The Relationship Between Graphical Representations of Regular Vine Copulas and Polytrees

Diana Carrera[1(✉)], Roberto Santana[1], and Jose A. Lozano[1,2]

[1] Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Manuel de Lardizabal, 1, 20018 Donostia, Gipuzkoa, Spain
dianamaria.carrera@ehu.es

[2] Basque Center for Applied Mathematics, BCAM, 48009 Bilbao, Spain

Abstract. Graphical models (GMs) are powerful statistical tools for modeling the (in)dependencies among random variables. In this paper, we focus on two different types of graphical models: R-vines and polytrees. Regarding the graphical representation of these models, the former uses a sequence of undirected trees with edges representing pairwise dependencies, whereas the latter uses a directed graph without cycles to encode independence relationships among the variables. The research problem we deal with is whether it is possible to build an R-vine that represents the largest number of independencies found in a polytree and vice versa. Two algorithms are proposed to solve this problem. One algorithm is used to induce an R-vine that represents in each tree the largest number of graphical independencies existing in a polytree. The other one builds a polytree that represents all the independencies found in the R-vine. Through simple examples, both procedures are illustrated.

Keywords: Regular vine copulas · Polytrees
(In)dependence relationships · Graphical models

1 Introduction

Graphical models (GMs) [7,13] have been widely used for modeling the dependence structure of multivariate probability distributions through two closely related components: (i) The qualitative component is a graph where nodes correspond to random variables and edges to graphical relationships among them; (ii) The quantitative component is given by a set of local probability distributions that quantify the strength and uncertainty of the (in)dependencies encoded in the graph (or network). According to the type of the graph, directed and undirected, we can distinguish two different GMs: Bayesian networks (BNs) and Markov networks respectively. The interpretation of graphical (in)dependencies is different in directed and undirected graphs.

J. Medina et al. (Eds.): IPMU 2018, CCIS 855, pp. 678–690, 2018.
https://doi.org/10.1007/978-3-319-91479-4_56

In this paper, we focus on two GMs representative of undirected and directed graphs, namely regular vine copulas [4,12] (or simply R-vines) and a subclass of BNs called polytrees [8] respectively.

A copula is a probability distribution function with uniformly distributed margins [14,17]. Copulas allow us to model the dependence structure of multivariate distributions and its margins separately. Despite the generality of the copula-based framework, it turns out that building high-dimensional joint copulas is a difficult problem [1].

Pair copula constructions (PCCs) [11] and their graphical model, called regular vines (R-vines) [3,4], overcome the lack of flexibility of the copula-based modeling in the high-dimensional case. R-vines build multivariate copulas in terms of bivariate copulas (pair-copulas) taking advantage of the fact that the bivariate copulas are more tractable than multidimensional ones. Besides that, bivariate copulas of different families, can be combined in the same decomposition allowing the specification of different types of non-linear dependencies. The qualitative component of R-vines is specified by an R-vine structure (or graph) – a set of nested trees, where the variables are represented by nodes linked by edges, each associated with a pair-copula that captures certain types of pairwise dependence. It is in this sense that we say that R-vine structures encode dependence relationships rather than independencies relationships.

Polytrees (also known as singly connected networks) are directed acyclic graphs (DAGs) where there is no more than one undirected path that connects any two nodes (without undirected cycles). In these graphs, missing edges can represent either conditional independencies or conditional dependencies among random variables.

In general, Bayesian networks, particularly polytrees, have well-studied mathematical properties that have been developed throughout decades. In contrast, R-vines have boomed in the last few years. Previous works have addressed the question of the relationship between directed GMs and R-vines from different perspectives. In [9], a new method for learning the structure of a BN based on PCCs is introduced. In [10], a non-parametric Bayesian belief net as an alternative to a particular subclass of R-vines is introduced. The paper discusses the differences between both models and offers some guidelines on when to use one or the other from a quantitative perspective. In [2], a Bayesian network with pair-copulas is built using PCCs.

However, the problem of verifying whether the graphical independencies found in a polytree can be represented in an R-vine and vice versa has not been answered yet. In this work, we investigate the relationship between the graphical representations of R-vines and polytrees in both directions: (i) Given the graph of a polytree, we want to obtain an R-vine tree-structure that represents the largest number of independencies existing in the starting graph. To this end, a heuristic is proposed that, from the list of independences found in the polytree, performs this task locally, tree-by-tree of the R-vine. (ii) Given an R-vine, we want to build a polytree that represents the largest number of independences existing in the R-vine. Similarly, a heuristic is proposed that, based on the

independence list extracted from the R-vine, builds a polytree that represents the independencies existing in the R-vine. These results are useful as they make it clear that properties and algorithms that can be applied to polytrees can be carried over to R-vines, and vice versa.

The paper is organized as follows: In Sects. 2 and 3, we provide the basic concepts as well as a short review of R-vines and polytrees respectively. In Sect. 4, we present the main contribution of this work: two algorithms that induce the graph of a R-vine from the graph of a polytree and vice versa. Section 5 offers a short summary and an outline of future work.

2 Regular Vines

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be an n-dimensional random vector with joint density function $f : \mathbb{R}^n \to [0, \infty)$ and cumulative distribution function $F : \mathbb{R}^n \to [0, 1]$. Furthermore, let $F_i : \mathbb{R} \to [0, 1]$, $i = 1, \ldots, n$ be the corresponding marginal distributions of X_i[1]. Capital letters denote variables and lower letters are their assignments.

A n-dimensional copula C is a multivariate probability distribution function for which the univariate margins are uniform: $C : [0, 1]^n \to [0, 1]$ [14]. Copulas are used to describe the dependence structure among random variables.

The relevance of copulas in probabilistic modeling is given by Sklar's theorem [17], which states that an n-dimensional (multivariate) distribution function F of a random continuous vector $\mathbf{X} = (X_1, \ldots, X_n) \in \mathbb{R}^n$ can be expressed in terms of its marginal distributions $F_i(x_i)$ and a unique copula C. Sklar's theorem for densities is given by

$$f(x_1, \ldots, x_n) = c(F_1(x_1), \ldots, F_n(x_n)) \cdot \prod_{i=1}^{n} f_i(x_i) \tag{1}$$

where f and c denote the density functions corresponding to F and C respectively.

In (1), the copula c can be approximated by an PCC. This decomposition is represented graphically by an R-vine – a sequence of trees, of which each edge corresponds to a pair-copula. An R-vine is a probabilistic graphical model represented as a pair (G, θ). G is the structural part that is composed of a sequence of trees $T_1, T_2, \ldots, T_{n-1}$, where the nodes of T_j are edges in T_{j-1}. Two nodes in T_j (for $j \geq 2$) can only be adjacent if the corresponding edges in the previous tree have a common node (known as proximity condition) [2]. θ contains, for each edge of the trees, a pair-copula and its parameters. The number of edges in an R-vine is $n(n-1)/2$. Figure 1-(right panel) illustrates an R-vine copula c_{12345} for $n = 5$ and its respective factorization.

We define an R-vine formally by following the definition given in [5]: If we denote $T_j = (N_j, E_j)$, the tree of the decomposition at level j, where N_j and E_j

[1] We assume that all multivariate, marginal and conditional distributions are absolutely continuous with corresponding densities.

denote the node and edge sets of the j^{th} tree, the edge $e \in E_j$ joins two vertices of N_j, $X_{k(e)}$ and $X_{l(e)}$, which are determined by the set of indices $k(e)$ and $l(e)$ respectively. Then, in the pair-copula $c_{k(e),l(e)|D(e)}$, the nodes $X_{k(e)}$ and $X_{l(e)}$ are the conditioned nodes, whereas $\mathbf{X}_{D(e)}$, which represents a subvector of $\mathbf{X}$ determined by the indices in $D(e)$, is the conditioning set. Consequently, a regular vine distribution is the distribution of the random vector $\mathbf{X} = (X_1, \ldots, X_n)$ with marginal densities $f_i(x_i)$, $i = 1, \ldots, n$, and where the conditional density of $\left(X_{k(e)}, X_{l(e)}\right)$ given $\mathbf{X}_{D(e)}$ is specified as $c_{k(e),l(e)|D(e)}$ for the R-vine copula with $n-1$ trees, set of nodes $\mathbf{N} = \{N_1, \ldots N_{n-1}\}$ and set of edges $\mathbf{E} = \{E_1, \ldots E_{n-1}\}$. If the dependence structure of $\mathbf{X}$ is represented by an R-vine copula, then the n-dimensional density $f_{\mathbf{R-vine}}(x_1, \ldots, x_n)$ is given by

$$\underbrace{\prod_{j=1}^{n-1} \prod_{e \in E_j} c_{k(e),l(e)|D(e)} \left(F\left(x_{k(e)} \mid \mathbf{x}_{D(e)}\right), F\left(x_{l(e)} \mid \mathbf{x}_{D(e)}\right)\right)}_{\text{R-vine copula}} \cdot \underbrace{\prod_{i=1}^{n} f_i(x_i)}_{\text{Margins}} \tag{2}$$

In the approximation given in (2), only in the first tree are the pair-copulas unconditional as their arguments are marginal distributions. In the remaining trees, the pair-copulas are conditional as their arguments are conditional distributions. The number of variables in the conditioning set increases in one variable as we go deeper into the R-vine tree-structure: in the second tree, we have first-order conditional copulas; in the third tree, second-order conditional copulas, and so on; that is, in the tree j we have conditional copulas of the order $j-1$.

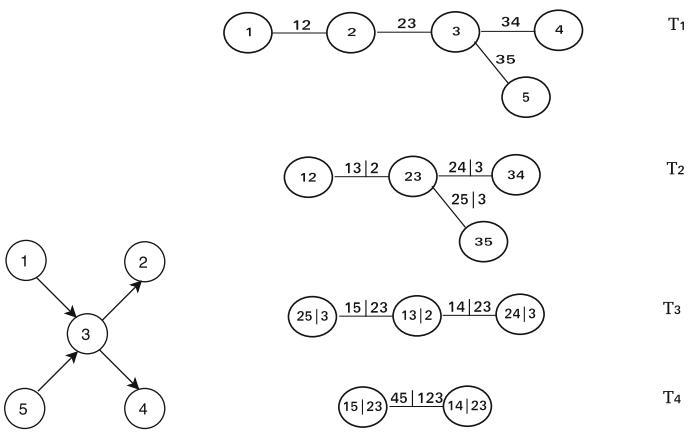

Fig. 1. Example of a polytree (left panel) and an R-vine (right panel) where $n = 5$. The polytree factorization is given as $p(x_1) \cdot p(x_5) \cdot p(x_3 \mid x_1, x_5) \cdot p(x_2 \mid x_3) \cdot p(x_4 \mid x_3)$. The R-vine factorization is given as $\underbrace{c_{12} \cdot c_{23} \cdot c_{34} \cdot c_{45}}_{T_1} \cdot \underbrace{c_{13|2} \cdot c_{24|3} \cdot c_{25|3}}_{T_2} \cdot \underbrace{c_{15|23} \cdot c_{14|23}}_{T_3} \cdot \underbrace{c_{45|123}}_{T_4} \cdot$

2.1 Graphical (In)Dependence in R-vines

We define two types of edges (or links): dashed edges indicate linked nodes are independent and continuous edges indicate linked nodes are dependent with each other.

To provide the R-vine tree-structure with a semantic interpretation in terms of independencies, we define the vine-graphical independence criterion as follows.

Definition 1 (r-separation). *Let $X_{k(e)}$, $X_{l(e)}$, $\mathbf{X}_{D(e)}$ three disjoint subsets in and R-vine. We say that $\mathbf{X}_{D(e)}$ r-separates $X_{k(e)}$ from $X_{l(e)}$ if there is a dashed edge in some tree that joins two vertices, on of them is associated with the vertices in $X_{k(e)}$ the other to the vertices in $X_{l(e)}$, and $\mathbf{X}_{D(e)}$ r-separates nodes $X_{k(e)}$ and $X_{l(e)}$.*

When $\mathbf{X}_{D(e)}$ v-separates $X_{k(e)}$ and $X_{l(e)}$ in G, we write $I\left(X_{k(e)}, X_{l(e)} \mid X_{D(e)}\right)_G$ to indicate that this graphical conditional independence relationship is represented in the graph G. We write $D\left(X_{k(e)}, X_{l(e)} \mid X_{D(e)}\right)_G$ to indicate that $X_{k(e)}$ and $X_{l(e)}$ are conditionally dependent given $\mathbf{X}_{D(e)}$ in the graph G.

3 Polytrees

3.1 Directed Graphs

Let $G = (\mathbf{X}, \mathbf{E})$ be a directed acyclic graph (DAG), consisting of the node set $\mathbf{X}$ and the edge set $\mathbf{E}$. Directed graphs only contain directed edges. A directed edge from node X to node Y is represented as $X \rightarrow Y$. A path from $X = X_1$ to $Y = X_d$ is a sequence of nodes $X_1, \ldots, X_d$ connected by edges in the graph G, where the edge $(X_i, X_{i+1}) \in \mathbf{E}$, $i = 1, \ldots, d-1$. A cycle is a path where $X = Y$. We say that X is an ancestor of Y if we can find a path that, starting from X, reaches the node Y, such that $X \rightarrow ... \rightarrow Y$; correspondingly, Y is a descendant of X. An undirected path is a path in which the directions of the edges are not considered. The skeleton of G is the undirected graph obtained by eliminating the directions of edges from G. A head-to-head (h-h) connection is a subgraph $X \rightarrow Z \leftarrow Y$ in which Z is a h-h node (i.e., a node with convergent edges). A comprehensive introduction to graph theory and graphical models is found in [6].

3.2 Graphical (In)Dependence in DAGs

The concept of d-separation [15] is the graphical independence criterion that provides the DAG a semantic interpretation, allowing it to determine the independence relationships encoded by the topology of the network.

Definition 2 (d-separation). *If $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are three disjoint subsets of nodes in a DAG G, then $\mathbf{Z}$ d-separates $\mathbf{X}$ from $\mathbf{Y}$ or, similarly, $\mathbf{X}$ and $\mathbf{Y}$ are graphically independent given $\mathbf{Z}$ if and only if along any undirected path between any node of $\mathbf{X}$ and any node of $\mathbf{Y}$ there is an intermediate node A such that (i) either$A$ is a head-to-head node in the path and neither A nor its descendants are in $\mathbf{Z}$, or (ii) A is not a head-to-head node in the path and it is in $\mathbf{Z}$.*

When $\mathbf{Z}$ d-separates $\mathbf{X}$ and $\mathbf{Y}$ in G, we write $I(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_G$ to indicate that the independence relationship is given by the graph G. We write $D(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_G$ to indicate that $\mathbf{X}$ and $\mathbf{Y}$ are conditionally dependent given $\mathbf{Z}$ in the graph G.

3.3 Dependence Models

Necessary definitions on dependence models are taken from [6]. The terms of the dependence and the independence models refer exclusively to the qualitative structure of the relationships existing in a set of variables. These models allow us to check which sets of variables are unconditionally or conditionally dependent or independent.

Definition 3 (Dependence Model). *A model M of the variable set $\{X_1, \ldots, X_n\}$ is called a dependence model if it allows to determine whether $I(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_M$ is true for all the possible triples of subsets $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$.*

Two possible correspondences between a graphical representation G and a dependence model M are I-map and D-map.

Definition 4 (I-map). *The graph G is an I-map of the dependence model M if $I(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_G \Rightarrow I(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_M$, i.e., if all independence relationships derived from G are verified in M.*

Definition 5 (D-map). *The graph G is a D-map of the dependence model M if $D(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_G \Rightarrow D(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z})_M$, i.e., if all dependence relationships derived from G are verified in M.*

An I-map G of M includes some of the independence relationships of M, but not necessarily all of them. An I-map guarantees that the d-separated nodes correspond to independent variables in M, but does not guarantee that connected nodes correspond to dependent variables in M. On the other hand, a D-map G of M includes some of the dependence relationships of M, but not necessarily all of them. A D-map guarantees that connected nodes correspond to dependent variables in M, but does not guarantee that the d-separated nodes correspond to independent variables in M. Empty graphs (the set of edges is empty) and complete graphs (there is an edge between each pair of nodes) are called trivial D-maps and I-maps respectively.

3.4 Polytrees

Bayesian networks (BNs) are GMs based on DAGs. A BN is a pair $(G(\mathbf{X}, \mathbf{E}), P)$, where G is a DAG, $\mathbf{X}$ and $\mathbf{E}$ are the set of variables (or nodes) and the set of directed edges in G respectively, and $P = \{P(X_1 \mid \mathbf{Pa}_1), \cdots, P(X_n \mid \mathbf{Pa}_n)\}$ is a set of n conditional probability distribution functions (one for each variable) where $\mathbf{Pa}_i$ is the set of parents of X_i in G. The set P defines a probability function given by

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i \mid \mathbf{Pa}_i) \tag{3}$$

A particular subclass of BNs are polytrees (of course, (3) also applies to polytrees). In these networks, there is no more than one undirected path that connects any two nodes. Particular types of polytrees include chains: each node has at most one parent and/or only one child, and trees: each node has only one parent. The number of edges in a polytree is $n-1$. Figure 1-(left panel) shows a polytree where $n=5$ and its respective factorization.

4 The Graphical Relationship Between R-vines and Polytrees

This section proposes two methods that induce the graph of an R-vine from the graph of the polytree and vice versa, so that the resulting graph represents the largest number of independencies existing in the other graph.

4.1 From Polytrees to R-vines

We want to obtain an R-vine G_{R-vine} that represents the largest number of independencies found in the polytree G_P. For this purpose, we propose Algorithm 1. To simplify the notation used in Definition 1, we use $X = X_{k(e)}$, $Y = X_{l(e)}$, and $\mathbf{V} = \mathbf{X}_{D(e)}$. Moreover, $|\mathbf{V}| = j-1$ denotes the cardinality of $\mathbf{V}$, and conditioning sets with cardinality $|\mathbf{V}|$ belong to the tree at the level $j = |\mathbf{V}| + 1$.

We consider a dependence model M that contains the list of independencies and dependencies represented in the polytree G_P, L_I and L_D respectively. These lists are obtained via the d-separation criterion in Step 1. The elements of these lists have the form $I(X, Y \mid \mathbf{V})$ and $D(X, Y \mid \mathbf{V})$ respectively. Both lists are arranged in ascending order according to $|\mathbf{V}|$.

In Step 2, we obtain the first tree of G_{R-vine}, which is nothing more than the skeleton of the polytree. Notice that T_1 and the skeleton of the polytree have the same edge set, so that both structures represents the same set of unconditional dependencies.

In Steps 3 and 4, the next trees of G_{R-vine} are built inside a *for-loop* that runs over the levels $j = 2, \ldots, n-1$. These trees are maximum weighted spanning trees (MWSTs) [16] that satisfy the R-vine properties. As the edge's weight we use zero for continuous edges and one for dashed edges respectively. Afterward, each relation in L_D of order $j-1$ suggests an edge, which is inserted in the graph if the following two conditions are satisfied: (i) the edge to be inserted does not introduce undirected cycles, which ensures that the resulting R-vine remains singly connected; (ii) the proximity condition holds. If both conditions are met, the boolean function $\varphi(T_j, D_i(X, Y \mid \mathbf{V}))$ is true.

Note that not all independencies found in the polytree can be represented in the R-vine, but those inserted exist in the polytree. On the other hand, we have that all the independence relationships represented in the R-vine exist in the polytree and we denote it as $I(G_{\text{R-vine}}) \subseteq I(G_{\text{P}})$, thus the R-vine is an I-map of the dependence model M obtained from the polytree.

It is worth noting that if the R-vine is a D-vine – a subclass of R-vines where the trees have chain structure – we do not need to select MWSTs since the first tree determines completely the structure of the next trees [1]).

Example 1. Let us illustrate Algorithm 1 based on the polytree of Fig. 2-(left-panel).

Step 1. Obtain $M = \{L_I, L_D\}$ from G_P.

$$L_I = \{I_1(2,4 \mid 3), I_2(1,4 \mid 3), I_3(1,3 \mid 4), I_4(1,4 \mid 2,3)\}$$
$$L_D = \{D_1(1,3 \mid 2), D_2(1,4 \mid 2), D_3(1,2 \mid 3,4), D_4(1,3 \mid 2,4), D_5(2,3 \mid 1,4)\}$$

Step 2. From T_1 (Fig. 2-(right panel)) we obtain the connected nodes: $1-2$, $2-3$, $3-4$.

Step 3. As T_1 is a chain (as in D-vines) we do not need to built an MWST at each level, but only determine if the edges are dashed or continuous. Next we pass to T_2, and from L_I we see that $I_1(2,4 \mid 3)$ may be represented with the dashed edge 23-··-34. However, the dashed edges 13-··-34 and 14-··-34 corresponding to the relationships $I_2(1,4 \mid 3)$ and $I_3(1,3 \mid 4)$, respectively, cannot be inserted in T_2 since the nodes 13 and 14 do not belong to the node set of this tree, which are: 12, 23, and 34.

Step 4. $D_1(1,3 \mid 2)$ can be represented by the continuous edge $12-23$ in T_2 without violating the graphical constraints that should hold an R-vine, which implies that $\varphi(T_2, D_1(1,3 \mid 2))$ is true.

Step 5. To build T_3, only I_4 has to be inserted. This can be done by means of the dashed edge 14 | 2-··-24 | 3.

Notice that the independence relationships represented in this R-vine structure, namely: $[I_1(2,4 \mid 3), I_2(1,4 \mid 2,3)]$, exist in G_{P}, thus this R-vine is an I-map of the dependence model obtained from the polytree. This way of building the structure of R-vines guarantees that only graphical independencies found in the polytree are inserted in the corresponding R-vine tree. Consequently, all the independencies represented in the R-vine are true in the polytree.

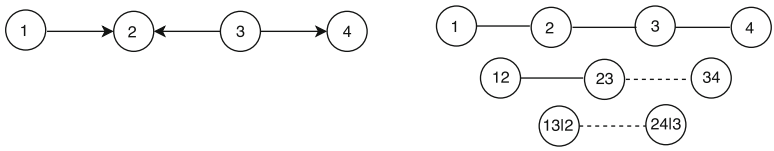

Fig. 2. Illustration of Example 1: (left panel) polytree G_P; (right panel) the resulting R-vine G_{R-vine}.

Algorithm 1. Procedure to build the R-vine tree-structure from the graph of a polytree.

Input: G_P
Output: G_{R-vine}
Step 1 Create the lists L_I and L_D from G_P.
Step 2 Obtain T_1 = skeleton of G_P.
for $j = 2, \ldots, n-1$:
Step 3:
if G_{R-vine} = D-vine:
for each i in $I_i\,(X, Y \mid V)$ of order $j-1$ in L_I
if $\varphi\,(T_j, I_i\,(X, Y \mid \mathbf{V}))$:
Add a dashed edge X-··-Y in T_j.
else
Build the MWST T_j with dashed edges only.
Step 4:
for i, $D_i\,(X, Y \mid \mathbf{V})$, in L_D:
if $\varphi\,(T_j, D_i\,(X, Y \mid \mathbf{V}))$:
Add a continuous edge $X - Y$ in T_j.

4.2 From R-vines to Polytrees

Similarly to previous section, the idea of this section is to build a polytree G_P that represents the largest numbers of independencies found in an R-vine G_{R-vine}. The heuristic proposed is shown in Algorithm 2.

We consider a denpendence model M that contains the list of independencies and dependencies represented in the R-vine G_{R-vine}, L_I and L_D respectively. Step 1 consists of extracting both lists: L_I is represented by dashed lines and L_D is represented by continuous lines via the v-separation criterion. In Step 2, we obtain the skeleton of G_P that is no other than tree T_1 of G_{R-vine}. To extract L_I and L_D, we can use the procedure given in [6]

Steps 3 and 4 are responsible for determining the direction of the edges of the polytree skeleton. Firstly, the algorithm goes through the list L_I in order to insert the independence relationships. In Step 3, for each $I\,(X, Y \mid \mathbf{V})$ the edges of the corresponding subgraph $X - \mathbf{V} - Y$ are oriented preventing any node of $\mathbf{V}$ from being a h-h. In Step 4, the algorithm goes through the list L_D in order to insert those dependencies that do not eliminate any independencies previously represented. So, for each $D\,(X, Y \mid \mathbf{V})$ the algorithm allows a node belonging to $\mathbf{V}$ to be a h-h if the independencies already inserted are still represented; otherwise, edges are not oriented towards any node of $\mathbf{V}$.

Notice that a complete R-vine structure (all its edges are continuous) is a trivial I-map of the dependence model obtained from the polytree. In the opposite case, if all the R-vine's edges are discontinuous or continuous in the first tree, it is a trivial D-map of the dependence model obtained from polytree.

Example 2. Let us illustrate Algorithm 2 based on the R-vine of Fig. 3-(left panel).

Step 1. Obtain $M = \{L_I, L_D\}$ from G_{R-vine}.
$L_I = \{I_1\,(1,3 \mid 2)\}$
$L_D = \{D_1\,(2,4 \mid 3)\,, D_2\,(1,4 \mid 2,3)\}$
Step 2. The skeleton of G_P is the tree T_1 of G_{R-vine}.
Step 3. By representing the independence I_1, the graphs of Fig. 3-(right panel, top) are obtained.
Step 4. To represent $D_1\,(2,4 \mid 3)$, the node 3 must be considered a head-to-head node. This is done by adding the directed edge $4 \rightarrow 3$. This can be done without adding or removing independencies. However, the relationship D_2 cannot be represented since it can only be inserted as an independence in the polytree. Therefore, the final graph remains as shown in Fig. 3-(right panel, middle).

The resulting polytree, in addition to representing the same independence relationships existing in the R-vine structure, also includes others that are not verified in M (obtained from a polytree). As the independencies of the R-vine are a subset of those of the polytree, we can say that the R-vine is an I-map of M obtained from a polytree, and that the polytree is a D-map of M obtained from an R-vine as all the dependence relationships of the polytree exist in the R-vine. Notice that from the same R-vine, more that one polytree can be obtained, which is illustrated in Fig. 3-(right panel, bottom).

Example 3. Let us illustrate Algorithm 2 based on the R-vine of Fig. 4-(left panel).

Step 1. Obtain $M = \{L_I, L_D\}$ from G_{R-vine}.
$L_I = \{(I_1\,(2,5 \mid 4)\,, I_2\,(3,5 \mid 2,4)\,, I_3\,(1,5 \mid 2,3,4))\}$
$L_D = \{(D_1\,(1,3 \mid 2)\,, D_2\,(3,4 \mid 2)\,, D_3\,(1,4 \mid 2,3))\}$
Step 2. The skeleton of G_P is the tree T_1 of G_{R-vine}.
Step 3. By representing the independence relationships I_1, I_2 and I_3, the graphs of Fig. 4-(middle panel) are obtained. If we take a look at the last graph, we can see that in G_P not only independencies found in G_{R-vine} are represented, but also other independencies that are not visible in L_I. In order to preserve the same independence relationships, in the next step we proceed to insert the conditional dependencies that are in L_D.
Step 4. To represent $D_1\,(1,3 \mid 2)$, 2 must be considered as head-to-head node, this is done by changing the direction of the edge between the nodes 2 and 3. This can be performed without affecting independence relationships. Analogously, regarding relation D_2, the edge between 2 and 4 is redirected. Nevertheless, D_3 cannot be represented since it would change the existing independencies in the graph. Therefore, the final graph remains as shown in Fig. 4-(right panel).

Algorithm 2. Procedure to build the graph of a polytree from the R-vine tree-structure.

Input: G_{R-vine} consistent
Output: G_P
Step 1 Create the lists L_I and L_D from G_{R-vine}.
for edge e in G_{R-vine}:
if e is a dashed edge:
Add $I(X, Y \mid \mathbf{V})$ to L_I.
else e is a continuous edge:
Add $D(X, Y \mid \mathbf{V})$ to L_D.
Step 2 Obtain the skeleton of G_P as T_1 of G_{R-vine}.
for $I(X, Y \mid \mathbf{V})$ in L_I:
if at least one edge between nodes $X, Y, \mathbf{V}$ is an undirected edge:
Step 3 Orient the edges of the subgraph $X - \mathbf{V} - Y$ without creating a head-to-head node.
for $D(X, Y \mid \mathbf{V})$ in L_D:
if possible to set some node of $\mathbf{V}$ as a head-to-head node:
Step 4 Insert the subgraph $X \rightarrow \mathbf{V} \leftarrow Y$.

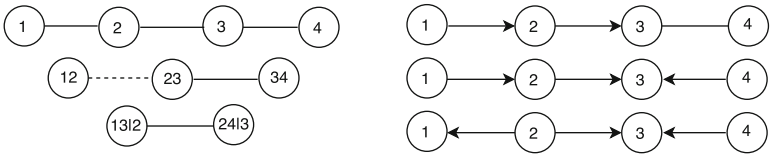

Fig. 3. Illustration of Example 2: (left panel) R-vine G_{R-vine}; (right panel) edge orientation to represent the relationships of $L_I = \{I_1\}$ (top) and $L_D = \{D_1\}$ (middle), and an example of another polytree that can be obtained from the R-vine on the left.

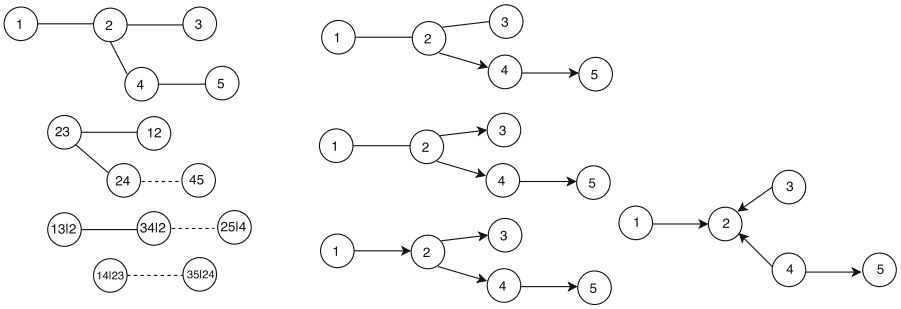

Fig. 4. Illustration of Example 3: (left panel) starting R-vine G_{R-vine}; (middle panel) edge orientation to represent the independence relationships of $L_I = \{I_1, I_2, I_3\}$ (from top to bottom); (right panel) edge orientation to represent the dependence relationships of $L_D = \{D_1, D_2, D_3\}$.

5 Conclusions

In this work, we have studied the connection between the graphical representations of polytrees and R-vines. We have introduced two algorithms for translating between the underlying semantics of polytrees and regular vines from the graphical perspective.

We have shown that we can find an R-vine where all independencies it encodes exist in the polytree, although not all independencies existing in the polytree can be represented in an R-vine. Thus, the R-vine is an I-map of the dependence model obtained from the polytree. On the other hand, given an R-vine, the resulting polytree includes the same independence relationships existing in the R-vine and also others that are not true in the R-vine. As all the dependence relationships inserted in the polytree exist in the R-vine, the polytree is a D-map of the dependence model obtained from the R-vine. An ongoing topic demanding future work is the extension of this study to BNs with undirected cycles.

Acknowledgements. The author would like to thank Dr. Aritz Perez, of Basque Center for Applied Mathematics, BCAM, 48009 Bilbao, Spain, for valuable comments and suggestions. This work is partially supported by the Basque Government (IT609-13 and Elkartek), and Spanish Ministry of Science and Innovation (TIN2016-78365-R). Jose A. Lozano is also supported by BERC 2014–2017 and Elkartek programs (Basque government) and Severo Ochoa Program SEV-2013-0323 (Spanish Ministry of Economy and Competitiveness).

References

1. Aas, K., Czado, C., Frigessi, A., Bakken, H.: Pair-copula constructions of multiple dependence. Insur. Math. Econ. **44**(2), 182–198 (2009)
2. Bauer, A., Czado, C.: Pair-copula Bayesian networks. arXiv:1211.5620 [stat.ME] (2012)
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 vew 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. Castillo, E., Gutiérrez, J.M., Hadi, A.S.: Sistemas Expertos y Modelos de Redes Probabilísticas. Monografias de la Academia de Ingeniería (1996)
7. Cowell, G., Dawid, A.P., Lauritzen, S.L., Spiegelhalter, D.J.: Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks. Springer, New York (2003). https://doi.org/10.1007/b97670
8. de Campos, L.M.: Independency relationships and learning algorithms for singly connected networks. J. Exper. Theor. Artif. Intell. **10**(4), 511–549 (1998)
9. Haff, I.H., Aas, K., Frigessi, A., Lacal, V.: Structure learning in Bayesian networks using regular vines. Comput. Stat. Data Anal. **101**, 186–208 (2016)
10. Hanea, A.M.: Non-parameteric Bayesian belief nets versus vines. In: Joe, H., Kurowicka, D. (eds.) Dependence Modeling: Vine Copula Handbook, pp. 281–303. World Scientific Publishing (2011)

11. Joe, H.: Families of m-variate distributions with given margins and $m(m-1)/2$ bivariate dependence parameters. In: Rüschendorf, L., Schweizer, B., Taylor, M.D. (eds.) Distributions with Fixed Marginals and Related Topics, pp. 120–141 (1996)
12. Kurowicka, D., Cooke, R.M.: Uncertainty Analysis with High Dimensional Dependence Modelling. Wiley, New York (2006)
13. Lauritzen, L.: Graphical Models. Oxford University Press, Oxford (1996)
14. Nelsen, R.B.: An Introduction to Copulas, 2nd edn. Springer, New York (2006). https://doi.org/10.1007/0-387-28678-0
15. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan and Kaufmann, San Mateo (1998)
16. Prim, R.C.: Shortest connection networks and some generalizations. Bell Labs Tech. J. **36**(6), 1389–1401 (1957)
17. Sklar, A.: Fonctions de repartition à n dimensions et leurs marges. Publications de l'Institut de Statistique de l'Universite de Paris **8**, 229–231 (1959)