Learning Continuous High-Dimensional Models using Mutual Information and Copula Bayesian Networks

Marvin Lasserre,[1] **Régis Lebrun,** [2] **Pierre-Henri Wuillemin** [1]

[1] Laboratoire d'Informatique de Paris 6, 4 place Jussieu, 75005 Paris, France
[2] Airbus Central Research & Technology, 22 rue du Gouverneur Général Eboué, 92130 Issy les Moulineaux, France
marvin.lasserre@lip6.fr, regis.lebrun@airbus.com, pierre-henri.wuillemin@lip6.fr

Abstract

We propose a new framework to learn non-parametric graphical models from continuous observational data. Our method is based on concepts from information theory in order to discover independences and causality between variables: the conditional and multivariate mutual information (such as (Verny et al. 2017) for discrete models). To estimate these quantities, we propose non-parametric estimators relying on the Bernstein copula and that are constructed by exploiting the relation between the mutual information and the copula entropy (Ma and Sun 2011; Belalia et al. 2017). To our knowledge, this relation is only documented for the bivariate case and, for the need of our algorithms, is here extended to the conditional and multivariate mutual information. This framework leads to a new algorithm to learn continuous non-parametric Bayesian networks. Moreover, we use this estimator to speed up the BIC algorithm proposed in (Elidan 2010) by taking advantage of the decomposition of the likelihood function in a sum of mutual information (Koller and Friedman 2009). Finally, our method is compared in terms of performances and complexity with other state of the art techniques to learn Copula Bayesian Networks and shows superior results. In particular, it needs less data to recover the original structure and generalizes better on data that are not sampled from Gaussian distributions.

1 Introduction

Modeling multivariate continuous distributions is a task of central interest in statistics and machine learning with many applications in science and engineering. In general, high-dimensional distributions are difficult to manipulate and may lead to intractable computations. Bayesian networks (BNs) exploit conditional independences between random variables to reduce the complexity of a joint probability distribution by expressing it as a set of conditional probability distributions (CPDs) of lower dimension. These independences are encoded by a Directed Acyclic Graph (DAG) (Koller and Friedman 2009) and to each node is associated a CPD. The representation of CPD is complex and led to numerous and very different solutions: discretization, parametric representation (for instance using the Gaussian hypothesis), approximation using truncated basis functions (Shenoy and West 2011; Langseth et al. 2012), etc.

On the other hand, copula function allows to model the dependence structure between continuous variables, ruling out the marginal behavior of each variable. From a constructive perspective, this allows to dissociate the choice of the marginals from the choice of the dependence structure. In practice, however, copulas are limited to a few variables and constructing or manipulating high-dimensional ones is difficult.

The Copula Bayesian Network model (CBN) (Elidan 2010) takes advantage of both copula theory and BNs to model continuous high-dimensional multivariate distributions. Whereas there have been many attempts to merge the two frameworks such as the pair-copula construction (Czado 2010), the Vine model (Bedford, Cooke et al. 2002) or the cumulative distribution network (Huang 2009), the CBN model is the most attractive since it uses the same graphical language as a classical BN.

This paper focuses on learning continuous graphical models from data. While there is many learning algorithm in the discrete case (Neapolitan et al. 2004), they can hardly be extended to the continuous models (Romero, Rumí, and Salmerón 2006). This is mainly due to the number of parameters of these models which makes the computing of scores or statistics difficult. They can be applied for simpler models such as the linear Gaussian model (Lauritzen and Wermuth 1989), but the model itself lacks of expressiveness. For the reason given in the last paragraph, the CBN model gives access to similar learning techniques as for discrete BNs. Score based (Elidan 2010) and constraint-based (Lasserre, Lebrun, and Wuillemin 2020) methods have been proposed to learn a CBN structure from data. The latter, named CPC, relies on a PC-algorithm and has shown better performances than the former which relies on a BIC score and a local search optimization. However, it is well-known (Colombo and Maathuis 2014) that such constraint-based methods suffer from the need of an ordering over the variables and can lead to significantly different results. In the discrete case, the MIIC algorithm (Verny et al. 2017) avoid this ordering by driving the algorithm with a ranking relying on the mutual information.

The contributions of this paper are the following. First, we extend the link between the copula entropy and the mutual information proved in (Ma and Sun 2011) to the conditional and three-point mutual information. We then describe non-

parametric estimators of these quantities based on the empirical Bernstein copula. Next, we use these estimators (i) to speed up the BIC algorithm presented in (Elidan 2010) using the decomposition of the likelihood function in a sum of mutual information and (ii) to propose a new learning algorithm for non-parametric CBN. Finally, a benchmark is made on synthetic data between these methods and the CPC algorithm in terms of structural scores and time complexity.

The paper is organized as follows. Section 2 reviews the necessary concepts about copulas and presents the CBN model. Section 3 extends the link between mutual information and copula entropy and then introduces the estimators that are used by our continuous version of the MIIC algorithm and the fastened version of the BIC algorithm in Section 4. These methods are compared with the CPC algorithm in Section 5. Section 6 concludes the paper.

2 Copula Bayesian Networks

Consider a random vector $\boldsymbol{X} = (X_1, \ldots, X_D)$ whose components X_i take values x_i from domains Ω_i. A BN structure $\mathcal{G}$ is a DAG whose nodes $\mathbf{X} = \{X_1, \ldots, X_D\}$ represent random variables. Let $\mathbf{Pa}_i$ and $\mathbf{ND}_i$ respectively denote the parents and the non-descendants of X_i in $\mathcal{G}$. A multivariate probability distribution P over variables $\mathbf{X}$, is said to factorize according to $\mathcal{G}$, if it can be written as

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

Thus, $\mathcal{G}$ encodes the set of independencies:

$$\mathcal{I}(\mathcal{G}) = \{(X_i \perp \mathbf{ND}_i|\mathbf{Pa}_i)\}. \quad (2)$$

A BN is a pair $\mathcal{B} = (\mathcal{G}, P)$ where $\mathcal{G}$ is defined as previously and P is a joint probability distribution factorizing over $\mathcal{G}$. To each node X_i of the BN structure is associated its corresponding Conditional Probability Distribution (CPD) $P(X_i|\mathbf{Pa}_i)$ that appears in the factorization of P. CPDs are usually represented via conditional probability tables in the discrete case whereas there is no general model for the continuous case. The linear Gaussian model (Lauritzen and Wermuth 1989) $f(x_i|\mathbf{pa}_i) = \mathcal{N}(\beta_{i0} + \beta_i^T \mathbf{pa}_i; \sigma_i^2)$ allows fast probabilistic computations and estimations but lacks of expressiveness. On the other side, models based on mixtures of functions (Langseth et al. 2012) are expressive but hard to learn.

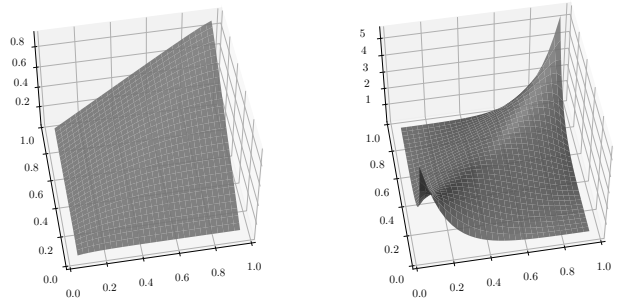
The CBN model introduced in (Elidan 2010), parametrized the CPDs with copula functions whose definition is now given:

Definition 1 (Copula) *Let $\mathbf{U} = \{U_1, \ldots, U_D\}$ be a random vector whose components U_i are uniformly distributed on $\mathbb{I}$. A D-dimensional copula function is a (cumulative) distribution function on $\mathbb{I}^D$:*

$$C(u_1, \ldots, u_D) = \mathbb{P}(U_1 \leq u_1, \ldots, U_D \leq u_D)$$

As a distribution function on $\mathbb{I}^D$ with uniform marginals, the copula respects the following properties:

1. $C(u_1, \ldots, u_D) = 0$ *if there exists i such that* $u_i = 0$,
2. $C(1, \ldots, 1) = 1$,
3. $C(1, \ldots, u_i, \ldots, 1) = u_i$.



(a) Distribution function (b) Density function

Figure 1: Two-dimensional Gaussian copula with a correlation parameter $\rho_{12} = 0.8$ (plots obtained using numerical integration).

The relation between the joint distribution and its univariate marginals is a central result of copula theory (Sklar 1959):

Theorem 1 (Sklar 1959) *Let F be any multivariate distribution function over a random vector $\mathbf{X}$ and F_i its one-dimensional marginal distributions* [1]*. There exists a copula function C such that*

$$F(x_1, \ldots, x_D) = C(F_1(x_1), \ldots, F_D(x_D)). \quad (3)$$

Furthermore, if each F_i is continuous then C is unique.

Sklar's theorem may be used to construct new copulas from known multivariate distributions by inverting (3) :

$$C(u_1, \ldots, u_D) = F(F_1^{-1}(u_1), \ldots, F_d^{-1}(u_D))$$

where $u_i = F(x_i)$. Taking $F = \Phi_\rho$, the multivariate standard Gaussian distribution with correlation matrix ρ, we obtain the Gaussian copula (Nelsen 2007) (see Figure 1) :

$$C_G(u_1, \ldots, u_D) = \Phi_\rho(\phi^{-1}(u_1), \ldots, \phi^{-1}(u_D))$$

where ϕ is the univariate standard Gaussian distribution. The copula density function is obtained by derivation of C: $c(u_1, \ldots, u_D) = \frac{\partial^D C(u_1, \ldots, u_D)}{\partial u_1 \ldots \partial u_D}$. Similarly, deriving equation (3) leads to the following corollary:

Corollary 1.1 *Let f be any multivariate density function over $\boldsymbol{X}$ and c its copula density. The copula density relates the joint density to its 1-dimensional marginals f_i:*

$$f(x_1, \ldots, x_D) = c(F_1(x_1), \ldots, F_D(x_D)) \prod_{i=1}^{D} f_i(x_i). \quad (4)$$

This formula generalizes the case of independent variables where the joint distributions may be decomposed as a product of its marginals: $f(\boldsymbol{x}) = \prod_{i=1}^{D} f(x_i)$. Then, as the marginals encode the individual behavior of each variables, the copula function and its density encode the dependence between random variables. This is interesting from a constructive perspective since the choice of marginals can be separated from the choice of the dependence structure. This leads to the definition of a CBN as given by (Elidan 2010) :

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

Definition 2 (CBN) *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 density functions* $c_i\left(F(x_i), \{F(pa_{ik_i})\}\right)$, *where* $k_i = |\mathbf{pa}_i|$, *and* Θ_f *is a set of marginal densities* f_i. *To each node of* $\mathcal{G}$, *a copula and a marginal function is associated.* $f(\mathbf{x})$ *then factorizes as*

$$f(\mathbf{x}) = \prod_{i=1}^{D} R_{c_i}(F(x_i), \{F(pa_{ik_i})\}) f(x_i), \tag{5}$$

where $R_{c_i}(F(x_i), \{F(pa_{ik_i})\}) = \frac{c_i(F(x_i), \{F(pa_{ik_i})\})}{c_i(\{F(pa_{ik_i})\})}$.

An example of CBN is given on figure 2.

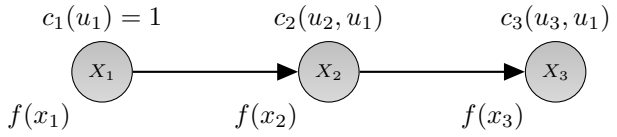


Figure 2: A CBN with three variables. $\forall i, f(x_i)$ is a marginal density and $c_i(.)$ is a copula. The joint density is: $f(x_1, x_2, x_3) = f(x_1)c_2(F(x_2), F(x_1))f(x_2)c_3(F(x_3), F(x_2))f(x_3)$

3 Copula and Continuous Information Theory

It has been shown that continuous mutual information is the negative copula entropy (Ma and Sun 2011). We generalize this relation for multivariate and conditional mutual information and use it to define estimators that will be used in the next section to implement a continuous MIIC.

Before introducing the mutual information, we recall the definitions of differential and relative entropy.

Definition 3 (Differential entropy) *The differential entropy* h *over a set* $\boldsymbol{S} \subseteq \boldsymbol{X}$ *of variables is given by:*

$$h(\boldsymbol{S}) = -\int_{\Omega_{\boldsymbol{S}}} f(\boldsymbol{s}) \log f(\boldsymbol{s}) \mathrm{d}\boldsymbol{s}.$$

Definition 4 (Relative entropy) *The relative entropy* $D(f||g)$ *between two densities* f *and* g *is defined by*

$$D(f||g) = \int_{\Omega_{\mathbf{X}}} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{g(\mathbf{x})} \mathrm{d}\mathbf{x}. \tag{6}$$

The mutual information is defined as the relative entropy between the joint density and its marginals.

Definition 5 (Mutual information) *The mutual information between two random variables* X_i *and* X_j *is given by:*

$$\begin{aligned} I(X_i; X_j) &= D(f(x_i, x_j)||f(x_i)f(x_j)) \\ &= \iint_{\Omega_i \times \Omega_j} f(x_i, x_j) \log \frac{f(x_i, x_j)}{f(x_i)f(x_j)} \, \mathrm{d}x_i \mathrm{d}x_j. \end{aligned} \tag{7}$$

As a consequence, since $D(f||g) \geq 0$ (Cover and Thomas 2012), the mutual information is also positive. Moreover, it is vanishing if and only if the variables are independent which makes it a good measure of dependence. Various dependence measures such as Spearman's rho or Kendall's tau are functionals of the copula density (Genest and Favre 2007). The mutual information makes no exception as it is the negative copula entropy(Ma and Sun 2011):

Definition 6 (Copula entropy) *The copula entropy* h_c *of a random vector* $\boldsymbol{U}$ *is given by:*

$$h_c(\boldsymbol{U}) = -\int_{[0,1]^{|\boldsymbol{U}|}} c(\boldsymbol{U}) \log c(\boldsymbol{U}) \mathrm{d}\boldsymbol{u} \tag{8}$$

Theorem 2 *The mutual information is the negative copula entropy*

$$I(X_i, X_j) = -h_c(X_i, X_j). \tag{9}$$

We now extend this relation to the conditional mutual information whose definition is given by:

Definition 7 (Conditional mutual information) *The conditional mutual information between* X_i *and* X_j *conditioned on a set of variables* $\boldsymbol{U} \subseteq \boldsymbol{X}$ *is defined by:*

$$\begin{aligned} I(X_i; X_j|\boldsymbol{U}) &= \mathbb{E}_{\boldsymbol{U}}[D(f(x_i, x_j|\boldsymbol{u})||f(x_i|\boldsymbol{u})f(x_j|\boldsymbol{u})] \\ &= \iiint_{\Omega_i \times \Omega_j \times \Omega_{\boldsymbol{U}}} f(x_i, x_j, \boldsymbol{u}) \\ &\qquad \times \log\left(\frac{f(x_i, x_j, \boldsymbol{u})f(\boldsymbol{u})}{f(x_i, \boldsymbol{u})f(x_j, \boldsymbol{u}))}\right) \mathrm{d}x_i \mathrm{d}x_j \mathrm{d}\boldsymbol{u}. \end{aligned}$$

By its definition, the conditional mutual information is positive using the positiveness of the relative entropy. It is straightforward to show from its definition that

$$\begin{aligned} &I(X_i; X_j|\boldsymbol{U}) = \\ &\quad h(X_i, \boldsymbol{U}) + h(X_j, \boldsymbol{U}) - h(X_i, X_j, \boldsymbol{U}) - h(\boldsymbol{U}). \end{aligned} \tag{10}$$

Using the following lemma proved in (Ma and Sun 2011),

Lemma 3 *The differential entropy can be written as the sum of the entropy of each variable and the copula entropy:*

$$h(X_1, \ldots, X_D) = \sum_{i=1}^{D} h(X_i) + h_c(X_1, \ldots, X_D) \tag{11}$$

the relation between conditional information and copula entropy is easily obtained.

Theorem 4 *The conditional mutual information is related to the copula entropy by:*

$$\begin{aligned} I(X_i; X_j|\boldsymbol{U}) =& h_c(X_i, \boldsymbol{U}) + h_c(X_j, \boldsymbol{U}) \\ &- h_c(X_i, X_j, \boldsymbol{U}) - h_c(\boldsymbol{U}) \end{aligned} \tag{12}$$

The definition of mutual information has been extended to a set of variables by (McGill 1954):

$$I(X_1; \ldots; X_D) = \sum_{\boldsymbol{T} \subseteq \boldsymbol{X}} (-1)^{|\boldsymbol{T}|+1} h(\boldsymbol{T}).$$

Especially, the case $n = 3$ called three-point information and which is of interest for the next section, is given by:

$$\begin{aligned} I(X_i; X_j; X_k) = & h(X_i) + h(X_j) + h(X_k) \\ & - h(X_i, X_j) - h(X_i, X_k) - h(X_j, X_k) \\ & + h(X_i, X_j, X_k). \end{aligned} \tag{13}$$

One important remark is that the generalized mutual information is no longer positive and can be negative. It turns out that the negativity of the mutual information between three variables is the signature of a v-structure in the associated graph. As with conditional mutual information, we want to relate the three-point information to the copula entropy and more precisely, to the conditional three-point information.

Looking closely to 7, 10 and 13, it can be proved that the three-point information verify the relation

$$I(X_i; X_j; X_k) = I(X_i; X_j) - I(X_i; X_j|X_k).$$

This is sometimes taken as a definition of the three-point information and is used here to define the conditional three-point information:

Definition 8 (Conditional three-point information) *The conditional three-point information is defined as:*

$$I(X_i; X_j; X_k|\mathbf{U}) = I(X_i; X_j|\mathbf{U}) - I(X_i; X_j|X_k, \mathbf{U})$$

Replacing $I(X_i; X_j|\mathbf{U})$ by (12) and using the same relation with $\{X_k, \boldsymbol{U}\}$ in place of $\mathbf{U}$, the sought result can be derived for the conditional three-point information:

Theorem 5 *The conditional three-point information is related to the copula entropy by:*

$$\begin{aligned} I(X_i; X_j; X_k|\boldsymbol{U}) = \\ h_c(X_i, \boldsymbol{U}) + h_c(X_j, \boldsymbol{U}) + h_c(X_k, \mathbf{U}) \\ - h_c(X_i, X_j, \boldsymbol{U}) - h_c(X_i, X_k, \boldsymbol{U}) - h_c(X_j, X_k, \boldsymbol{U}) \\ + h_c(X_i, X_j, X_k, \boldsymbol{U}) - h_c(\boldsymbol{U}). \end{aligned} \quad (14)$$

In the case where $\boldsymbol{U} = \emptyset$,it simplifies into

$$\begin{aligned} I(X_i; X_j; X_k) = h_c(X_i, X_j, X_k) - h_c(X_i, X_j) \\ - h_c(X_j, X_k) - h_c(X_i, X_k) \end{aligned} \quad (15)$$

Finally, all the previous quantities can be estimated from a data set of size M using the following estimator of the copula entropy:

$$\hat{h}_c(\mathbf{X}) = -\sum_{m=1}^{M} \hat{c}(\mathbf{x}[m]) \log(\hat{c}(\mathbf{x}[m])), \quad (16)$$

where $\hat{c}$ can be any copula model estimated from the data. In order to obtain a non-parametric estimator, the empirical Bernstein copula (Sancetta and Satchell 2004) will be used in our version of MIIC but an alternative version using estimated Gaussian copula will be used for comparison.

4 Learning Copula Bayesian Networks

CBNs share the same graphical interpretation of independences than classical BNs (i.e. the d-separation), allowing to use similar techniques to learn their structures from data. These algorithms can be roughly divided into two classes: score-based and constraint- based methods. Score based methods view the learning task as a model selection and is guided by a scoring function to measure how well the model fits the data. However, the set of DAG structures being superexponential in the number of nodes, local search methods are needed to maximize the score. Constraint-based methods on the other hand consider the graph as a set of conditional independences (2) and use CI tests to obtain information about the underlying structure. The MIIC algorithm presented in this section is a hybrid method that follows a constraint-based outline mixed with an information theoretic score. This score allows to avoid the arbitrary ordering over the variables that is made in constraint-based methods, and which can lead to different results.

Improving Continuous BIC (CBIC)

In (Elidan 2010), a score-based method is used to learn the structure of a CBN. The proposed score is the well-known Bayesian information criterion (BIC) (Schwarz 1978). Its expression on a CBN structure $\mathcal{G}$ is given by :

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

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 (5), we have :

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

where $u_i = F(x_i)$ and $\pi_{ij} = F(\mathrm{pa}_{ij})$. The R_{c_i}'s are computed using Gaussian copula parametrized by a 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 (Lindskog, McNeil, and Schmock 2003). Finally, the BIC score is maximized using a TABU list algorithm with random restarts (Glover and Laguna 1998). The downside of this technique is that the score needs to be computed over the entire graph every time a local modification is done. As an improvement of this algorithm, we propose here to replace the factor R_i by its expression in the likelihood function which gives:

$$\ell(\mathcal{D} : \hat{\theta}, \mathcal{G}) = M \sum_{i=1}^{D} \hat{I}(X_i; \mathbf{Pa}_i),$$

where $\hat{I}(X_i; \mathbf{Pa}_i) = \hat{h}_c(X_i, \mathbf{Pa}_i) - \hat{h}_c(\mathbf{Pa}_i)$. This last equation allows to compute the variation of the score for each operation made during the local search in the graph space, hence avoiding to compute it over the entire graph (see p.818 of (Koller and Friedman 2009) for more details).

Continuous PC Algorithm (CPC)

The PC algorithm introduced by (Spirtes et al. 2000) can be divided in three main steps : skeleton learning, v-structure search and constraint propagation. The skeleton search consists in removing edges from the complete non-oriented graph on $\mathbf{X}$. To do so, a Conditional Independence (CI) test is used between pairs of connected variables (X_i, X_j) given a subset $\boldsymbol{S}$ of their common neighbors $\boldsymbol{Adj}(X_i, X_j)$. If it

is found that $X_i \perp X_j | \boldsymbol{S}$, $\boldsymbol{S}$ is then noted $\mathbf{Sepset}(X_i, X_j)$ and the edge is removed from the graph. The tests are made by increasing size $l = |\boldsymbol{S}|$ of conditioning set until all adjacency sets in the current graph are smaller than l. Once this first step completed, triplets $X_i - X_k - X_j$ such that X_i and X_j are not neighbors and X_k is not in $\mathbf{Sepset}(X_i, X_j)$, are oriented as v-structures: $X_i \rightarrow X_k \leftarrow X_j$. 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 order in which the pairs of variables and their adjacency sets are processed is not unique. Yet, it has a direct effect on the skeleton search and the separating sets. Indeed, the skeleton is updated after each edge removal and the adjacency sets of variables that are treated after may change.Thus, changing the order can lead to different CI tests. For additional information on the PC algorithm, see page 84 of (Spirtes et al. 2000) and (Colombo and Maathuis 2014). A continuous version relying on a non-parametric CI test, named CPC, has been proposed in (Lasserre, Lebrun, and Wuillemin 2020) to learn CBN structures.

A New Learning Algorithm for CBN: Continuous MIIC (CMIIC)

MIIC algorithm consists in the same three main steps than the CPC algorithm: skeleton learning, v-structure orientation and constraint propagation. However, it makes use of mutual information in order to rank the nodes and overcome the problem of the ordering discussed in the case of the PC-algorithm. It has been shown to be more efficient than constraint-based method in the discrete case (Verny et al. 2017) and we are extending it to continuous data.

The starting point of the algorithm is the likelihood function which has to be slightly adapted to the continuous case:

$$\mathcal{L}(\mathcal{D}|\mathcal{G}) = \prod_{m=1}^{M} f_{\mathcal{G}}(\boldsymbol{x}[m]) = \exp\left(-M\hat{H}(f, f_{\mathcal{G}})\right)$$

where $M\hat{H}(f, f_{\mathcal{G}}) = -\sum_{m=1}^{M} \log f_{\mathcal{G}}(\boldsymbol{x}[m])$ is the Monte-Carlo estimator of the cross-entropy between the model density $f_{\mathcal{G}}$ and the true density f that generated the data defined as:

$$H(f, f_{\mathcal{G}}) = \mathbb{E}_f[-\log f_{\mathcal{G}}(X)] = -\int_{\Omega_{\boldsymbol{X}}} f(\boldsymbol{x}) \log f_{\mathcal{G}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

The rank is then derived from the decomposition of the cross-entropy over the structure $\mathcal{G}$ and by computing ratio of likelihood function. Only its expression is reported here and the interested reader might find details about its origin in (Affeldt, Verny, and Isambert 2016). The rank is based on the probability for the triplet (X_i, X_j, X_k) to not be a v-structure conditioned on $\mathbf{U}$:

$$P_{\mathrm{nv}}(X_i; X_j; X_k|\mathbf{U}) = \left(1 + e^{-MI(X_i;X_j;X_k|\mathbf{U})}\right)^{-1}$$

and the probability that its base is X_i and X_j:

$$P_{\mathrm{b}}(X_i, X_j; X_k|\mathbf{U}) = \frac{1}{1 + \frac{e^{-MI(X_i;X_k|\mathbf{U})}}{e^{-MI(X_i;X_j|\mathbf{U})}} + \frac{e^{-MI(X_j;X_k|\mathbf{U})}}{e^{-MI(X_i;X_j|\mathbf{U})}}}$$

Algorithm 1: MIIC algorithm (Verny et al. 2017)

Input: Data set $\mathcal{D}$
Result: Structure $\mathcal{G}$

$\mathcal{G} \leftarrow$ complete undirected graph on $\mathbf{X}$;
// Skeleton search
forall *Edge* (X_i, X_j) **do**
  if $I'(X_i; X_j) < 0$ **then**
    Delete edge $X_i - X_j$ from G;
    $\mathbf{Sepset}(\mathbf{X_i}, \mathbf{X_j}) \leftarrow \{\}$;
  else
    $X_k \leftarrow \arg\max_{\boldsymbol{Adj}(X_i,X_j)} r(X_i, X_j; X_k|\{\})$;
while *There exists an edge* (X_i, X_j) *with* $r(X_i, X_j; X_k|\mathbf{U})$ **do**
  for *Top edge* (X_i, X_j) *with highest rank* $r(X_i, X_j; X_k|\mathbf{U})$ **do**
    Expand contributing set: $\mathbf{U} \leftarrow \mathbf{U} \cup \{X_k\}$;
    if $I'(X_i, X_j|\mathbf{U}) \leq 0$ **then**
      Delete edge $X - Y$ from G;
      $\mathbf{Sepset}(\mathbf{X_i}, \mathbf{X_j}) \leftarrow \mathbf{U}$;
    else
      $X_k \leftarrow \arg\max_{\boldsymbol{Adj}(X_i,X_j)} r(X_i, X_j; X_k|\mathbf{U})$;
    Sort the list of ranks $r(X_i, X_j; X_k|\mathbf{U})$;
  // V-structure search
  Sort list L of unshielded triples $X_i - X_k - X_j$ in decreasing order of $|I'(X_i; X_j; X_k|\mathbf{U})|$;
  repeat
    Take $(X_i, X_k, X_j) \in L$ with highest $|I'(X_i; X_j; X_k|\mathbf{U})|$ on which R_0 or R_1 operation rule can be applied;
    if $I'(X_i; X_j; X_k|\mathbf{U}) < 0$ **then**
      If (X_i, X_k, X_j) has no diverging orientation, apply $R_0 : \{X_i - X_k - X_j \,\&\, \mathrm{not}(X_i - X_j) \& X_k \notin \mathbf{Sepset}(\mathbf{X_i}, \mathbf{X_j})\} \Rightarrow \{X_i \rightarrow X_k \leftarrow X_j\}$
    else
      If (X_i, X_k, X_j) has one converging orientation, apply $R_1 : \{X_i \rightarrow X_k - X_j \,\&\, \mathrm{not}(X_i - X_j)\} \Rightarrow \{X_k \rightarrow X_j\}$
    Apply new orientations to all other $(X'_i, X'_k, X'_j) \in L$;
  until *no additional orientation can be obtained*;

Combining these two probabilities, the pairs of node (X_i, X_j) with the most likely contribution from a third node X_k can be ranked according to:

$$r(X_i, X_j; X_k|\mathbf{U}) = \max_{X_k \in \boldsymbol{X}} \left(\min\left[P_{\mathrm{nv}}(X_i; X_j; X_k|\mathbf{U}), P_{\mathrm{b}}(X_i, X_j; X_k|\mathbf{U})\right]\right).$$

MIIC algorithm is listed on (1). The conditional two-point and three-point mutual information terms appearing in the previous probabilities are computed using the equations and the copula entropy estimator of Section (3). Our estimators for conditional and three-point information are computed on finite size data sets and are then biased. For this reason, corrections based on criteria such than Normalized Maximum Likelihood (NML)(Shtar'kov 1987), Maximum Description Length (MDL) (Rissanen 1978) or BIC (Koller and Friedman 2009) are used in the discrete case. However, the first two criteria cannot be extended to continuous variables since there are diverging when taking the continuous limit. As for BIC, it cannot be applied to our case since it is only defined for parametric models. Consequently, we have decided to use a parameter α such that $I'(X_i; X_j|\mathbf{U}) = I(X_i; X_j|\mathbf{U}) - \alpha$ and $I'(X_i; X_j; X_k|\mathbf{U}) =$

$I(X_i; X_j; X_k|\mathbf{U}) + \alpha$. The fact that we add α in the case of three-point information means that we favor non v-structure over v-structure since a negative value of three-point information leads to a v-structure in the graph. This parameter can be considered as a confidence threshold : the more α decreases (and the more accurate is our test), the more data are needed to decide the independence. The value $\alpha = 0.01$ has been proved experimentally to be a good compromise between the data needed to learn independences and the confidence of the test.

5 Experimental Results

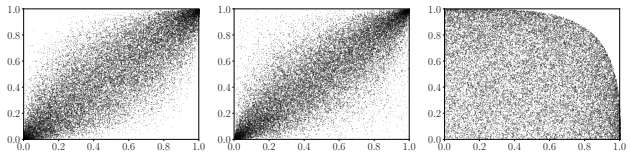
This section compares the results obtained from CBIC, CPC and CMIIC methods. Two models of copulas, Gaussian and Bernstein, are used to estimate the copula entropy with MIIC. This leads to two versions of the algorithm that will be denoted G-CMIIC and B-CMIIC. The comparison is made in terms of performances using F-score and structural Hamming distance and in terms of time complexity. These experiments have been carried out using the libraries aGrUM (Gonzales, Torti, and Wuillemin 2017) and OpenTURNS (Baudin et al. 2015) to respectively build graphical models and model continuous multivariate distributions.

Simulation Setup

The algorithms have been tested on data generated either from the ALARM network structure (Beinlich et al. 1989) in order to have a real-world structure, or from random Bayesian networks for more generality. The random Bayesian networks have been generated following (Ide and Cozman 2002) which proposes to build a MCMC converging to a uniform distribution on the set of DAGs with a desired number of nodes and arcs. For a given dimension D, a random graph contains $1.2 \times D$ arcs. Once a structure is selected (ALARM or random), the local copulas of the CBN are parametrized using three models: Gaussian, Student and Dirichlet. These models have been chosen in order to build worst-case scenarios for our algorithm and compare its performances with parametric ones 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.The three models have been parametrized such that it induces strong correlations between variables (correlation matrices having off-diagonal parameters set to 0.8, Dirichlet copula with $\boldsymbol{\alpha} = (1/D, 2/D, \ldots, 1)$). Figure (3) shows two-dimensional samples obtained using these parametrizations. The CBNs are then sampled using the forward sampling procedure described in (Koller and Friedman 2009).

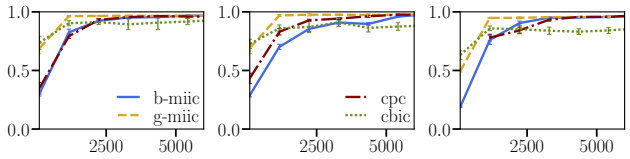
Skeleton Performances

The structural performances of the four learning algorithms have been computed by comparing the skeleton of the learned graph with the skeleton of the reference structure that have 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



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

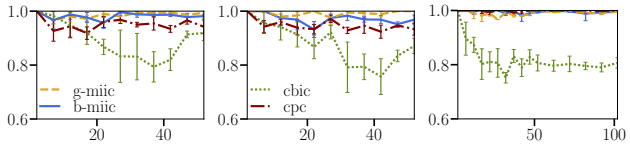
Figure 3: Samples from Gaussian, Student and Dirichlet copula densities. The correlation parameter of the Gaussian copula is set to $\rho = 0.8$, the Student copula is taken with $\nu = 5$ degrees of freedom and correlation parameter $\rho = 0.8$, the Dirichlet copula parameters are set to $\boldsymbol{\alpha}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



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

Figure 4: Evolution of the F-score for CBIC, CPC, G-CMIIC and B-CMIIC 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.

reference skeleton has been perfectly retrieved, the value of the F-score is 1. Figures 4 and 5 show the results for the ALARM networks and MCMC generated structures. As it



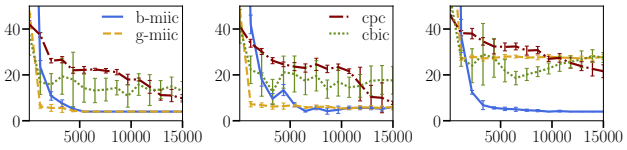
(a) Gaussian case (b) Student case (c) Dirichlet case

Figure 5: Evolution of the F-score for CBIC, CPC, G-CMIIC and B-CMIIC 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 $M = 10000$.

can be observed, G-CMIIC converges faster than the other algorithms but B-CMIIC and CPC converge approximately to the same value. Surprisingly, G-CMIIC conserves good results even for Dirichlet data. The CBIC method however is less performing compared to the three others whatever the generative model.

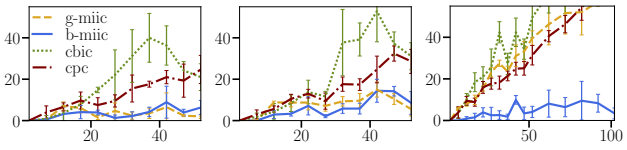
CPDAG Performances

In order to score the oriented structure, structural hamming distance (Colombo and Maathuis 2014) is used. This metric works on the completed partially directed acyclic graphs (CPDAG) that represents the Markov class equivalence of the DAG, that is all the graphs which represents the same set of independences(Koller and Friedman 2009). It counts



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

Figure 6: Evolution of the SHD for CBIC, CPC, G-CMIIC and B-CMIIC 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.



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

Figure 7: Evolution of the SHD for CBIC, CPC, G-CMIIC and B-CMIIC 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 $M = 10000$.

the number of elementary operations that are needed to recover the original structure from the estimated one. These operations consist of edge insertions, deletions and flipping. Figure 6 and 7 show the results for the ALARM network and random structures.

These results are similar to those obtained for the skeleton. The Gaussian G-CMIIC method recovers almost perfectly the CPDAG in the case of Gaussian and Student generative models and faster than the other techniques. However, its performances are quite low in the case of Dirichlet data. B-CMIIC on the other hand performs equally well whatever the generative model, illustrating the power of a non-parametric method. In the case of small structures, CPC seems to converge to the same value as the CMIIC methods but needs a lot more data. Its performances decrease when the dimension grows. As for CBIC method, its results are poor compared to the other techniques and in addition decrease for high dimensions

Time Complexity

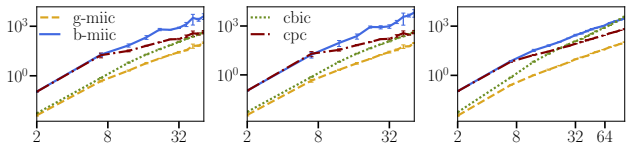
The learning times have been computed for the four methods as a function of the dimension of the random graphs and for data sets of size $M = 10000$. The results are shown on figure 8. The learning time of the B-MIIC algorithm is the most important despite its good structural performances. G-MIIC on the other hand is the fastest and as such, should be used when the Gaussian assumption is known to be valid. On the contrary, when no information is available, B-CMIIC should be used due to its better results on any distribution.

6 Conclusion and Future Works

The CBN model makes use of copula functions to parametrize the CPDs of a continuous BN. The BN representation on the other hand, limits the size of the copula functions which can be hard to manipulate for high dimensions. Furthermore, the CBN structure represents the same graphical independences than a classical BN. With some adaptations, this allows the use of the same learning techniques that are used for discrete data. In this regard, we proposed a continuous MIIC algorithm which lies between score-based and constraint-based methods. It required us to extend the link between mutual information and copula entropy to the conditional and three-point information. This extension led us to build non-parametric estimators of these quantities by use of the empirical Bernstein copula.

The experimental section illustrated the superiority of non-parametric methods over parametric ones when the model that generated the data is far from the estimated model. Moreover the lack of an arbitrary order allowed our algorithm to learn better results with less data than the CPC method. However, as it is often the case, the non-parametric methods are slower to learn and if information about the model is known, parametric methods should be preferred as illustrated by the Gaussian version of CMIC. All the source files to manage and learn CBNs with the CMIIC method can be find in the experimental plugin otagrum which is part of the OpenTURNS library and makes use of the aGrUM library.

While our results are very encouraging, the theoretical ground of the corrective parameter α is not satisfying. In place, a continuous score penalty could be used but discrete ones are either diverging in the continuous limit (NML, MDL) or only extendable for parametric models (BIC). A more promising idea would be to extend the estimator of mutual information introduced in (Belalia et al. 2017) to conditional and three-point information. This estimator being distributed according to a standard distribution in the limit of large samples, p-values could be used to quantify the confidence in independences. Finally, these results have been obtained through the use of generated data and have to be completed using real world data.



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

Figure 8: Learning time in seconds for CBIC, CPC, G-CMIIC and B-CMIIC 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 $M = 10000$.

Code Availability

The source code of our algorithms and tests are respectively available on the GitHub repositories openturns/otagrum and MLasserre/otagrum-experiments.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

Affeldt, S.; Verny, L.; and Isambert, H. 2016. 3off2: A network reconstruction algorithm based on 2-point and 3-point information statistics. In *BMC bioinformatics*, volume 17, S12. Springer.

Baudin, M.; Dutfoy, A.; Iooss, B.; and Popelin, A.-L. 2015. Open TURNS: An industrial software for uncertainty quantification in simulation. *arXiv preprint arXiv:1501.05242* .

Bedford, T.; Cooke, R. M.; et al. 2002. Vines–a new graphical model for dependent random variables. *The Annals of Statistics* 30(4): 1031–1068.

Beinlich, I. A.; Suermondt, H. J.; Chavez, R. M.; and Cooper, G. F. 1989. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89*, 247–256. Springer.

Belalia, M.; Bouezmarni, T.; Lemyre, F.; and Taamouti, A. 2017. Testing independence based on Bernstein empirical copula and copula density. *Journal of Nonparametric Statistics* 29(2): 346–380.

Colombo, D.; and Maathuis, M. H. 2014. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research* 15(1): 3741–3782.

Cover, T. M.; and Thomas, J. A. 2012. *Elements of information theory*. John Wiley & Sons.

Czado, C. 2010. Pair-copula constructions of multivariate copulas. In *Copula theory and its applications*, 93–109. Springer.

Elidan, G. 2010. Copula bayesian networks. In *Advances in neural information processing systems*, 559–567.

Genest, C.; and Favre, A.-C. 2007. Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of hydrologic engineering* 12(4): 347–368.

Glover, F.; and Laguna, M. 1998. *Tabu Search*, 2093–2229. Boston, MA: Springer US. ISBN 978-1-4613-0303-9. doi: 10.1007/978-1-4613-0303-9_33. URL https://doi.org/10.1007/978-1-4613-0303-9_33.

Gonzales, C.; Torti, L.; and Wuillemin, P.-H. 2017. 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.

Huang, J. C. 2009. *Cumulative distribution networks: Inference, estimation and applications of graphical models for cumulative distribution functions*. Citeseer.

Ide, J. S.; and Cozman, F. G. 2002. Random generation of Bayesian networks. In *Brazilian symposium on artificial intelligence*, 366–376. Springer.

Koller, D.; and Friedman, N. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

Langseth, H.; Nielsen, T. D.; Rumı, R.; and Salmerón, A. 2012. Mixtures of truncated basis functions. *International Journal of Approximate Reasoning* 53(2): 212–227.

Lasserre, M.; Lebrun, R.; and Wuillemin, P.-H. 2020. Constraint-Based Learning for Non-Parametric Continuous Bayesian Networks. In *FLAIRS 33 - 33rd Florida Artificial Intelligence Research Society Conference*, 581–586. Miami, United States: AAAI. URL https://hal.archives-ouvertes.fr/hal-02615379.

Lauritzen, S. L.; and Wermuth, N. 1989. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The annals of Statistics* 31–57.

Lindskog, F.; McNeil, A.; and Schmock, U. 2003. Kendall's tau for elliptical distributions. In *Credit Risk*, 149–156. Springer.

Ma, J.; and Sun, Z. 2011. Mutual information is copula entropy. *Tsinghua Science & Technology* 16(1): 51–54.

McGill, W. 1954. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory* 4(4): 93–111.

Neapolitan, R. E.; et al. 2004. *Learning bayesian networks*, volume 38. Pearson Prentice Hall Upper Saddle River, NJ.

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

Rissanen, J. 1978. Modeling by shortest data description. *Automatica* 14(5): 465–471.

Romero, V.; Rumí, R.; and Salmerón, A. 2006. Learning hybrid Bayesian networks using mixtures of truncated exponentials. *International Journal of Approximate Reasoning* 42(1-2): 54–68.

Sancetta, A.; and Satchell, S. 2004. The Bernstein copula and its applications to modeling and approximations of multivariate distributions. *Econometric Theory* 20(03): 535–562.

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

Shenoy, P. P.; and West, J. C. 2011. Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning* 52(5): 641–657.

Shtar'kov, Y. M. 1987. Universal sequential coding of single messages. *Problemy Peredachi Informatsii* 23(3): 3–17.

Sklar, A. 1959. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris* 8: 229–231.

Spirtes, P.; Glymour, C. N.; Scheines, R.; Heckerman, D.; Meek, C.; Cooper, G.; and Richardson, T. 2000. *Causation, prediction, and search*. MIT press.

Verny, L.; Sella, N.; Affeldt, S.; Singh, P. P.; and Isambert, H. 2017. Learning causal networks with latent variables from multivariate information in genomic data. *PLoS computational biology* 13(10): e1005662.