1}^{d} u_i$, the H-volume of a d-box is the usual notion of volume in an Euclidean space.

Definition 2 (Distribution Function) The distribution function $H : \overline{\mathbb{R}}^d \to \mathbf{I}$ of a random vector $\mathbf{X}$ is given by

$$H(x_1, \ldots, x_d) := \mathbb{P}\left(X_1 \leqslant x_1, \ldots, X_d \leqslant x_d\right),$$

Equivalently, a function H is a distribution function if it satisfies the following properties:

1. H is right-continuous in each of its variables,
2. For each d-box $\mathbf{B} \subseteq \overline{\mathbb{R}}^d$, $V_H(\mathbf{B}) \geqslant 0$,
3. $H(x_1, \ldots, x_d) = 0$ if there exists i such that $x_i = -\infty$,
4. $H(+\infty, \ldots, +\infty) = 1$.

In the case $d = 1$, the H-volume is given by $V_H([a, b]) = H(b) - H(a)$ and then the property $V_H([a, b]) \geqslant 0$ means that the function H is increasing. For $d > 1$, this may be considered as an extension of the definition of an increasing function for multivariate functions.

The distribution function of a subset of component variables may be obtained from marginal distributions.

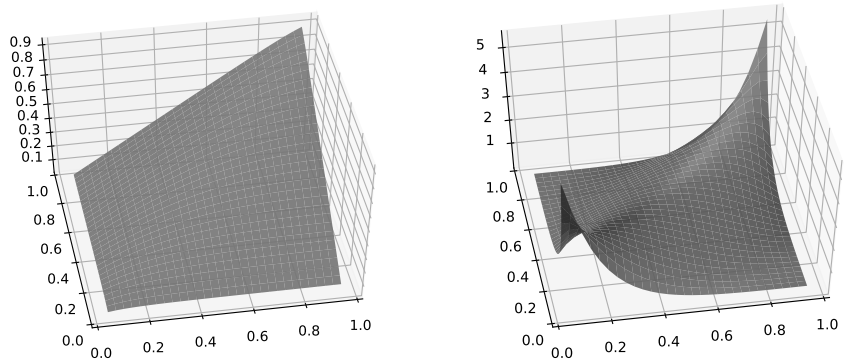
Definition 3 (Marginal distribution) Let H be a d-dimensional distribution function and $\boldsymbol{j} = (j_1, \ldots, j_k)$ a sub-vector of $(1, \ldots, d)$ with $1 \leqslant k \leqslant d - 1$. The $\boldsymbol{j}$-marginal of H is the distribution function $H_{\boldsymbol{j}} : \mathbb{R}^k \to \boldsymbol{I}$ defined by:

$$H_{\boldsymbol{j}}(x_1, \ldots, x_k) = H(y_1, \ldots, y_d),$$

where $y_i = x_i$ if $i \in \{j_1, \ldots, j_k\}$, and $y_i = +\infty$ otherwise.

In particular, the 1-dimensional marginal distribution[1] F_i, for each component X_i, is obtained by the formula $F_i(x_i) = H(+\infty, \ldots, x_i, \ldots, +\infty)$.

[1] When it is clear from context, the index i will be dropped in order to alleviate notations.



(a) Distribution function (b) Density function

Fig. 1: Vizualisation of a two-dimensional gaussian copula with a correlation parameter $\rho = 0.8$.

When variables are independent, the joint distribution can be expressed in terms of its univariate marginals:

$$H(x_1, \ldots, x_d) = \prod_{i=1}^{d} F_i(x_i). \tag{1}$$

Thus, giving any set of arbitrary marginal distributions F_i, a joint distribution can be constructed by taking their product. Copula functions allow to achieve the same objective but with dependent variables.

Definition 4 (Copula) Let $\mathbf{U} = (U_1, \ldots, U_d)$ be a random vector whose components are uniformly distributed on $\mathbf{I}$. A copula function $C : \mathbf{I}^d \to \mathbf{I}$ is a distribution:

$$C(u_1, \ldots, u_d) = \mathbb{P}(U_1 \leqslant u_1, \ldots, U_d \leqslant u_d)$$

The relation between the joint distribution and its univariate marginals is a central result of copula theory due to [25]:

Theorem 1 (Sklar 1959) *Let H be any multivariate distribution function over a random vector $\mathbf{X}$, there exists a copula function C such that*

$$H\left(x_1, \ldots, x_d\right) = C\left(F_1(x_1), \ldots, F_d(x_d)\right). \tag{2}$$

Furthermore, if each $F_i(x_i)$ is continuous then C is unique.

As the marginals encode the individual behavior of each variables, the copula function C encodes the dependence between these variables. This is interesting from a constructive perspective since the choice of marginals can be separated from the choice of the dependence structure.

The independent copula, denoted Π, can be derived from (1) and the previous theorem and has for expression $\Pi\left(\mathbf{u}\right) = \prod_{i=1}^{d} u_i$. More generally, Sklar's theorem may be used to construct new copulas from known multivariate distributions by inverting (2) :

$$C(u_1, \ldots, u_d) = H(F_1^{-1}(u_1), \ldots, F_d^{-1}(u_d)) \tag{3}$$

where $u_i = F(x_i)$. However, if a marginal distribution is not strictly increasing, it may be non-invertible and the quasi-inverse is used instead.

Definition 5 (Quasi-Inverse) The quasi-inverse (or quantile function) of a 1-dimensional distribution function $F : \overline{\mathbb{R}} \to \mathbf{I}$ is the function F^{-1} on $\mathbf{I}$ defined by

$$F^{-1}(y) = \inf\{x|F(x) \geqslant y\} = \sup\{x|F(x) \leqslant y\}.$$

If F is continue and strictly increasing, F^{-1} is the classic inverse function.

Then taking $H = \Phi_R$, the multivariate standard gaussian distribution with correlation matrix R, the gaussian copula [21] can be extracted using (3) :

$$C_G(u_1, \ldots, u_d) = \Phi_R(\phi^{-1}(u_1), \ldots, \phi^{-1}(u_d)) \tag{4}$$

where ϕ is the univariate standard gaussian distribution. A representation of a two-dimensional gaussian copula is given on Figure 1a.

Copula functions are invariant under increasing transformations of random variables. Indeed, let $\{\psi_i\}$ be a family of such transformations and let $U_i = \psi_i(X_i)$, then

$$H'(u_1, \ldots, u_d) = C'(F'_1(u_1), \ldots, F'_d(u_d)).$$

By definition of marginal distributions,

$$\begin{aligned} F'_i(u_i) &= \mathbb{P}(U_i \leqslant u_i) = \mathbb{P}(\psi_i(X_i) \leqslant u_i) \\ &= \mathbb{P}(X_i \leqslant \psi_i^{-1}(u_i)) = F(\psi_i^{-1}(u_i)) \end{aligned}$$

and injecting it in the previous equation, it gives that

$$\begin{aligned} H'(u_1, \ldots, u_d) &= \mathbb{P}(U_1 \leqslant u_1, \ldots, U_d \leqslant u_d) \\ &= \mathbb{P}(X_1 \leqslant \psi_1^{-1}(u_1), \ldots, X_d \leqslant \psi_d^{-1}(u_d)) \\ &= H(\psi_1^{-1}(u_1), \ldots, \psi_d^{-1}(u_d)) \\ &= C(F_1(\psi_1^{-1}(u_1)), \ldots, F_d(\psi_d^{-1}(u_d))) \\ &= C(F'_1(u_1), \ldots, F'_d(u_d)), \end{aligned}$$

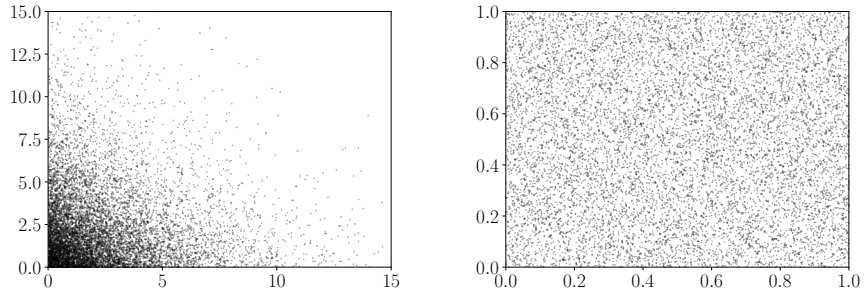
proving that $C' = C$. A multivariate gaussian distribution $\Phi(x_1, \ldots, x_d)$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ can be reparameterized as a standard one using the transformations $\psi_i(x_i) = \frac{x_i - \mu_i}{\sigma_i^2}$. These transformations being increasing, they don't affect the copula according to the last property and this explains why the simplest parameterization can be used when defining the gaussian copula. Moreover, using this last property with $\psi_i = F_i$, we have that $H'(u_1, \ldots, u_d) = C(u_1, \ldots, u_d)$ which allows to work directly with the copula function and to look at the dependence structure. However, in many applications the F_i's are usually unknown and empirical distributions are used instead :

Definition 6 (Rank variables) Let $\mathbf{X}$ be a set of random variables and $\mathcal{D} = \{\boldsymbol{x}[m]\}_{1 \leqslant m \leqslant n}$ a sample containing n realizations of $\mathbf{X}$. The rank variables $\mathbf{U}$ are defined by:

$$\mathbf{U} = (F_1^{emp}(X_1), \ldots, F_d^{emp}(X_d)),$$

where $F_j^{emp}(u) = \dfrac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{X_j^i \leqslant u}$ for $u \in [0, 1]$ is the empirical distribution of X_j. The associated sample, called rank sample, is given by $\mathcal{R} = \{\mathbf{u}[m]\}_{1 \leqslant m \leqslant n}$ where

$$\mathbf{u}[m] = \left(F_1^{emp}(\mathbf{x}_1[m]), \ldots, F_d^{emp}(\mathbf{x}_d[m])\right).$$



(a) Sample (b) Copula

Fig. 2: A sample distributed according to $X \sim \exp(0.5)$, $Y \sim \exp(0.5)$ and the associated rank variables. While the two random variables are independent, looking at the sample it seems that they are dependent. However, the copula exhibits clearly the independence and what can be mistaken to be a dependence is in fact due to a marginal behavior.

Figure 2 shows a sample of a 2-dimensional random vector and the associated rank variables.

If a distribution function is continuous, its joint density is obtained by deriving it : $h(\mathbf{x}) = \frac{\partial^d H(x_1,\ldots,x_d)}{\partial x_1 \ldots \partial x_d}$. A copula density function[2] can be equivalently defined by derivation $c(u_1,\ldots,u_d) = \frac{\partial^d C(u_1,\ldots,u_d)}{\partial u_1 \ldots \partial x_d}$. As the univariate marginals of a copula function are distributed uniformly over $\mathbf{I}$, that is $C(u_i) = u_i$, then the univariate copula densities are identically equal to 1 over $\mathbf{I}$. Now using Sklar's theorem, the joint density can be related to the copula density:

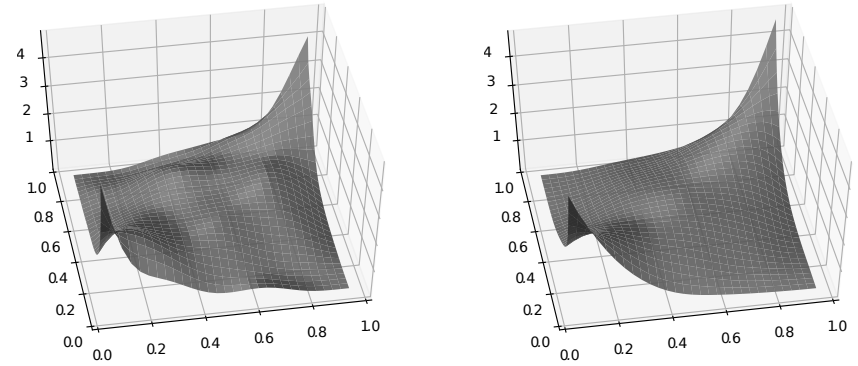
$$\begin{aligned} h(x_1,\ldots,x_d) &= \frac{\partial^d H(x_1,\ldots,x_d)}{\partial x_1 \ldots \partial x_d} \\ &= \frac{\partial^d C(F_1(x_1),\ldots,F_d(x_d))}{\partial F_1(x_1) \ldots \partial F(x_d)} \prod_{i=1}^{d} \frac{\partial F_i(x_i)}{\partial x_i} \\ &= c\left(F_1(x_1),\ldots,F_d(x_d)\right) \prod_{i=1}^{d} f_i(x_i). \end{aligned} \tag{5}$$

This formula will be used extensively in the next section to define CBNs. Similarly to the Sklar's theorem, the last relation can be inverted to obtain a copula density:

$$c\left(u_1,\ldots,u_d\right) = \frac{h\left(F^{-1}(u_1),\ldots,F^{-1}(u_d)\right)}{\prod_{i=1}^{d} f_i\left(F^{-1}(u_i)\right)}. \tag{6}$$

The gaussian copula being continuous, a density can be obtained from derivation of equation (4). A representation of a two-dimensional gaussian copula density is given on Figure 1b. While the gaussian copula has been used in order to illustrate the different notions about copula theories there exists many other parametric copulas and the interested reader can refer to [15], [21] for an exhaustive list.

[2] By abuse of terminology, the copula density is often named copula.



(a) $M = 100$ (b) $M = 1000$

Fig. 3: Approximation of gaussian copula by an empirical Bernstein copula with different size of samples.

This section ends with the introduction of the empirical Bernstein copula [23], a non-parametric copula estimator, which is at the center of the learning method introduce later. Its definition relies on the empirical copula [8]:

Definition 7 (Empirical copula) Given a sample $\mathcal{D}$ of size n and $\mathcal{R}$ the associated rank sample, the empirical copula is defined as

$$\hat{C}_n(\boldsymbol{u}) = \frac{1}{n} \sum_{m=1}^{n} \prod_{i=1}^{d} \mathbb{1}\{u_i[m] \leqslant u_i\}.$$

The Bernstein copula is a smoothed version of the empirical copula using Bernstein polynomials:

Definition 8 (Empirical Bernstein copula) The empirical Bernstein copula on a sample $\mathcal{D}$ of size n and associated rank sample $\mathcal{R}$ is defined as:

$$\hat{C}^B_{K,n}(\boldsymbol{u}) = \sum_{v_1=0}^{K} \cdots \sum_{v_d=0}^{K} \hat{C}_n\left(\frac{v_1}{K}, \ldots, \frac{v_d}{K}\right) \prod_{i=1}^{d} B_{v_i,K}(u_i) \tag{7}$$

where K is a bandwidth parameter and $B_{i,n}(x) = \binom{n}{i} x^i (1-x)^{n-i}$ are the Bernstein polynomials.

Despite his name, the empirical Bernstein copula is truly a copula if and only if n is a multiple of K. For this reason, certain instances of the data set may be ignored. As for any copula, a density function can be defined for the Bernstein copula by deriving equation (7):

Theorem 2 (Bernstein copula density) *The Bernstein copula density has for expression*

$$\hat{c}_B(\boldsymbol{u}) = \frac{1}{n} \sum_{m=1}^{n} \left(K^d \sum_{v_1=0}^{K} \cdots \sum_{v_d=0}^{K} \mathbb{1}(\boldsymbol{u}[m] \in S_{\boldsymbol{v}}) \prod_{i=1}^{d} B_{v_i,K-1}(u_i) \right), \tag{8}$$

where

$$S_{\boldsymbol{v}} = \left[\frac{v_1}{K}, \ldots, \frac{v_1+1}{K}\right] \times \cdots \times \left[\frac{v_d}{K}, \ldots, \frac{v_d+1}{K}\right] \quad \textit{and } \boldsymbol{v} = (v_1, \ldots, v_d).$$

Finally, K is chosen such that it minimizes the mean square error (MSE) of the density estimator, that is such that $\|\hat{c}_B - c\|^2$ with $\|.\|$ the L_2 norm and c the true copula density. This is achieved by taking $K_{\text{MSE}} = \lfloor 1 + n^{2/(d+4)} \rfloor$ [23]. The figure 3 shows the approximation of a Gaussian copula by an empirical Bernstein copula.

3 Copula Bayesian Networks

A BN structure $\mathcal{G}$ is a DAG whose vertices $\mathbf{X} = \{X_1, \ldots, X_d\}$ represent random variables. Let $\mathbf{Pa}_i = (\text{Pa}_{i1}, \ldots, \text{Pa}_{ik_i})$ be the k_i parents of X_i in $\mathcal{G}$ and $\mathbf{ND}_i$ be the variables that are non-descendants of X_i in the graph. A multivariate probability distribution P over variables $\mathbf{X}$, is said to factorize according to $\mathcal{G}$, if it can be expressed as the product

$$P(X_1, \ldots, X_d) = \prod_{i=1}^{d} P(X_i|\mathbf{Pa}_i).$$

and $\mathcal{G}$ then encodes the set of independencies :

$$\mathcal{I}(\mathcal{G}) = \{(X_i \perp \mathbf{ND}_i|\mathbf{Pa}_i)\}.$$

A BN is a pair $\mathcal{B} = (\mathcal{G}, P)$ where $\mathcal{G}$ is defined as previously and P factorizes over $\mathcal{G}$. To each node X_i of the BN structure is associated its corresponding CPD $P(X_i|\mathbf{Pa}_i)$ that appears in the factorization of the joint distribution P. In the discrete case, CPDs are most often represented via conditional probability tables (CPT) while in the continuous case, they are represented via linear gaussian model [19] $f(x_i|\mathbf{pa}_i) = \mathcal{N}(\beta_{i0} + \beta_i^T \mathbf{pa}_i; \sigma_i^2)$. Although gaussian distributions allow fast probabilistic computations and estimation, they lack of expressiveness and some distributions, like rare events ones, cannot be well approximated by such models. The CBN model introduced by [9] tries to address this problem by using copula functions to parametrize the BN.

In order to do so, the first step is to use (5) in the Bayes formula for $f(x_i|\mathbf{pa}_i)$:

$$\begin{aligned} f(x_i|\mathbf{pa}_i) &= \frac{f(x_i, \mathbf{pa}_i)}{f(\mathbf{pa}_i)} \\ &= \frac{c(F(x_i), F(\text{pa}_{i1}), \ldots, F(\text{pa}_{ik_i}))f(x_i)\prod_{j=1}^{k_i} f(\text{pa}_{ij})}{c(F(\text{pa}_{i1}), \ldots, F(\text{pa}_{ik_i}))\prod_{j=1}^{k_i} f(\text{pa}_{ij})} \\ &= R_{c_i}(F(x_i), F(\text{pa}_{i1}), \ldots, F(\text{pa}_{ik_i}))f(x_i) \end{aligned}$$

where

$$R_{c_i}(F(x_i), F(\text{pa}_{i1}), \ldots, F(\text{pa}_{ik_i})) = \frac{c(F(x_i), F(\text{pa}_{i1}), \ldots, F(\text{pa}_{ik_i}))}{c(F(\text{pa}_{i1}), \ldots, F(\text{pa}_{ik_i})}$$

and $k_i = |\mathbf{pa}_i|$. Consequently, if $f(\mathbf{x})$ that is supposed to be strictly positive, factorizes on $\mathcal{G}$ as $f(\mathbf{x}) = \prod_{i=1}^{d} f(x_i|\mathbf{pa}_i)$, it is the same for the copula density :

$$\begin{aligned} c(F(x_1), \ldots, F(x_d)) &= \frac{f(\mathbf{x})}{\prod_{i=1}^{d} f(x_i)} = \frac{\prod_{i=1}^{d} f(x_i|\mathbf{pa}_i)}{\prod_{i=1}^{d} f(x_i)} \\ &= \frac{\prod_{i=1}^{d} R_{c_i}(F(x_i), F(\mathrm{pa}_1), \ldots, F(\mathrm{pa}_{K_i}))f(x_i)}{\prod_{i=1}^{d} f(x_i)} \\ &= \prod_{i=1}^{d} R_{c_i}(F(x_i), F(\mathrm{pa}_1), \ldots, F(\mathrm{pa}_{K_i})). \end{aligned}$$

Like with BNs, the converse is also true :

Theorem 3 (Elidan 2010) *Let $\mathcal{G}$ be a DAG over $\mathbf{X}$. In addition, let $\{c_i(F(x_i), F(pa_{i1}), \ldots, F(pa_{ik_i}))\}$ be a set of strictly positive copula densities associated with the nodes of $\mathcal{G}$ that have at least one parent. If $\mathcal{I}(\mathcal{G})$ holds then the function*

$$h(F(x_1), \ldots, F(x_d)) = \prod_{i=1}^{d} R_{c_i}(F(x_i), \{F(pa_{ik})\})f(x_i),$$

is a valid density over $\mathbf{X}$.

This leads to the definition of a CBN as given by [9] :

Definition 9 (Copula Bayesian Network) A Copula Bayesian Network is a triplet $\mathcal{C} = (\mathcal{G}, \Theta_C, \Theta_f)$ that encodes the joint density $f(\mathbf{x})$. Θ_C is a set of local copula densities functions $c_i\,(F(x_i), \{F(pa_{ik})\})$ that are associated with the nodes of $\mathcal{G}$ that have at least one parent. Θ_f is the set of parameters representing the marginal densities $f(x_i)$. $f(\mathbf{x})$ is parametrized as

$$f(\mathbf{x}) = \prod_{i=1}^{d} R_{c_i}(F(x_i), \{F(pa_{ik})\})f(x_i). \tag{9}$$

We finish this section by giving a simple example of CBN illustrated by figure 4. This CBN encodes a joint density function over variables X_1, X_2 and X_3. Each node represents a random variable to which is associated its marginal density function f_i and a local copula density function c_i. The set of local copulas and marginals are respectively given by $\Theta_C = \{c_1, c_2, c_3\}$ and $\Theta_f = \{f_1, f_2, f_3\}$. The structure of the CBN encodes the factorization of the joint density f:

$$\begin{aligned} f(x_1, x_2, x_3) &= R_{c_1}(F(x_1))R_{c_2}(F(x_2), F(x_1))R_{c_3}(F(x_3, x_2))f_1(x_1)f_2(x_2)f_3(x_3) \\ &= [f(x_1)]\,[c_2(F(x_2), F(x_1))f(x_2)]\,[c_3(F(x_3), F(x_2))f(x_3)]\,. \end{aligned}$$

The simplification here is due to the fact that the univariate copula densities are identically equal to 1 over $\mathbf{I}$. The parametrization of marginals and copulas is not specified and could be any model. For example, the local copulas could be Gaussian and the marginal densities from an exponential distribution.

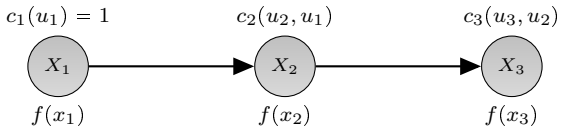


Fig. 4: A CBN with three variables X_1, X_2 and X_3.

4 Learning

CBNs have the same local properties as the (classical) BNs allowing to use similar algorithms in order to learn the structure of a CBN. Those algorithms can be roughly divided into two classes: score based methods and constrained based methods. For score based method, the learning task is viewed as a model selection and a scoring function is used to measure how good the model fit the dataset. The space of all DAG structures being superexponential, this score is often maximized using local search methods such as hill-climbing, gradient ascent, simulated annealing, TABU list, etc. Constrained-based methods on the other hand look at the graph as a set of (conditional) independences and use CI tests, such as χ^2 in the discrete case, to obtain information about the underlying structure. We present one method of each kind in this section and compare them in the next section.

4.1 Score based method (CBIC)

In [9], a score-based method is used to learn the structure of a CBN. The proposed score is the well-known Bayesian information criterion (BIC) [24]. Its expression on a CBN structure $\mathcal{G}$ is given by :

$$\mathcal{S}_{BIC}(\mathcal{G} : \mathcal{D}) = \ell(\mathcal{D} : \hat{\theta}, \mathcal{G}) - \frac{1}{2}\log(n)|\Theta_{\mathcal{G}}|,$$

where ℓ is the log-likelihood, $\hat{\theta}$ are the maximum likelihood parameters estimators (MLE) and $|\Theta_{\mathcal{G}}|$ is the number of free parameters associated with the graph structure. Using the factorization of the joint density (9), we have :

$$\begin{aligned}
\ell(\mathcal{D} : \mathcal{G}) &= \sum_{m=1}^{n} \log f(x_1[m], \ldots, x_d[m]) \\
&= \sum_{m=1}^{n} \log \prod_{i=1}^{d} R_i(F(x_i[m]), \{F(pa_{ik_i}[m])\}) f(x_i[m]) \\
&= \sum_{m=1}^{n} \sum_{i=1}^{d} \log R_i(F(x_i[m]), \{F(pa_{ik_i}[m])\}) + \sum_{m=1}^{n} \sum_{i=1}^{d} \log f(x_i[m])
\end{aligned}$$

and switching to the rank sample, this reduces to

$$\ell(\mathcal{D} : \mathcal{G}) = \sum_{m=1}^{n} \sum_{i=1}^{d} \log R_i\left(u_i[m], \pi_{i1}[m], \ldots, \pi_{ik_i}[m]\right))$$

where $u_i = F(x_i)$ and $\pi_{ij} = F(\text{pa}_{ij})$. [9] uses several copula models in order to define the R_{c_i}'s but we only retain the most expressive one which is the Gaussian copula model parametrized by a full correlation matrix Σ. Finding directly the MLE for Σ may be difficult in high dimension and this is why a proxy is used. It relies on the relation $\Sigma_{ij} = \sin(\frac{\pi}{2}\tau_{ij})$ between Kendall's tau τ_{ij} and correlation matrix Σ_{ij} that holds for every elliptical distribution [20]. The τ_{ij} are given by [10]

$$\tau(X_i, X_j) = \frac{2}{n(n-1)} \sum_{m_1=1}^{n-1} \sum_{m_2>m_1}^{n} \text{sign}\Big((X_i[m_1] - X_i[m_2])(X_j[m_1] - X_j[m_2])\Big).$$

However, the matrix obtained by this process is not necessarily a correlation matrix, that is a positive semi-definite (PSD) matrix, and regularization techniques may be needed to obtain one [22]. Finally, the BIC score is maximized using a TABU list algorithm with random restarts [11].

4.2 Continuous PC algorithm (CPC)

The PC algorithm introduced by [26] and on which relies our method can be divided in three main steps : skeleton learning, v-structures search and constraint propagation. Starting from the complete non-oriented graph on $\mathbf{X}$, the skeleton search consists in removing edges using CI tests between pairs of variables (X, Y) conditioned on subset $\boldsymbol{Z}$ of common neighbors. If the test finds an independence $X \perp\!\!\!\perp Y \mid \mathbf{Z}$, the edge between the two corresponding variables is removed and the conditioning set is recorded as the separating set between X and Y noted $\mathbf{Sepset}(X, Y)$. Once this first step is completed, the triplets $X - Z - Y$ such that X and Y are not neighbors and Z is not in $\mathbf{Sepset}(X, Y)$, are oriented as $X \rightarrow Z \leftarrow Y$ which is called a v-structure. Finally, the remaining non-oriented edges are oriented under the constraint that no new v-structures are added into the graph unless it implies adding an oriented cycle. The PC algorihtm is reported on Algorithm (1), for further details, see page 84 of [26].

The CI test, which is based on Hellinger's distance, is taken from [4, 5] and [27]. Taking two random variables X, Y and $\mathbf{Z} = \{Z_1, \ldots, Z_d\}$ a set of random variables; and with $C_{X,Y,\mathbf{Z}}$ a copula and $c_{X,Y,\mathbf{Z}}$ its density, the article proposes to test:

$$X \perp\!\!\!\perp Y \,|\, \mathbf{Z} \iff \mathbb{P}\left(c_{X,Y,\mathbf{Z}} = \frac{c_{X,\mathbf{Z}} \cdot c_{Y,\mathbf{Z}}}{c_{\mathbf{Z}}}\right) = 1 \tag{10}$$

The Hellinger's distance is then used as a measure of the conditional independence [4] [3].:

$$H(X, Y|\mathbf{Z}) = \int_{[0,1]^{d+2}} \left(1 - \sqrt{\frac{c_{X,\mathbf{Z}}(x, \mathbf{z}) \cdot c_{Y,\mathbf{Z}}(y, \mathbf{z})}{c_{X,Y,\mathbf{Z}}(x, y, \mathbf{z}) \cdot c_{\mathbf{Z}}(\mathbf{z})}}\right)^2 c_{X,Y,\mathbf{Z}}(x, y, \mathbf{z}) dx dy d\mathbf{z}. \tag{11}$$

[3] Equations 10, 11 and 12 were only valid for $|\mathbf{Z}| = 1$ in [4,28] due to the omission of $c_{\mathbf{Z}}$ in the denominators. The derivation of the test statistics still follows the step described in [4]

Algorithm 1: PC algorithm [26]

Input: Data set $\mathcal{D}$
Result: Structure $\mathcal{C}$.
// Initialisation
$\mathcal{C} \leftarrow$ complete undirected graph on $\mathbf{X}$
$n \leftarrow 0$
// Skeleton search
while $|\mathbf{Adjacencies}(X)\backslash Y| < n$ ***or*** $n = n_{max}$ **do**
  foreach $X \in \mathbf{X}$ **do**
    foreach $Y \in \mathbf{Adjacencies}(X)$ **do**
      foreach $\mathbf{Z} \in \mathbf{Adjacencies}(X)\backslash Y$ ***and*** $|Z| = n$ **do**
        if $X \perp\!\!\!\perp Y \,|\mathbf{Z}$ **then**
          Delete edge $X - Y$ from G
          Add $\mathbf{Z}$ to $\mathbf{Sepset}(\mathbf{X}, \mathbf{Y})$
        end
      end
    end
  end
end
// V-structure search
foreach *triple of nodes* (X, Y, Z) *such that* $X - Z$ ***and*** $Z - Y$ ***and not*** $X - Y$ **do**
  if $Z \notin \mathbf{Sepset}(\mathbf{X}, \mathbf{Y})$ **then**
    Orient $X - Z - Y$ as $X \rightarrow Z \leftarrow Y$
  end
end
// Constraint propagation
foreach *edges in* G *that can be oriented* **do**
  Rule 1 : if $X \rightarrow Y$, $Y - Z$, X and Z are not adjacent then orient $Y - Z$ as $Y \rightarrow Z$.
  Rule 2 : if $X - Z$ and $X \rightarrow Y \rightarrow Z$ then orient $X - Z$ as $X \rightarrow Z$.
  Rule 3 : if $X - Z$, $X - Y \rightarrow Z$ and $X - W \rightarrow Z$ such that Y and W are not adjacent, then orien $X - Z$ as $X \rightarrow Z$.
end

The copula $C_{X,Y,\mathbf{Z}}$ is estimated using the empirical Bernstein copula $\widehat{C}_{X,Y,\mathbf{Z}}$ that has been introduced in section 2. The Hellinger distance is then estimated by [4]:

$$\widehat{H} = \frac{1}{n} \sum_{m=1}^{n} \left(1 - \sqrt{\frac{\widehat{c}_{X,\mathbf{Z}}(x[m], \mathbf{z}[m]) \cdot \widehat{c}_{Y,\mathbf{Z}}(y[m], \mathbf{z}[m])}{\widehat{c}_{X,Y,\mathbf{Z}}(x[m], y[m], \mathbf{z}[m]) \cdot \widehat{c}_{\mathbf{Z}}(\mathbf{z}[m])}} \right)^2 \tag{12}$$

where $(x[m], y[m], \mathbf{z}[m])$ is a realization of the variables $(X, Y, \mathbf{Z})$ in the database of M instances from the true copula C. Based on this estimation of the distance, [4] proposes a BRT statistic for CI test[4] for any dimension of $\mathbf{Z}$. Under the assumption $H_0 : X \perp\!\!\!\perp Y \,|\mathbf{Z}$, it can be proven that BRT $\sim \mathcal{N}(0, 1)$.

Our contribution is a PC algorithm using a continuous CI test relying on the BRT to learn CBNs. This method follows the same idea from the work of [28] which proposes a learning procedure to factorize a joint distribution and then learn a mixture of Gaussians for the CPDs. However, in the case of [28], the structure learning and parameter learning models being different, this can lead to non-consistent results. In our case, copulas are at the core of the model since they are used to parametrize the CBN and using a copula based CI test makes perfect sense.

[4] For the expression of BRT, we refer to theorem 1 of [4].

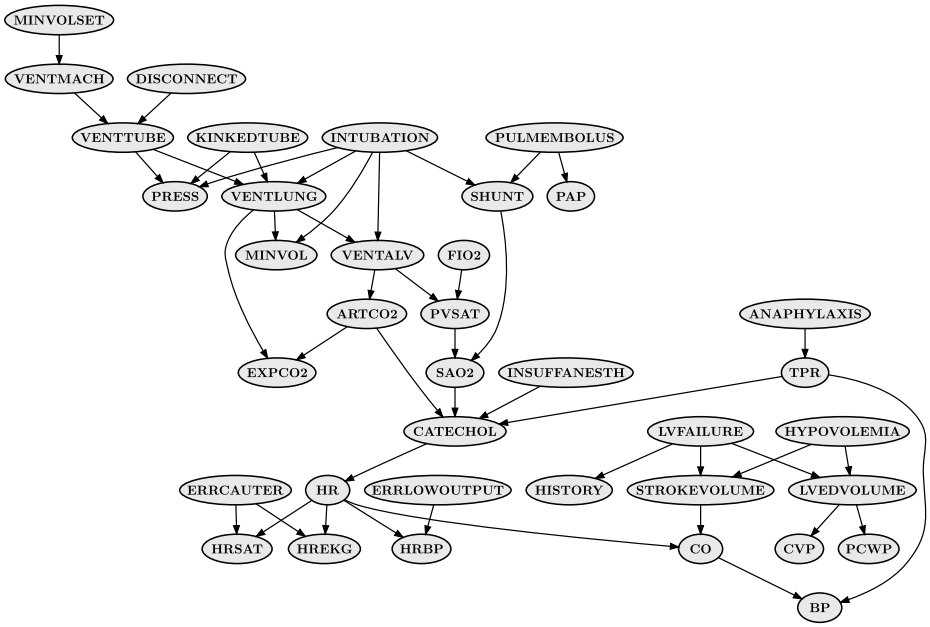


Fig. 5: Structure of the Alarm network structure used to compare the CPC and CBIC algorithms.

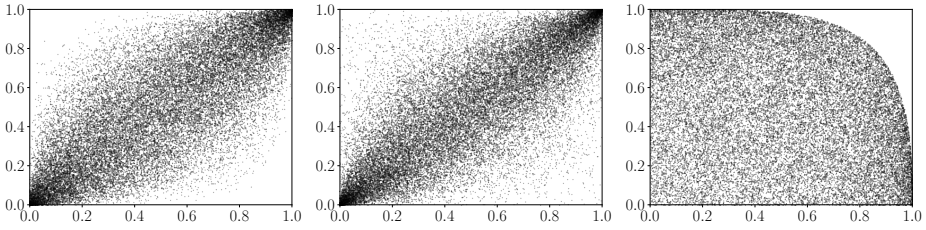
5 Experimental Results

This section presents the results of the comparison between CPC and CBIC methods[5]. The CBIC method used for experiments is an improved version [18] that uses the decomposition of the BIC score as a sum of mutual information (see page 802 of [17]). However, the maximum likelihood parameters are still computed as described above and the score maximization still relies on a TABU list method. The experiments have been carried out with the C++ libraries aGrUM [12], which allows to build graphical models, and OpenTURNS [1] which allows to model continuous multivariate probabilistic distributions.

5.1 Simulation setup

The two algorithms have been tested on data generated either from the Alarm network structure [3], which is represented on figure 5, or from random Bayesian networks. While the Alarm network is a discrete BN, only its structure is used here in order to generate data from it. It is used in order to have a real-world structure whereas random structures are used for more generality. The random structures are generated following [14] which proposes to build a MCMC converging to a uniform distribution over the set of DAGs with a given number of node and arc.

[5] While linear Gaussian model is the standard when learning BNs with continuous variables, it has not been compared to our model since it turns out to be less efficient than the CBIC method [9].



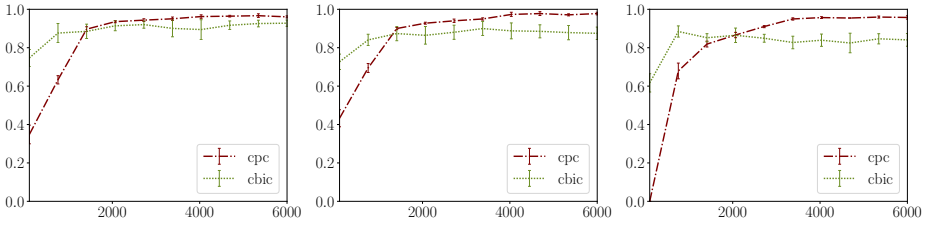
(a) Gaussian copula (b) Student copula (c) Dirichlet copula

Fig. 6: Samples from Gaussian, Student and Dirichlet copula densities. The correlation parameter of the Gaussian copula is set to $\rho = 0.8$, the Student copula is taken with $\nu = 5$ degrees of freedom and correlation parameter $\rho = 0.8$, the Dirichlet copula parameters are set to $\boldsymbol{\alpha} = (1/3, 2/3, 1)$.

For a random DAG with d nodes, it will contains $1.2 \times d$ arcs in order to have the same arc density than the Alarm structure. Once a structure is selected (ALARM or random), the local copulas of the CBN are parametrized using three models: Gaussian, Student or Dirichlet. These have been chosen in order to build worst-case scenarios for our algorithm and compare its performances with the CBIC algorithm when dealing with Gaussian or Student data. In turn, the Dirichlet copula has been chosen in order to challenge our algorithm because of its restrained support (see Figure (6)). Moreover, the three models have been parametrized such that it induces strong correlations between variables. Gaussian and Student copulas are parameterized such that their correlation matrices have off-diagonal parameters all set to $\rho = 0.8$ and the number of degrees of freedom ν is set to 5 for Student copulas. As for Dirichlet copulas, they are parameterized with $\boldsymbol{\alpha} = (\frac{1}{d+1}, \frac{2}{d+1}, \ldots, \frac{d}{d+1}, 1)$. Figure (6) shows a parametrization for the two dimensional case. The obtained CBNs are then sampled using the forward sampling procedure described in [17]. In short, the factors R_i associated to each node X_i are sampled using the inverse transform method and following a topological order over the variables $\boldsymbol{X}$, that is when $X_i \to X_j$ then $X_i < X_j$.

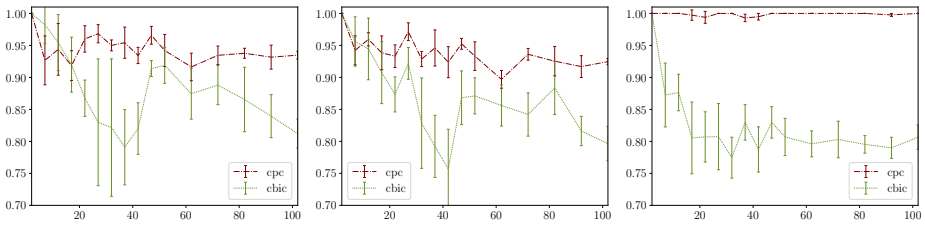
5.2 Skeleton performances

The structural performances of the two learning algorithms have been computed by comparing the skeleton of the learned graph with the skeleton of the reference structure that has been used to generate the data. Precision (P) is the proportion of learned edges that are actually in the reference structure while recall (R) is the proportion of edges that are in the reference structure that have been recovered. The F-score is then defined as $\mathrm{F} = 2\mathrm{PR}/(\mathrm{P}+\mathrm{R})$. If the reference skeleton has been perfectly retrieved, the value of the F-score is 1. Figure 7 shows the evolution of the F-score with respect to the sample size for the Alarm structure while Figure 8 shows the evolution of the F-score with respect to the size of the random structures using a sample of size $M = 10^4$. The case of Alarm structure, CBIC converges faster than CPC but giving enough data, CPC has a higher F-score value, even for Gaussian data. As for the random structures, the CPC method has also better results and is even almost always retrieving the skeleton of reference in the case of



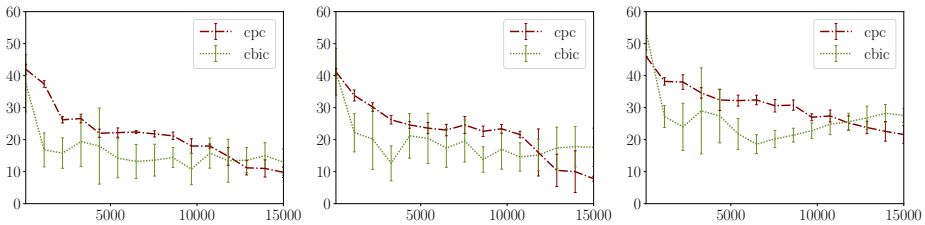
(a) Gaussian case (b) Student case (c) Dirichlet case

Fig. 7: Evolution of the F-score for CBIC (dotted green line) and CPC (dot-dashed red line) methods with respect to the size of the dataset. The results are averaged over 5 restarts with different data sets generated from the ALARM network structure.



(a) Gaussian case (b) Student case (c) Dirichlet case

Fig. 8: Evolution of the F-score for CBIC (dotted green line) and CPC (dot-dashed red line) methods with respect to the dimension of the random graphs. The results are averaged over 2 different random graphs of the same dimension and over 5 different data sets of size $M = 10000$.



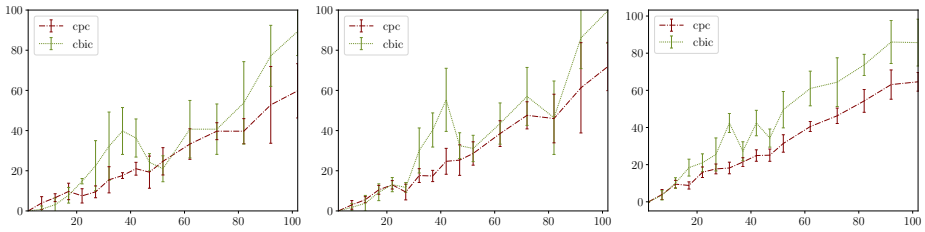
(a) Gaussian case (b) Student case (c) Dirichlet case

Fig. 9: Evolution of the SHD for CBIC (dotted green line) and CPC (dot-dashed red line) methods with respect to the size of the dataset. The results are averaged over 5 restarts with different data sets generated from ALARM network structure.

Dirichlet data. It can also be observed that CPC is not sensitive to the model that generated the data, illustrating the strength of a non-parametric method. Finally, looking at the standard deviation, CPC seems to be more stable than CBIC.

5.3 CPDAG performances

In order to score the oriented structure, structural hamming distance [6] has been used. This metric works on the completed partially directed acyclic graphs (CPDAG) that represents the Markov class equivalences of the DAG [17] and



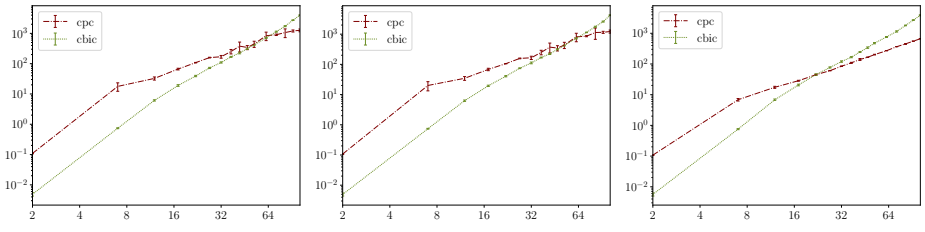
(a) Gaussian case (b) Student case (c) Dirichlet case

Fig. 10: Evolution of the SHD for CBIC (dotted green line) and CPC (dot-dashed red line) methods with respect to the dimension of the random graphs. The results are averaged over 2 different random graphs of the same dimension and over 5 different data sets of size $M = 10000$.

counts the numbers of elementary operations that are needed to obtain the reference structure from the estimated one. Those transformations are edge insertions, deletions and flipping. Figure 9 shows the evolution of the F-score with respect to the sample size for the Alarm structure while Figure 10 shows the evolution of the F-score with respect to the size of the random structures using a sample of size $M = 10^4$. The same observations can be made from the SHD evolution than from the F-score evolution. Indeed, for the Alarm structure, CBIC converges faster than CPC but CPC seems to converge to a lower SHD value. However, this time CPC needs a lot more data to do so. Moving to random structures, CPC has always better or equivalent performances than CBIC. In particular, CPC has better results for Dirichlet data and its results are, once again, independent from the generative model. Finally, CBIC is less stable than CPC looking at standard deviation but also on average.

5.4 Time complexity

The learning time with respect to the number of nodes in random structures has been used to compare the time complexities of the two methods. The results are obtained using samples of size $M = 10^4$ and are shown on figure 11.



(a) Gaussian case (b) Student case (c) Dirichlet case

Fig. 11: Learning time in seconds for CBIC (dotted green line) and CPC (dot-dashed red line) methods with respect to the dimension of the random graphs. The results are averaged over 2 different random graphs of the same dimension and over 5 different data sets of size $M = 10^4$.

While the CBIC method is faster for small structures, it is the CPC that is faster for large ones. However, the scale being logarithmic it has to be noted that the difference in learning time is in fact almost negligible in the case of small structures unlike for large ones where the difference can be hours.

6 Conclusion and Future Work

CBN is a promising model for dealing with continuous data in the BN context and for dealing with high-dimensional copula functions. One of the strength of the model is that it allows to use similar techniques used to learn the structure of classical BNs for the structure of a CBN. In this regard, [9] proposed a parametric method using the BIC score. In turn, we proposed a constraint based method which uses a PC algorithm and a non-parametric CI test, thus making no assumptions on the model that generated the data. The experimental part illustrated this last property since, unlike CBIC method, CPC can deal with data far from the Gaussian model such as Dirichlet data. In addition, the time complexity of CBIC growing fast, it makes it impractical to learn high dimensional structures (> 50 nodes).

Concerning future works, we plan to test our method on application cases in order to complete the results presented here. However, in real world data sets, variables are often both discrete and continuous. For this reason, it would be interesting to extend CBNs and the learning methods to mixed data. [16] introduced a hybrid CBN model which uses transformations in order to obtain continuous variables from discrete ones. Then, it could be easy to extend the learning methods presented here.

7 Declarations

7.1 Funding

This work was partially supported by Airbus Research through the AtRandom project (CRT/VPE/XRD).

7.2 Conflict of interest

The authors declare that they have no conflict of interest.

7.3 Availability of data and material

The source code to generate data and figure is provided on the GitHub repository MLasserre/otagrum-experiments.

7.4 Code availability

The source code for CBIC and CPC methods is available on the GitHub repository openturns/otagrum.

References

1. Baudin, M., Dutfoy, A., Iooss, B., Popelin, A.L.: Openturns: An industrial software for uncertainty quantification in simulation (2015)
2. Bedford, T., Cooke, R.M., et al.: Vines–a new graphical model for dependent random variables. The Annals of Statistics **30**(4), 1031–1068 (2002)
3. Beinlich, I.A., Suermondt, H.J., Chavez, R.M., Cooper, G.F.: The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In: AIME 89, pp. 247–256. Springer (1989)
4. Bouezmarni, T., Rombouts, J., Taamouti, A.: A nonparametric copula based test for conditional independence with applications to granger causality. Economics working papers, Universidad Carlos III, Departamento de Economía (2009)
5. Bouezmarni, T., Rombouts, J.V., Taamouti, A.: Asymptotic properties of the bernstein density copula estimator for α-mixing data. Journal of Multivariate Analysis **101**(1), 1 – 10 (2010). DOI https://doi.org/10.1016/j.jmva.2009.02.014
6. Colombo, D., Maathuis, M.H.: Order-independent constraint-based causal structure learning. The Journal of Machine Learning Research **15**(1), 3741–3782 (2014)
7. Czado, C.: Pair-copula constructions of multivariate copulas. In: Copula theory and its applications, pp. 93–109. Springer (2010)
8. Deheuvels, P.: La fonction de dépendance empirique et ses propriétés. un test non paramétrique d'indépendance. Bulletins de l'Académie Royale de Belgique **65**(1), 274–292 (1979)
9. Elidan, G.: Copula bayesian networks. In: Advances in neural information processing systems, pp. 559–567 (2010)
10. Genest, C., Favre, A.C.: Everything you always wanted to know about copula modeling but were afraid to ask. Journal of hydrologic engineering **12**(4), 347–368 (2007)
11. Glover, F., Laguna, M.: Tabu Search, pp. 2093–2229. Springer US, Boston, MA (1998). DOI 10.1007/978-1-4613-0303-9_33. URL https://doi.org/10.1007/978-1-4613-0303-9_33
12. Gonzales, C., Torti, L., Wuillemin, P.H.: aGrUM: a Graphical Universal Model framework. In: International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems, Proceedings of the 30th International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems. Arras, France (2017). URL https://hal.archives-ouvertes.fr/hal-01509651
13. Huang, J.C.: Cumulative distribution networks: Inference, estimation and applications of graphical models for cumulative distribution functions. Citeseer (2009)
14. Ide, J.S., Cozman, F.G.: Random generation of bayesian networks. In: Brazilian symposium on artificial intelligence, pp. 366–376. Springer (2002)
15. Joe, H.: Multivariate models and multivariate dependence concepts. CRC Press (1997)
16. Karra, K., Mili, L.: Hybrid copula bayesian networks. In: Conference on Probabilistic Graphical Models, pp. 240–251 (2016)
17. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
18. Lasserre, M., Lebrun, R., Wuillemin, P.H.: Learning continuous high-dimensional models using mutual information and copula bayesian networks. AAAI (2021). Forthcoming
19. Lauritzen, S.L., Wermuth, N.: Graphical models for associations between variables, some of which are qualitative and some quantitative. The annals of Statistics pp. 31–57 (1989)
20. Lindskog, F., McNeil, A., Schmock, U.: Kendall's tau for elliptical distributions. In: Credit Risk, pp. 149–156. Springer (2003)
21. Nelsen, R.B.: An introduction to copulas. Springer Science & Business Media (2007)
22. Rousseeuw, P.J., Molenberghs, G.: Transformation of non positive semidefinite correlation matrices. Communications in Statistics–Theory and Methods **22**(4), 965–984 (1993)
23. Sancetta, A., Satchell, S.: The bernstein copula and its applications to modeling and approximations of multivariate distributions. Econometric Theory **20**(03), 535–562 (2004)
24. Schwarz, G.: Estimating the dimension of a model. The annals of statistics **6**(2), 461–464 (1978)
25. Sklar, A.: Fonctions de répartition à n dimensions et leurs marges. Publ. Inst. Statist. Univ. Paris **8**, 229–231 (1959)
26. Spirtes, P., Glymour, C.N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., Richardson, T.: Causation, prediction, and search. MIT press (2000)
27. Su, L., White, H.: A nonparametric hellinger metric test for conditional independence. Econometric Theory **24**(4), 829–864 (2008)
28. Wan, J., Zabaras, N.: A probabilistic graphical model based stochastic input model construction. J. Comput. Physics **272**, 664–685 (2014)