PC Algorithm for Gaussian Copula Data

L'algoritmo PC per dati generati da copula gaussiana

Vincenzina Vitale and Paola Vicard

Abstract The PC algorithm is the most popular algorithm used to infer the structure of a Bayesian network directly from data. For Gaussian distributions, it infers the network structure using conditional independence tests based on Pearson correlation coefficients. Here, we propose two modified versions of PC, the R-vine PC and D-vine PC algorithms, suitable for elliptical copula data. The correlation matrix is inferred by means of the estimated structure and parameters of a regular vine. Simulation results are provided, showing the very good performance of the proposed algorithms with respect to their main competitors.
Abstract *L'algoritmo PC è l'algoritmo piu diffuso per l'apprendimento della struttura di una rete bayesiana direttamente dai dati. Quando i dati sono gaussiani, esso apprende la struttura della rete per mezzo di test di indipendenza condizionata basati sui coefficienti di correlazione di Pearson. In questo lavoro, proponiamo due versioni modificate del PC, gli algoritmi R-vine PC e D-vine PC, validi per dati generati da copule ellittiche. La matrice di correlazione è calcolata sulla base della struttura e dei parametri stimati di un regular vine. Vengono forniti i risultati delle simulazioni che mostrano l'ottima performance degli algoritmi qui proposti rispetto ai loro principali competitor.*

Key words: Structural learning, Bayesian networks, R-vines, Gaussian copulae, PC algorithm.

Vincenzina Vitale
Dipartimento di Economia - Università Roma Tre, Via Silvio D'Amico, 77 - 00145 Roma, e-mail: vincenzina.vitale@uniroma3.it

Paola Vicard
Dipartimento di Economia - Università Roma Tre, Via Silvio D'Amico, 77 - 00145 Roma, e-mail: paola.vicard@uniroma3.it

1 Introduction

A Bayesian network (BN,[6]) is a multivariate statistical model satisfying sets of (conditional) independence statements encoded in a *Directed Acyclic graph* (DAG). Each node in the graph represents a random variable while the edges between the nodes represent probabilistic dependencies among the corresponding variables. If there is an arrow from X_i to X_j, X_j is said *child* of X_i and X_i is said *parent* of X_j. The set of parents of X_j in the graph G is denoted by $pa(X_j)$. In a BN each node is associated with a conditional distribution given its parents and the joint distribution can be factorized according to the DAG structure as:

$$p(X_1 \dots X_p) = \prod_{j=1}^{p} p(X_j | pa(X_j)) \tag{1}$$

BN structure can be elicited by the expert knowledge or learnt directly from data by means of structural learning techniques [16]. Among the *constraint-based* algorithms, the most known is the PC algorithm [18]: it estimates the Markov equivalence class of a DAG performing three main steps: i) *skeleton*[1] *identification* by recursively testing marginal and conditional independencies using Pearson correlation coefficients, for a fixed significance level α; ii) v-*structures identification*: an unshielded triple $i-k-j$, such that the pairs i and k and j and k are connected while i and j is not, is oriented as $i \to k \leftarrow j$ if k is not in the separation sets of nodes X_i and X_j; iii) *orientation of some of the remaining edges* without producing additional v-structures and/or directed cycles.

In the BN framework, when the analysed variables are continuous, joint normality[2] is assumed. Unfortunately, in many applied context the normality assumption may not be reasonable. In such cases, copula modeling has become very popular and, recently, there is a growing literature where the theory of copulae and Bayesian networks are combined [10, 14, 8, 15, 11, 2]. Here, a modified version of the PC algorithm suitable for Gaussian and Student t copula distributions is proposed. In particular, we work in the theoretical framework of pair-copula constructions with reference to the subclass of regular vines [4, 12]. From the estimated structure and parameters of a regular vine, we infer the corresponding marginal correlation matrix valid under the assumption that data are drawn from a Gaussian copula family. The correlation coefficients are then used as sufficient statistics in the conditional independence tests implemented in the PC algorithm. Simulations are carried out in order to evaluate the performance of the proposed algorithms and to compare them with the PC and the Rank PC (RPC) algorithms. RPC has been recently introduced by [15] to overcome the normality assumption limitation and can be used for Gaussian copula data.

---

[1] The skeleton of a DAG is the undirected graph obtained replacing arrows with undirected edges.

[2] In the mixed continous and discrete case, the conditional Gaussian distribution is assumed.

The paper is organized as follows. In Section 2 pair copula construction and regular vines are introduced; in Section 3 the new algorithms are illustrated and simulation results are shown and discussed.

2 Pair copula construction and regular vines

Let F be a n-dimensional distribution function of the random vector $\mathbf{X} = (X_1, \cdots X_n)$ with univariate marginals $F_1 \dots F_n$. A *$n-$variate copula* is a multivariate cumulative distribution function (cdf) $C : [0,1]^n \rightarrow [0,1]$ with $n \in N$ and uniformly distributed marginals $U(0,1)$ on the interval $[0,1]$. By Sklar theorem [17], every cdf F with marginals $F_1 \dots F_n$ can be written as:

$$F(\mathbf{x}) = C(F_1(x_1), \dots, F_n(x_n)) \tag{2}$$

for some appropriate n-dimensional copula C. By copulas, multivariate distribution modeling is split into univariate marginals and dependence structure modeling. While for bidimensional case there is an exaustive literature on bivariate copula families, their extension to multivariate case is not straightforward[3]. In [3, 4] and [14] the decomposition of the multivariate copulae into the product of bivariate ones, known in literature as *pair-copula construction* (PCC), is proposed. Each pair-copula can be selected independently from the others allowing for a great flexibility in dependence modeling. Since in higher dimensions the number of possible pair-copulae constructions grows up significantly, in [3, 4] a graphical representation (called *regular vine*) to organize them, is introduced.

Generally speaking, a regular vine (R-vine) is a sequence of trees whose edges correspond to bivariate copulae; see [14] for a formal definition. A n-dimensional R-vine is a set of $n-1$ trees such that the first tree comprises n nodes, identifying $n-1$ pairs of variables and $n-1$ corresponding edges. Each subsequent tree is derived so that all the edges of tree i turn into nodes of the tree $i+1$; furthermore, two edges in T_i, becoming nodes in T_{i+1}, are joined by an edge in T_{i+1} only if these edges share a common node in T_i. The graphical structures of R-vines allow the specification of all bivariate copulae of the pair copula construction. In particular, each edge corresponds to a bivariate copula density. The copulae defined in the first tree are unconditional copulae while the others are all conditional [4]. The importance of these results arises from the fact that all bivariate copulae can belong to different families and their parameters can be specified independently from each other. Many applications concern a special case of R-vines, the Drawable vines (D-vines), see [1]. D-vines only need the ordering definition of their first tree sequence to completely

---

[3] Standard multivariate copulae such as Gaussian or Student t lack the flexibility of accurately modeling the dependence structure in higher dimensions.

[4] The copulae of the second tree have only one node as conditioning set, the third two nodes and so on.

identify the structure. Differently, R-vines suffer from the fact that many possible tree sequences can be specified.

Three separate steps have to be done to specify the vine structure and distribution:

1. selecting the structure with all its trees;
2. selecting the appropriate bivariate copula family for each of the $n(n-1)/2$ pair copulae associated with the vine structure;
3. estimating the parameters for each bivariate copula identified at the previous step.

Since the number of possible R-vines on n variables increases exponentially with n, a sequential method has been proposed by [7] and implemented in the `VineCopula R` package. Among the possible copula types, the independence copula can also be chosen by means of a preliminary independence test based on Kendall's tau [9]. The concept of independence copula is strictly connected to that of conditional independence: it allows to reduce the number of parameters to be estimated. In [5] the truncated R-vines, for which independence is assumed for the k last trees, is proposed. More recently, [11] have deeply analysed the truncation procedure showing the relationship between the truncated R-vines and the decomposable graphs.

3 The R-vine and D-vine PC algorithms: simulations and results

Here, we take advantage of the use and properties of regular vines to estimate the BN dependence structure under the assumption of data coming from a Gaussian copula distribution. More precisely, we infer the R-vine (and D-vine) structure together with its copula parameters in order to extract the corresponding marginal correlation matrix. Note that the copula family is limited to Gaussian or Student t case for which the correlation coefficients can be estimated. The last step consists in using the marginal correlation coefficients as sufficient statistics for Pearson correlation tests implemented in the classical version of the PC algorithm. According to these definitions and purposes, we propose four algorithms: the R-vine PC algorithm, the D-vine PC algorithm and their truncated versions respectively. They work along the following four steps:

1. transforming data in pseudo-observations;
2. fitting a R-vine (or D-vine) to the transformed data based on the AIC criterion and the maximum likelihood estimation of copula parameters;
3. inferring the marginal correlation matrix from the estimated R-vine (or D-vine);
4. running the PC algorithm providing, in input, the estimated marginal correlation matrix of the previous step.

All functions used in the first three steps are implemented in the `VineCopula R` package. Regarding the D-vine, the `TSP R` package has been used to determine the order of the nodes of its first tree. The fourth step functions are implemented in the `pcalg R` package.

We argue that the two non truncated algorithms allow the specification of the independence copula as proposed by [9][5]. The procedure of truncation applied in this work follows the approach of [5]. To choose the optimal truncation level, a R function has been written in order to recursively perform the likelihood ratio based test between different levels.

Two random DAGs, one not decomposable and the other decomposable, with sparsity parameter $s = 0.4$ and $s = 0.3$ respectively, are simulated according to the procedure ensuring faithfulness [13]. 250 datasets are drawn from each DAG following a Gaussian copula distribution, fixing $n = 500$ and $\alpha = 0.01$. The *structural Hamming distance* (SHD, [19]) has been computed in order to evaluate the different performances of the algorithms.

Table 1 Simulation results by algorithms

Graph (n=500)	Algorithm	SHD (mean)	SHD(Median)	SHD (s.d.)	SHD (IQR)
Decomposable graph	PC	8,29	8	1,62	2
	RPC	6,43	6	1,14	1
	R-vine PC	6,14	6	1,24	2
	Truncated R-vine PC	6,38	6	1,05	1
	D-vine PC	6,23	6	1,21	2
	Truncated D-vine PC	6,56	6	1,34	1
Non decomposable graph	PC	5,87	6	2,02	2
	RPC	3,69	3	2,35	4
	R-vine PC	2,04	1	1,66	2
	Truncated R-vine PC	2,36	2	1,71	2
	D-vine PC	2,88	3	1,92	3
	Truncated D-vine PC	4,07	4	2,51	4

The simulation results, shown in Tab. 1, are very promising. As expected, under the assumption of Gaussian copula data, the performance of the PC algorithm, in terms of mean value, is worse than all the others. For a non decomposable graph, the performance of R-vine PC algorithm, followed by that of its truncated version is extremely good. The mean value of the errors is the smallest, about 2; if its median value is taken into account, it is equal to 1. As far as variability is concerned, its standard deviation is also very small. With the exception of the truncated D-vine PC algorithm, the proposed algorithms outperform their competitors.

For data generated from a decomposable graph, the differences in performance with respect to the PC algorithm are still evident. The distance between our proposals and the RPC algorithm is less remarkable. The R-vine and D-vine PC algorithms show the smallest mean values but larger variability. The truncated R-vine PC algorithm seems to balance these two aspects. Simulation results clearly show that the undirected graph R-vine is able to capture the underlying dependence structure of data. It considerably increases the capability of the PC to detect the best fitting

---

[5] The hypothesis test for the independence of pseudo-observations u_1 and u_2 is performed before bivariate copula selection. The independence copula is chosen for a (conditional) pair if the null hypothesis of independence cannot be rejected.

network. The main limitation of the proposed algorithms is that they are applicable only to elliptical copula distributions, restricting the choice of possible copula families. Future research will necessarily concern the definition of a new class of algorithms suitable for non normal data without any restriction to the class of copula families.

References

1. Aas, K., Czado, C., Frigessi, A., Bakken, H.: Pair-copula constructions of multiple dependence. Insurance: Mathematics and economics **44**(2), 182–198 (2009)
2. Bauer, A., Czado, C.: Pair-copula bayesian networks. J. Comput. Graph. Stat. **25**(4), 1248–1271 (2016)
3. Bedford, T., Cooke, R.M.: Probability density decomposition for conditionally dependent random variables modeled by vines. Ann. Math. Artif. Intell. **32**(1), 245–268 (2001)
4. Bedford, T., Cooke, R.M.: Vines: A new graphical model for dependent random variables. Ann. Stat. **30**(4), 1031–1068 (2002)
5. Brechmann, E.C., Czado, C., Aas, K.: Truncated regular vines in high dimensions with application to financial data. Can. J. Stat. **40**(1), 68–85 (2012)
6. Cowell, R.G., Dawid, P., Lauritzen, S.L., Spiegelhalter, D.J.: Probabilistic Networks and Expert Systems. Springer-Verlag, New York. (1999)
7. Dissmann, J., Brechmann, E.C., Czado, C., Kurowicka, D.: Selecting and estimating regular vine copulae and application to financial returns. Comput. Stat. Data. Anal. **59**, 52–69 (2013)
8. Elidan, G.: Copula bayesian networks. In: Advances in neural information processing systems, pp. 559–567 (2010)
9. Genest, C., Favre, A.: Everything you always wanted to know about copula modeling but were afraid to ask. J. Hydrol. Eng. **12**(4), 347–368 (2007)
10. Hanea, A.M., Kurowicka, D., Cooke, R.M.: Hybrid method for quantifying and analyzing bayesian belief nets. Qual. Reliab. Eng. Int. **22**(6), 709–729. (2006)
11. Hobæk Haff, I., Aas, K., Frigessi, A., Lacal, V.: Structure learning in bayesian networks using regular vines. Comput. Stat. Data Anal. **101**(C), 186–208 (2016)
12. Joe, H.: Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters. Lecture Notes-Monograph Series pp. 120–141 (1996)
13. Kalisch, M., Bühlmann, P.: Estimating high-dimensional directed acyclic graphs with the pc-algorithm. J. Mach. Learn. Res. **8**, 613–636. (2007)
14. Kurowicka, D., Cooke, R.: Uncertainty Analysis with High Dimensional Dependence Modelling. John Wiley & Sons, Ltd. (2006)
15. Naftali, H., Drton, M.: Pc algorithm for nonparanormal graphical models. J. Mach. Learn. Res. **14**, 3365–3383. (2013)
16. Neapolitan, R.E.: Learning Bayesian Networks. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. (2003)
17. Sklar, A.: Fonctions de répartition ǹ dimensions et leurs marges. Publications de l'Institut de Statistique de L'Université de Paris **8**, 229–231. (1959)
18. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, 2nd edn. MIT press, Cambridge, Massachusetts. (2000)
19. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. Mach. Learn. **65**(1), 31–78 (2006)