CHAPTER 11

Optimal Truncation of Vines

Dorota Kurowicka

Delft University of Technology
Mekelweg 4, 2628CD Delft, the Netherlands
D.Kurowicka@tudelft.nl

The vine copulae representation is a very flexible model of n-dimensional random vectors. However when n becomes too large, some simplifying assumptions have to be made as fitting this model becomes too cumbersome. In this chapter truncations of vines are discussed. We could reduce a vine to a Markov tree structure but Markov trees allow $n-1$ copulae to be specified out of $\binom{n}{2}$ possible for vines. Hence trees may be too restrictive for a given set of data. Another possibility would be to model subsets of variables with vines and connect these smaller vines in a tree structure. We suggest one more strategy of choosing the "most suitable" vine for the correlation matrix. The "best vine" is the one whose nodes of top trees (tree with the most conditioning) correspond to the smallest absolute values of partial correlations. To search for the "best vine" we developed a new algorithm of generating a regular vine. We start building the vine from the top node (node in tree n-1) and progress to the lower trees, ensuring that the regularity condition is satisfied and that the partial correlations corresponding to these nodes are the smallest. If we assume that we can assign the independent copula to nodes of the vine with small absolute values of partial correlations, then this algorithm can be used to find an optimal truncation of a vine structure. We advocate using it as a preprocessing step of fitting a vine to the data.

11.1 Introduction

The vine copula representation is a very flexible model of a joint n-dimensional distribution. It allows not only information in marginal distributions to be separated from information about dependencies in the joint distribution; it also gives a way of specifying $\binom{n}{2}$ different bivariate copulae as "building blocks" of the joint distribution. These bivariate building blocks can control correlation structure as well as other features of joint distribution, e.g., tail dependence.[7] Choosing copulae with different tail dependence parameters and different correlations gives us a very rich set of models.

Vine distributions can be quantified with data.[2] When their performance was compared with other existing models,[1,2] it was shown that they outperform other models significantly. If data are not available, vines can be quantified with structured expert judgment.[12]

In Aas *et al.*[2] only two types of regular vines, namely C-vines and D-vines (see Figs. 11.1 and 11.2), were used. Up to the fourth dimension, all possible vines are of these two types. For higher dimensions, other different types of vines are available (see Chapters 7 and 10). In many cases it can be more advantageous to fit different types of regular vines to data.[9] Fitting all possible vines is not feasible as the number of vines grows rapidly with

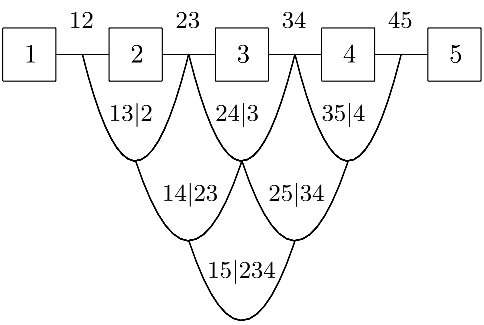

Figure 11.1. A D-vine on five elements showing conditioned and conditioning sets.

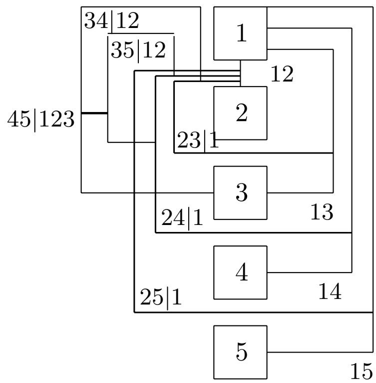

Figure 11.2. A C-vine on five elements showing conditioned and conditioning sets.

dimension.[11] There are three vines on three variables, 24 vines on four variables, 480 on five variables and more than 23,000 on six variables.

In high-dimensional cases where we are unable to fit all possible vines, simplified dependence structures must be considered. We could reduce the model to a Markov tree structure which is a special case of a vine. But as Markov trees allow $n-1$ out of $n(n-1)/2$ copulae to be specified, trees may be too restrictive for a given data. Another possibility would be to model subsets of variables with vines and connect these smaller vines in a tree structure.

In this chapter we suggest a different strategy for choosing the "most suitable" vine for the correlation matrix. The "best vine" is the one whose nodes of top trees (tree with the most conditioning) correspond to the smallest absolute values of partial correlations. To search for the "best vine" we developed a new algorithm of generating a regular vine. We start building the vine from the top node (node in tree $n-1$) and progress to the lower trees, ensuring that the regularity condition is satisfied and that the partial correlations corresponding to these nodes are the smallest. If we assume that we can assign the independent copula to nodes of the vine with small absolute values of partial correlations, then this algorithm will be useful in finding an optimal truncation of a vine structure. In Ref. 10 it is shown that such truncations of vines allow us to build distributions for which the dependence structure corresponds to a chordal graph. We advocate finding an appropriate vine truncation as a preprocessing step of fitting a vine to the data.

This chapter is organized as follows: We first summarize some known facts of regular vines and extend this exposition with a few new properties.

Next, known truncations of vines, namely Markov trees and vines in trees, are discussed. In Section 11.4 a new algorithm for generating a regular vine is presented, which is later used in a heuristic search for "the best" regular vine for the correlation matrix. Finally some results and conclusions are presented.

11.2 Vines

A *vine* on n elements $\mathcal{V} = (T_1, \ldots, T_{n-1})$ is a nested set of trees where the edges of tree j are nodes of tree $j+1$ and each tree has the maximum number of edges. A *regular* vine on n elements is one in which two edges in tree j are joined by an edge in tree $j+1$ only if these edges share a common node. Two edges a, b joined by an edge c in the next tree are called *m-children* of c and c is the *m-parent* of a and b. a and b are called *siblings*. Hence, a regularity property can be formulated: all siblings have a common child.

For each edge of a vine we define *constraint, conditioned* and *conditioning* sets of this edge as follows: The nodes of the first tree reachable from a given edge via the membership relation are called the constraint set of that edge. When two edges are joined by an edge in the next tree, the intersection of the respective constraint sets form the conditioning set, and the symmetric difference of the constraint sets is the conditioned set of this edge. Formal definitions can be found in any of the references above and in Chapter 3. We adopt the following notation: for each edge e of $\mathcal{V}$, let C_e and D_e denote conditioned and conditioning sets of e. Moreover we will exchangeably denote the constraint set of e as $\{C_e|D_e\}$ to indicate conditioned and conditioning sets or simply show variables that this set contains as $C_e \cup D_e$. If $C_e = \{x, y\}$, then we will call x and y *partners* and we will denote that y is a partner of x as $y = pt(x)$.

In Figs. 11.1 and 11.2, two special types of the regular vines, the C-vine and the D-vine, on five elements with conditioned sets and conditioning sets assigned to their edges are shown.

For vines in Figs. 11.1 and 11.2 we can easily check some general properties of regular vines (for proofs and rigorous formulation see Ref. 8 and Chapter 3).

Properties:

(1) There are $n-1$ trees and $\binom{n}{2}$ edges in a regular vine on n elements;
(2) Conditioned sets are doubletons;
(3) Each pair appears once as a conditioned set of an edge;

(4) There are $i-1$ and $i+1$ elements in the conditioning and constraint sets of an edge of the ith tree, respectively;
(5) If two nodes have the same constraint sets, they are the same node;
(6) If element i is a member of the conditioned set of an edge e of a regular vine, then i is a member of the conditioned set of exactly one of the m-children of e and the conditioning set of an m-child is a subset of D_e.

The following two lemmas can be added to vine properties.

Lemma 11.1. *Let $A \subset \{1,\ldots,n\}$ and $x_1, x_2 \notin A$, $x_1 \neq x_2$ and $y_1, y_2 \in A$. Let $N_1 = \{x_1, y_1 | A \setminus \{y_1\}\}$ and $N_2 = \{x_2, y_2 | A \setminus \{y_2\}\}$ be nodes of tree T_i of regular vine on n variables; then N_1 and N_2 have a common m-child. Moreover if $y_1 \neq y_2$, then this common m-child is: $\{y_1, y_2 | A \setminus \{y_1, y_2\}\}$.*

Proof. Node N_1 has two m-children with constraint sets $\{x_1, A \setminus \{y_1\}\}$ and A and N_2 has two m-children whose constraint sets are $\{x_2, A \setminus \{y_2\}\}$ and A. From property [5], N_1 and N_2 have a common m-child.

If $y_1 \neq y_2$, then from property [6], y_1 and y_2 have to be in the conditioned set of the m-child. □

Lemma 11.2. *Let $A \subset \{1,\ldots,n\}$, $x_1, x_2, \ldots, x_k \notin A$, $x_j \neq x_r$, for $j \neq r$ and $y_1, y_2, \ldots, y_k \in A$. Let $N_1 = \{x_1, y_1 | A \setminus \{y_1\}\}$, $N_2 = \{x_2, y_2 | A \setminus \{y_2\}\}$, $\ldots, N_k = \{x_k, y_k | A \setminus \{y_k\}\}$ be nodes of tree T_i of regular vine on n variables, then*

$$y_j = s \text{ or } y_j = t, \quad s, t \in \{y_1, \ldots, y_k\}.$$

Proof. Each N_j has two m-children. One of them has the constraint set A. Hence all N_j's have a common m-child. By Lemma 11.1 if $y_j \neq y_r$, then they both have to belong to the conditioned set which concludes the proof. □

11.3 Vine Distributions

In Bedford and Cooke[3] the following representation theorem for joint density in terms of product of (conditional) copulae assigned to the edges of a vine and the marginal densities is proven.

Theorem 11.1. *Let $(F, \mathcal{V}, B)$ be a copula vine specification where: $F = (F_1, \ldots, F_n)$ and each F_i has density f_i, $i = 1, \ldots, n$, $\mathcal{V}$ is a regular vine on n elements and $B = \{C_{jk} \mid e(j,k)$ where $e(j,k)$ is the unique edge with conditioned set $\{j,k\}$, and C_{jk} is a copula for $\{X_j, X_k\}$ conditional*

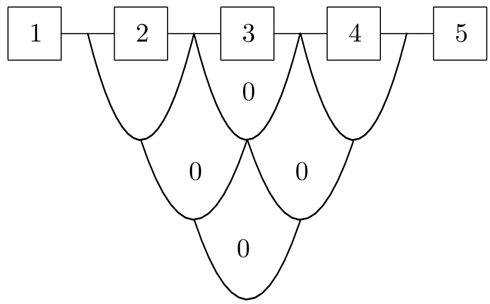

Figure 11.3. A D-vine on five variables with four conditional independent copulae assigned to top nodes.

on $D_{e(j,k)}$ with density $c_{jk|D_e}$}. Then the vine-dependent distribution for $(F, \mathcal{V}, B)$ is uniquely determined and has a density given by

$$f_{1\cdots n} = f_1 \cdots f_n \prod_{i=1}^{n-1} \prod_{e(j,k)\in E_i} c_{jk|D_e}(F_{j|D_e}, F_{k|D_e}). \tag{11.1}$$

Assigning independent copula to an edge e of the vine ensures that variables in the conditioned set of e are conditionally independent given variables in D_e. When the independent conditional copulae are assigned to top nodes (with the most conditioning) of the vine, then the density (11.1) simplifies significantly. For the D-vine in Fig. 11.3 with four independent conditional copulae (shown as "0"), the density is of the form:

$$\begin{aligned} f_{1\cdots 5} = f_1 \cdots f_5 c_{12}(F_1, F_2) c_{23}(F_2, F_3) c_{34}(F_3, F_4) c_{45}(F_4, F_5) \\ c_{13|2}(F_{1|2}, F_{3|2}) c_{35|4}(F_{3|4}, F_{5|4}). \end{aligned}$$

Independent conditional copulae assigned to top edges of the vine make the dependence structure much simpler. This process can be seen as a "truncation" of the vine to a more constrained model. Below, some special cases of vines are briefly discussed.

11.3.1 *Markov trees*

If all conditional copulae are assumed to be the independent copula, then the vine reduces to a Markov tree. The first trees of the D-vine and the C-vine in Figs. 11.1 and 11.2 are shown in Fig. 11.4. For both models, only four out of ten copulae can be specified: $c_{12}, c_{23}, c_{34}, c_{45}$ for the D-vine and $c_{12}, c_{13}, c_{14}, c_{15}$ for the C-vine.

To choose the best tree for the data we can use one of the minimum spanning tree algorithms (see, e.g., Ref. 6). Weights assigned to edges of the

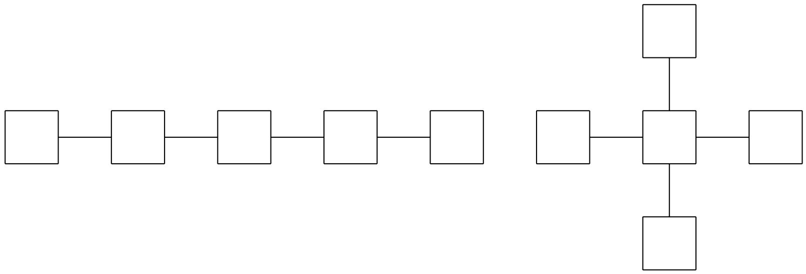

Figure 11.4. First trees of the D-vine (left) and the C-vine (right).

saturated graph might be chosen equal to correlations, tail indices, combinations of those or other features of the joint data.

11.3.2 *Vines in trees*

Markov trees allow $n-1$ out of $n(n-1)/2$ copulae to be specified. Hence trees may be too restrictive for a given set of data. Another possibility is to model subsets of variables with vines and connect these smaller vines in a tree structure. A simple example is presented in Fig. 11.5.

If the structure in Fig. 11.5 is such that variable 3 from the first subvine is connected with variable 4 from the second one, then this structure represents a truncation of the D-vine on six variables in which only the first tree and two conditional copulae in the second tree can be different from the independent copula.

If we understand the edge between two vines as connecting distributions of variables $\{1,2,3\}$ and $\{4,5,6\}$, then the structure in Fig. 11.5 will correspond to a chain graph with two chain components $\{1,2,3\}$ and $\{4,5,6\}$ (see, e.g., Ref. 5). Distributions represented as chain graphs, however, are not so flexible as vines in terms of sampling and quantification.

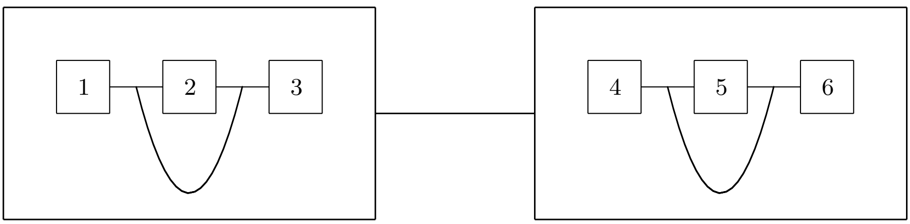

Figure 11.5. Vines in a tree.

11.4 Optimal Truncation

In this section a strategy for choosing the "most suitable" truncation of the vine for the correlation matrix is described. For this purpose we propose a new way of generating a regular vine.

11.4.1 *Generating regular vines*

A regular vine on n variables can be generated in different ways. One way is to follow its definition: choose the first tree; for $j = 2, \ldots, n-1$, build T_j by connecting two edges in T_{j-1} if they share a common node.

A different algorithm is presented in Morales-Nápoles *et al.*[11] and Chapters 9 and 10, based on extending a vine on $j-1$ variables by adding the variable j. Choices for these extensions have to satisfy a certain condition which assures the regularity. It is shown that there are 2^{j-3} possible choices for the extension.

Our algorithm will start building a vine from the top tree, T_{n-1}, that has only one edge. We can choose any pair of variables to be in the conditioned set of the top edge. If we have chosen n and $n-1$, then the constraint set of this edge is $\{n, n-1|1, \ldots, n-2\}$. Constraint sets of its two m-children are of the form $\{n, 1, \ldots, n-2\}$ and $\{n-1, 1, \ldots, n-2\}$. We must choose partners of n and $n-1$ in T_{n-2} such that regularity will be satisfied. Hence we first develop conditions that by Lemma 11.1 and 11.2 are necessary for regularity and then show that when applied recursively in all trees they will insure regularity.

Consider all sets of the form $B_x = \{x\} \cup A_x$ where $x \in C_e$ and $A_x = D_e$ for some edge e of tree T_j, $j = 2, \ldots, n-2$.

Condition 1.

Suppose $B_x = B_y$ and $x \neq y$, then there exists an edge f of T_{j-1} such that $C_f = \{x, y\}$ and $D_f = A_x \backslash \{y\}$.

Condition 2.

For all $B_{i_1}, \ldots, B_{i_k}$ such that $|B_{i_p} \triangle B_{i_u}| = 2$,

$$B_{i_p} = \{i_p, s|A_{i_p} \backslash \{s\}\} \quad \text{or} \quad B_{i_p} = \{i_p, t|A_{i_p} \backslash \{t\}\}, \; s, t \in A_{i_p}.$$

where $U \triangle V$ denotes the symmetric difference of sets U and V.

The algorithm that builds a regular vine can now be stated as follows:

Algorithm A

Step 1. Choose two variables, say $x, y \in \{1, \ldots, n\} = I$, to be in the conditioned set of the top edge; constraint sets of its m-children are $B_x = \{x\} \cup \{I \backslash \{x, y\}\}$ and $B_y = \{y\} \cup \{I \backslash \{x, y\}\}$; choose partners of x and y in T_{n-2} from the set $I \backslash \{x, y\}$. Then there are two edges in T_{n-2}:

$$E_{n-2} = \{\{x, pt(x)|I \backslash \{x, y, pt(x)\}\}, \ \ \{y, pt(y)|I \backslash \{x, y, pt(y)\}\}\}\,.$$

For all $j = n-2, \ldots, 1$

Step 2. Set $B_x = \{x\} \cup A_x$ such that $x \in C_e$ and $A_x = D_e$ for each edge e of tree T_j;

Step 3. Remove all sets for which $x_i = x_k$, for $i \neq k$;

Step 4. Apply Condition 1;

Step 5. Choose partners of variables x such that Condition 2 is satisfied;

Edge set of T_{j-1} is the set containing elements of the form $\{x, pt(x)|A_x \backslash \{pt(x)\}\}$.

We prove that the above algorithm always produces a regular vine.

Theorem 11.2. *Algorithm A produces a regular vine.*

Proof. We prove that the recursive application of Algorithm A to the jth level produces the jth tree of a regular vine. It is enough to show that the procedure insures that all siblings in tree T_j have a common m-child in tree T_{j-1}. Obviously the statement is true for initiating Step 1 as Condition 1 insures that two siblings in T_{n-2} have a common m-child in T_{n-3}. Suppose for all trees $j = n-1, \ldots, k+1$ that all siblings in T_j have a common child in T_{j-1}. Step 2 of the algorithm creates $2(n-k)$ constraint sets of m-children of edges of T_k. These sets are indexed by variables that were in conditioned sets of edges of T_k. Multiple instances of these sets are removed in Step 3. At this point some sets can be equal but they are indexed by different variables. We combine them in Step 4 to satisfy Condition 1 as by Property [5] they are constraint sets of the same edge. Since both indexing variables were in the conditioning set of an edge in T_k, then by Lemma 11.1 they have to be in the conditioned set of their common m-child. We get now that all constraint sets of m-children of edges in T_k are different. The symmetric difference of constraint sets of siblings in

T_k has two elements and the symmetric difference of constraint sets of their m-children has also two elements. Condition 2 applied to all sets B such that $|B_v \triangle B_u| = 2$, insures by Lemma 2 that all siblings in T_k have a common m-child in T_{k-1}. $\square$

Remark 11.1. Notice that the edges of T_1 are already obtained after Step 3 of Algorithm A.

We will use the algorithm described in this section in finding the "best vine" for the correlation matrix.

11.4.2 *Best vine*

Quantifying a vine with data is performed sequentially by first fitting copulae on the first tree, then transforming data through the fitted copulae to find copula parameters on the second tree, etc.[2] In fitting high-dimensional vine distributions it would be of interest to first find a vine structure with the maximum number of independent copulae in the top nodes that offers the best approximation of the data. We propose to base this choice on a partial correlation vine corresponding to the correlation matrix obtained from data. Partial correlations[13] are calculated from the correlation matrix as follows:

$$\rho_{ij;I\setminus\{i,j\}} = \frac{C_{i,j}}{\det(I)}$$

where $\det(I)$ denotes the determinant of the correlation matrix of variables in I and $C_{i,j}$, its (i,j)th cofactor. Partial correlations are assigned to the edges of a vine such that conditioning variables are equal to the conditioning set and the conditioned variables are equal to the conditioned set.

For elliptically contoured distributions, partial correlations are equal to conditional correlations. For the normal distribution, zero partial correlation corresponds to conditional independence. In general, however, zero partial correlation does not have to indicate conditional independence. Nevertheless, in our algorithm for finding the best vine, we will choose partners of variables x in Algorithm A such that Condition 2 is satisfied and such that the absolute values of partial correlations $\rho_{x,pt(x);A_x\setminus pt(x)}$ are the smallest.

11.5 Optimal Truncation: Results

We start this section with a simple example to show how the algorithm works and then test its performance. We conclude with an application of the algorithm to the data matrix analyzed in Ref. 9. A comparison of the performance of our algorithm and the algorithm based on the majorization principle proposed in Ref. 9 is performed.

11.5.1 *Example*

Consider the following correlation matrix:

$$M = \begin{bmatrix} 1 & 0.2 & 0.4 & 0.5 & 0.7 \\ 0.2 & 1 & 0.3 & 0.6 & 0.7 \\ 0.4 & 0.3 & 1 & 0.8 & 0.5 \\ 0.5 & 0.6 & 0.8 & 1 & 0.8 \\ 0.7 & 0.7 & 0.5 & 0.8 & 1 \end{bmatrix}.$$

The normalized inverse matrix of M is:

$$\begin{bmatrix} 1 & 0.5309 & -0.2034 & 0.2246 & -0.7437 \\ 0.5309 & 1 & 0.0914 & -0.0827 & -0.6021 \\ -0.2034 & 0.0914 & 1 & -0.7930 & 0.3236 \\ 0.2246 & -0.0827 & -0.7930 & 1 & -0.5613 \\ -0.7437 & -0.6021 & 0.3236 & -0.5613 & 1 \end{bmatrix}.$$

We choose the smallest absolute partial correlation which is 0.0827. This is the partial correlation $\rho_{24;13}$. Then we follow Algorithm A and obtain that the best vine, with the smallest partial correlations in the top nodes, is:

$$V = \begin{bmatrix} & C_e & & D_e & & \rho_{C_e;D_e} \\ 2 & 4 & | \, 1 & 3 & 5 & = 0.0827 \\ 2 & 3 & | \, 1 & 5 & & = 0.0425 \\ 4 & 1 & | \, 3 & 5 & & = 0.3179 \\ 3 & 1 & | \, 5 & & & = 0.0808 \\ 2 & 1 & | \, 5 & & & = 0.5686 \\ 4 & 3 & | \, 5 & & & = 0.7698 \\ 3 & 5 & & & & = 0.5 \\ 2 & 5 & & & & = 0.7 \\ 1 & 5 & & & & = 0.7 \\ 4 & 5 & & & & = 0.8 \end{bmatrix}.$$

The vine obtained is neither a D-vine nor a C-vine (see Fig. 11.6). If we assume that we can assign the independent copula to nodes of the vine with small absolute partial correlations, then we see that only seven out of ten copulae $c_{15}, c_{25}, c_{35}, c_{45}$ and $c_{43|5}, c_{12|5}, c_{41|35}$ have to be fitted. Moreover, because of the sequential fitting of copula in vines, the estimates for copulae

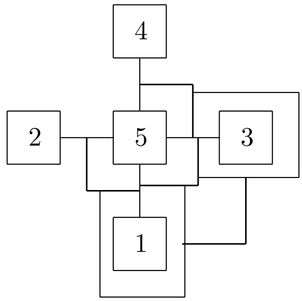

Figure 11.6. Vine corresponding to matrix M.

with more conditioned variables are not so accurate. Removing them may be of benefit.

If we decided to search for a vine with the highest correlations in lower trees, then we would end up with the following vine:

$$V^* = \begin{array}{c} \begin{array}{cccccc} & C_e & & D_e & & \rho_{C_e;D_e} \end{array} \\ \begin{bmatrix} 2 & 3 & \mid 1 & 4 & 5 & = -0.0914 \\ 2 & 4 & \mid 1 & 5 & & = 0.0169 \\ 1 & 3 & \mid 4 & 5 & & = 0.2985 \\ 3 & 5 & \mid 4 & & & = -0.3889 \\ 2 & 1 & \mid 5 & & & = -0.5686 \\ 4 & 1 & \mid 5 & & & = -0.1400 \\ 3 & 4 & & & & = 0.8 \\ 2 & 5 & & & & = 0.7 \\ 1 & 5 & & & & = 0.7 \\ 4 & 5 & & & & = 0.8 \end{bmatrix} \end{array}.$$

We see that in this case the top node is also associated with small partial correlation but in general it does not have to be the case.

Table 11.1 presents sums and average partial correlations in trees of V and V^*. We see that the partial correlation of the top edge of V is slightly smaller than the partial correlation of the top edge of V^*. The choice of slightly smaller correlations in T_4 leads to significantly higher average correlations in T_3.

11.5.2 *Comparison*

We consider an example treated in Ref. 4. We have eight variables corresponding to weather monitoring stations in Europe. The original data has

Table 11.1. Sum and average absolute values of partial correlations in trees of V and V^*.

	T_4	T_3	T_2	T_1
sum V	0.0827	0.3604	1.4192	2.7
average V	0.0827	0.1802	0.4731	0.675
sum V^*	0.0914	0.3154	1.0975	3
average V^*	0.0914	0.1577	0.3658	0.75

a sample correlation matrix:

$$\begin{bmatrix} 1 & .35 & .50 & .49 & .68 & .38 & .50 & .59 \\ & 1 & .79 & .69 & .12 & .64 & .62 & .49 \\ & & 1 & .72 & .18 & .61 & .58 & .43 \\ & & & 1 & .05 & .46 & .47 & .43 \\ & & & & 1 & .33 & .51 & .71 \\ & & & & & 1 & .97 & .77 \\ & & & & & & 1 & .90 \\ & & & & & & & 1 \end{bmatrix}.$$

In Kurowicka *et al.*[9] a heuristic search of a vine was adopted for which the logarithm of one minus squared partial correlations assigned to its edges majorizes all others based on minimizing the entropy function.

The heuristic works as follows:

(1) Choose an ordering of the variables.
(2) Start with subvine consisting of variables 1 and 2 in the ordering. For $j = 3, \ldots, n$, find the subvine extending the current subvine by adjoining variable $j+1$, so as to minimize entropy function of $\log(1-\rho^2_{C_e;D_e})$. Store the vine obtained for $j = n$.
(3) Go to 1.
(4) Choose the optimal partial correlation vine minimizing entropy among all those stored.

In general it is not feasible to search all permutations; heuristic search methods or Monte Carlo sampling must be used. The optimal vine V^* obtained with this procedure is shown below. We can see that there are many small partial correlations in this vine but they are not necessarily in the top trees. The vine V was obtained with Algorithm A. We can see that there are many more small correlation values in trees with higher indices.

Table 11.2. Sum and average absolute values of partial correlations in trees of V and V^*.

	T_7	T_6	T_5	T_4	T_3	T_2	T_1
sum V	0.0109	0.2382	1.1841	1.3341	2.3696	1.5308	4.3384
average V	0.0109	0.1191	0.3947	0.3335	0.4739	0.2551	0.6198
sum V^*	0.8883	1.4657	0.2035	1.3987	1.9273	1.5745	3.6045
average V^*	0.8883	0.7328	0.0678	0.3497	0.3855	0.2624	0.5149

$$V = \begin{bmatrix} & C_e & D_e & \rho_{C_e;D_e} \\ 2 & 5 & \mid 134678 & 0.0109 \\ 2 & 8 & \mid 13467 & 0.0823 \\ 5 & 7 & \mid 13468 & 0.1559 \\ 8 & 7 & \mid 1346 & 0.9499 \\ 2 & 6 & \mid 1347 & 0.0377 \\ 5 & 3 & \mid 1468 & 0.1965 \\ 7 & 6 & \mid 134 & 0.9666 \\ 8 & 3 & \mid 146 & -0.3310 \\ 2 & 1 & \mid 347 & 0.2756 \\ 5 & 6 & \mid 148 & 0.4229 \\ 6 & 3 & \mid 14 & 0.4250 \\ 7 & 1 & \mid 34 & 0.2854 \\ 8 & 6 & \mid 14 & 0.7166 \\ 2 & 4 & \mid 37 & 0.2802 \\ 5 & 4 & \mid 18 & 0.6624 \\ 3 & 1 & \mid 4 & 0.2495 \\ 7 & 4 & \mid 3 & 0.0936 \\ 8 & 4 & \mid 1 & 0.1944 \\ 6 & 1 & \mid 4 & 0.2037 \\ 2 & 7 & \mid 3 & 0.3236 \\ 5 & 1 & \mid 8 & 0.4660 \\ 7 & 3 & & 0.5822 \\ 6 & 4 & & 0.4602 \\ 2 & 3 & & 0.7888 \\ 5 & 8 & & 0.7140 \\ 3 & 4 & & 0.7155 \\ 1 & 4 & & 0.4890 \\ 8 & 1 & & 0.5886 \end{bmatrix} \quad V^* = \begin{bmatrix} & C_e & D_e & \rho_{C_e;D_e} \\ 7 & 8 & \mid 123456 & 0.8883 \\ 6 & 7 & \mid 12345 & 0.9783 \\ 4 & 8 & \mid 12356 & 0.4874 \\ 4 & 6 & \mid 1235 & 0.0171 \\ 4 & 7 & \mid 1235 & 0.1086 \\ 1 & 8 & \mid 2356 & 0.0778 \\ 1 & 7 & \mid 235 & -0.0591 \\ 1 & 6 & \mid 235 & -0.0961 \\ 1 & 4 & \mid 235 & 0.4632 \\ 5 & 8 & \mid 236 & 0.7803 \\ 2 & 4 & \mid 35 & 0.2928 \\ 1 & 5 & \mid 23 & 0.6998 \\ 5 & 6 & \mid 23 & 0.3068 \\ 5 & 7 & \mid 23 & 0.5450 \\ 3 & 8 & \mid 26 & -0.0829 \\ 1 & 3 & \mid 2 & 0.3935 \\ 3 & 4 & \mid 5 & 0.7190 \\ 2 & 5 & \mid 3 & -0.0309 \\ 3 & 6 & \mid 2 & 0.2208 \\ 3 & 7 & \mid 2 & 0.1918 \\ 2 & 8 & \mid 6 & -0.0185 \\ 1 & 2 & & 0.3491 \\ 2 & 3 & & 0.7888 \\ 3 & 5 & & 0.1762 \\ 4 & 5 & & 0.0487 \\ 2 & 6 & & 0.6424 \\ 2 & 7 & & 0.6210 \\ 6 & 8 & & 0.9783 \end{bmatrix}$$

11.6 Conclusions

The new algorithm for generating a regular vine presented in this chapter allows us to build a vine starting from the edge in tree $n-1$, progressing to lower trees, making sure that the regularity condition is satisfied. We proposed applying this algorithm in a heuristic search for a vine with small absolute values of partial correlations assigned to its top edges that corresponds to a given correlation matrix. It was observed that the choice of a small partial correlation in tree T_j may severely constrain the choices that we have in tree T_{j-1}. We could improve the heuristic by basing our choices on more than one tree at a time.

References

1. Aas K. and Berg D. (2009). Models for construction of multivariate dependence: A comparison study. *European Journal of Finance*, 15:639–659.
2. Aas K., Czado C., Frigessi A. and Bakken H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198.
3. Bedford T.J. and Cooke R.M. (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial Intelligence*, 32:245–268.
4. Callies U., Kurowicka D. and Cooke R. (2003). Graphical models for the evaluation of multisite temperature forecasts: Comparison of vines and independence graphs. *Proceedings of ESREL 2003, Safety and Reliability*, 1:363–371.
5. Cowell R., David A., Lauritzen S. and Spiegelhalter D. (1999). *Probabilistic Networks and Expert Systems, Statistics for Engineering and Information Sciences.* Springer-Verlag, New York.
6. Even S. (1979). *Graph Algorithms.* Computer Science Press, Potomac, MD.
7. Joe H. (1997). *Multivariate Models and Dependence Concepts.* Chapman & Hall, London.
8. Kurowicka D. and Cooke R.M. (2006). *Uncertainty Analysis with High Dimensional Dependence Modelling.* Wiley, Chichester.
9. Kurowicka D., Cooke R.M. and Callies U. (2006). Vines inference. *Brazilian Journal of Probability and Statistics*, 20:103–120.
10. Kurowicka D. and Cooke R.M. (2006). Completion problem with partial correlation vines. *Linear Algebra and Its Applications*, 418(1):188–200.
11. Morales-Nápoles O., Cooke R.M. and Kurowicka D. (2009). The number of vines and regular vines on n nodes. Submitted to *Discrete Applied Mathematics.*
12. Morales-Nápoles O., Kurowicka D. and Roelen A. (2007). Elicitation procedures for conditional and unconditional rank correlations. *Reliability Engineering and Systems Safety*, 95(5):699–710.
13. Yule G.U. and Kendall M.G. (1965). *An Introduction to the Theory of Statistics*, 14th ed. Charles Griffin & Co., Belmont, California.