Sampling, Conditionalizing, Counting, Merging, Searching Regular Vines.

R.M. Cooke,* D. Kurowicka, K. Wilson

Email: *cooke@rff.org* †

Abstract

We present a sampling algorithm for a regular vine on n variables which starts at an arbitrary variable. A sampling order whose nested conditional probabilities can be written as products of (conditional) copulas in the vine and univariate margins is said to be implied by the regular vine. We show that there are 2^{n-1} implied sampling orders for any regular vine on n variables. We show that two regular vines on n and m distinct variables can be merged in 2^{n+m-2} ways. This greatly simplifies the proof of the number of regular vines on n variables. A notion of sampling proximity based on numbers of shared implied sampling orders is introduced, and we use this notion to define a heuristic for searching vine space that avoids proximate vines.

Keywords: regular vine, number of regular vines, vine conditionalization, vine merging

1 Introduction

Vines celebrated their 20th anniversary in 2014. From the first construction in [1], and the formal definition in [2], there are now 481 articles on *"regular vine" OR "vine copula"* on Google scholar. Theoretical issues like the Fréchet problem and the completion problem motivated their introduction [3], but their subsequent popularity is largely due to the maximum likelihood estimation procedure of pair copula constructions [4]. As applications accumulate, current themes in vine research cluster around estimation, optimization and model selection. The enabling technology is the R-package from the TU Munich [5]. Vines were crucial in constructing continuous discrete non parametric Bayesian Belief Nets, which have since colonized another continent. Among the many active research themes, this article focuses on sampling, conditionalization, merging and searching vine space. A byproduct is a shorter proof of the number of regular vines on n variables [6] and a new result on merging distinct regular vines. An appendix gives definitions of regular vines and some properties used in the proofs.

Using regular vines for learning and updating requires fast algorithms for conditionalizing on observations. A fast conditionalization algorithm, called "Edging up" is presented in Section

---

3. This algorithm works for any set of variables which are connected in the lowest order tree. When that is not the case, the regular vine representation must first be transformed to one in which the conditioning variables are contiguous in the first tree. A vine version of Bayes theorem called the "thumb rule" can be useful in reducing the calculations somewhat. Just as Bayes' theorem gives equivalent representations for a joint probability, the thumb rule looks for equivalent representations of a joint density on regular vines. A measure of the proximity of vines is introduced based on the number of permutations of sampling orders that vines share. This is used to define a method of searching through vine space more efficiently. An appendix gives basic definitions, for background on vines see [1], [2], [7], [8].

2 Preliminaries and Notation

Definitions and basic properties of vines are given in the appendix. This section introduces some notation and develops a few properties. Throughout, $X_1, \ldots, X_n$ are continuous random variables with positive densities, and hence with invertible cumulative distribution functions. Values of the random variables are given in lower case. Conditioning is understood as a Radon Nikodym differentiation; however, as all variables have joint densities, we freely assume the standard representation of conditional probabilities as ratios of densities. The multivariate density decomposition in [7] shows that any multivariate density can be written as a product of bivariate and conditional bivariate densities on any regular vine.

The following notation is used throughout. It is based on [9] is designed to suppress cascading subscripts when switching from variables to probability integral transforms. '$\sim$' means 'has the same distribution as'.

$$
\begin{aligned}
F_i(a) &:= P(X_i \leq a); \quad a \in \Re, \\
F_{i|j}(a|x_j) = F_{i|x_j}(a) &:= P(X_i \leq a|X_j = x_j); \quad a \in \Re, \\
C_i(a) &:= P(F_i(X_i) \leq a); \quad a \in (0,1), \\
C_{i|j}(a|x_j) = C_{i|x_j}(a) &:= P(F_i(X_i) \leq a|X_j = x_j); \quad a \in (0,1), \\
C_{i|j,k}(a|x_j,x_k) = C_{i|x_j,x_k}(a) &:= P(F_i(X_i) \leq a|X_j = x_j, X_k = x_k); \quad a \in (0,1),
\end{aligned}
$$

Proposition 2.1. *Let $U \sim U(0,1)$ and $a \in (0,1)$, then*

1. $C_{i|x_j}(a) = F_{i|x_j}(F_i^{-1}(a))$;

2. $C_{i|x_j,x_k}(a) = P(C_{i|x_j}(F_i(X_i)) \leq a|X_k = x_k) = P(X_i \leq F_i^{-1} \circ C_{i|x_j}^{-1}(a)|X_k = x_k)$;

3. $(X_i|X_j = x_j) = (X_i|x_j) \sim F_i^{-1}(C_{i|x_j}^{-1}(U))$;

4. $(X_i|X_j = x_j, X_k = x_k, \ldots, X_n = x_n) \sim F_i^{-1} \circ C_{i|x_j}^{-1} \circ C_{i|x_j,x_k}^{-1} \circ \cdot \circ C_{i|x_j,x_k,\cdots x_n}^{-1}(U)$.

Proof: (1) Let $a \in (0,1)$. Then

$$C_{i|x_j}(a) = P(F_i(X_i) \leq a | X_j = x_j) = P(X_i \leq F_i^{-1}(a)|X_j = x_j) = F_{i|x_j}(F_i^{-1}(a)).$$

(2)

$$\begin{aligned}
C_{i|x_j,x_k}(a) &= P(F_i(X_i) \leq a | X_j = x_j, X_k = x_k) = \mathbb{E}[I(F_i(X_i) \leq a)|X_j = x_j, X_k = x_k] \\
&= \mathbb{E}\{\mathbb{E}[I(F_i(X_i) \leq a)|X_j = x_j]|X_k = x_k\} = \mathbb{E}\{I(C_{i|x_j}(F_i(X_i)) \leq a)|X_k = x_k\} \\
&= P(C_{i|x_j}(F_i(X_i)) \leq a | X_k = x_k).
\end{aligned}$$

(3) $(X_i|X_j = x_j) \sim F_{i|x_j}^{-1}(U) = F_i^{-1}(C_{i|x_j}^{-1}(U))$ since the distribution of $(X_i|X_j = x_j)$ is $F_{i|x_j}$ and $F_{i|x_j} = C_{i|x_j} \circ F_i$ from part (1).

(4) Consider $(X_i|X_j = x_j, X_k = x_k) = (X_0|X_k = x_k)$ where let $X_0 = (X_i|X_j = x_j)$. Then, by parts (3) and (1),

$$(X_0|X_k = x_k) \sim F_0^{-1} \circ C_{0|x_k}^{-1}(U) = F_0^{-1} \circ C_{i|x_j,x_k}^{-1}(U) = F_i^{-1} \circ C_{i|x_j}^{-1} \circ C_{i|x_j,x_k}^{-1}(U).$$

The general statement follows by iteration.

3 The Thumb Rule and Marginalization

A bivariate copula C_{12} for random variables (X_1, X_2) expresses the joint cumulative distribution function F_{12} as a function of the variables' probability integral transforms $F_1(X_1), F_2(X_2)$. The copula density c_{12} is obtained by differentiation and

$$\begin{aligned}
F_{12}(x_1, x_2) &= C_{12}(F_1(x_1), F_2(x_2)); \\
f_{12}(x_1, x_2) &= c_{12}(F_1(x_1), F_2(x_2))f_1(x_1)f_2(x_2);
\end{aligned}$$

The joint conditional copula density for X_i and X_j given X_k is written as

$$c(F_i(x_i), F_j(x_j)|X_k = x_k) = \frac{f_{ij|k}(x_i, x_j|x_k)}{f_{i|k}(x_i|x_k)f_{j|k}(x_j|x_k)} = c(F_{i|k}, F_{j|k}; X_k = x_k) = c_{i,j;k}.$$

Note that $c_{i,j;k}$ can depend on X_k in two ways. First, X_k influences the functions $F_{i|k}$ and $F_{j|k}$. Second, X_k may influence the function c. If the latter influence is not present, we speak of the 'simplifying assumption of constant conditional copula'. Note also that the first type of influence is masked in the very sparse notation $c_{i,j;k}$; the objected denoted by "i" may be a function of the object denoted by "k". The choice among these notations will depend on context.

Following [7] the density $f_{1,2,\ldots,n}$ of $(X_1, \ldots, X_n)$ can be written:

$$f_{1,2,\ldots,n}(x_1, \ldots, x_n) = f_1(x_1) \ldots f_n(x_n) \prod_{e \in \mathcal{V}} c_{e_1, e_2; D(e)} \tag{3.1}$$

where $\mathcal{V}(n)$ is any regular vine on n variables, e is an edge in $\mathcal{V}(n)$ with conditioned sets $\{e_1, e_2\}$, conditioning set $D(e)$ and $c_{e_1,e_2;D(e)}$ is the (conditional) copula associated with edge e. The conditional copulas in (3.1) in general do not satisfy the simplifying assumption. Archimedean copulas strictly satisfying the simplifying assumption in dimension $d \geq 4$ are based on the gamma Laplace transform or its extension, and the Student-t copula is the only one arising from a scale mixture of Normals [10]. In spite of these constraints it has proven to be a useful modeling tool [11]. The main barrier to discharging this assumption is the lack of useful low parameter alternatives.

Equation 3.1 gives some insight into the effect of the simplifying assumption. Consider a simple example where the density f of random vector (X_1, X_2, X_3) is represented on a D-vine with first tree $1 - 2 - 3$. Then

$$f_{123}(x_1, x_2, x_3) \quad = \quad f_1(x_1)f_2(x_2)f_3(x_3)c_{1,2}c_{2,3}c_{1,3;2}. \tag{3.2}$$

Proposition 3.1. *With X_1 and X_2 independent, and X_2 and X_3 independent, if the conditional copula function $c_{1,3;2}$ does not depend on the value of X_2, then X_2 is independent of $\{X_1, X_3\}$.*

Proof. We show $f_{2|1,3}(x_2|x_1, x_3) = f_2(x_2)$. Since X_2 is independent of X_1 and X_2 is independent of X_3 then in the density expression (3.2), we have $c_{1,2} = c_{2,3} = 1$. In this case $C_{1|2}$ and $C_{3|2}$ are uniform and they do not depend on X_2. Since the conditional copula function does not depend on X_2 then $c_{1,3;2}$ is constant in x_2. Hence

$$\frac{f_{123}(x_1, x_2, x_3)}{f_{13}(x_1, x_3)} = \frac{f_2(x_2)c_{1,3;2}f_1(x_1)f_3(x_3)}{f_{13}(x_1, x_3)}.$$

Integrating both sides with respect to x_2, we see that $c_{1,3;2} = f_{13}(x_1, x_3)/f_1(x_1)f_3(x_3) = c_{1,3}$ (which does not depend on 2), and $f_{2|1,3}(x_2|x_1, x_3) = f_2(x_2)$. $\square$

Remark 3.1. *It is easy to check that the partial correlation $\rho_{1,3;2}$ is equal to the correlation $\rho_{1,3}$ if $\rho_{1,2} = \rho_{2,3} = 0$. The simplifying assumptions allows us to replace "zero correlation" with "independence", which of course is a much stronger statement.*

Suppose we have two representations of the same density, in terms of 2 D-vines with first trees: 1 - 2 - 3 and 1 - 3 - 2. We may write (all densities are positive):

$$c_{1,2}c_{2,3}c_{1,3;2} \quad = \quad c_{1,3}c_{3,2}c_{1,2;3} \tag{3.3}$$

$$c_{1,3;2} \quad = \quad \frac{c_{1,3}c_{1,2;3}}{c_{1,2}}. \tag{3.4}$$

If we write this as probabilities we would get

$$p(1,3|2) \stackrel{?}{=} \frac{p(1,3)p(1,2|3)}{p(1,2)} \tag{3.5}$$

which would be true if $p(1|3) = p(1|2)$. However, if we just put our thumb over the top nodes' common variable, 1, in Equation 3.3, as if '1' were not there at all, then Equation 3.5 becomes

a statement of Bayes' theorem. If we consider a D-vine and C-vine representation of a density on 4 variables with '1' as root of the C-vine, then after substituting $c_{1,3;2} = \frac{c_{1,3}c_{2,3;1}}{c_{23}}$ (analogous to Equation 3.3) and cancelling terms:

$$c_{1,4;2,3} = \frac{c_{3,4;1,2}c_{2,4;1}c_{1,4}}{c_{2,4;3}c_{3,4}}. \tag{3.6}$$

Again, if we remove 4 we get a version of Bayes theorem. Mathematically, we remove variables by integration. Of course, 'integrating out' requires caution, as the '1' in $c_{1,2}$ does not mean the same thing as the '1' in $c_{1,3;2}$. In general, finding a marginal distribution of a regular vine density representation is a complex operation. However if the variable is in the conditioned set of the top node, then it is quite easy. A variable in the conditioned set of a node is also in the conditioned set of all its m-children, where an m-child of a node is one whose constraint set (conditioned and conditioning variables) is a subset of the node's constraint set [12]. A variable in the conditioned set of the top node is in the conditioned set of all nodes in which it appears. Integrating this variable out simply comes down to deleting all nodes in which this variable occurs.

Proposition 3.2. *Let $X_1, \ldots, X_n$ have positive density represented on a regular vine $\mathcal{V}(n)$, with nodes including $(n, n-1|1, \ldots, n-2), (n, n-2|1, \ldots, n-3), \ldots, (n, 1)$. Then the marginal density of $X_1, \ldots, X_{n-1}$ has a regular vine representation obtained by removing f_n and the nodes $(n, n-1|1, \ldots, n-2), (n, n-2|1 \ldots, n-3), \ldots, (n, 1)$.*

Proof: The variable X_n occurs only in f_n, $c_{n,n-1;1,\ldots,n-2}$, $c_{n,n-2;1,\ldots,n-3}$, $\ldots$, $c_{n,1}$. Since

$$c_{n,n-2;1,\ldots,n-3} = \frac{f_{n,n-2|1,\ldots,n-3}}{f_{n|1,\ldots,n-3}f_{n-2|1,\ldots,n-3}}$$

the product of all terms containing x_n becomes

$$\frac{f_{n,n-1,1,\ldots,n-2}f_{1,\ldots,n-2}}{f_{n,1,\ldots,n-2}f_{n-1,1,\ldots,n-2}} \frac{f_{n,n-2,1,\ldots,n-3}f_{1,\ldots,n-3}}{f_{n,1,\ldots,n-3}f_{n-2,1,\ldots,n-3}} \ldots \frac{f_{n,1}}{f_n f_1} \cdot f_n.$$

Cancelling terms, this becomes $f_{n|1,\ldots,n-1}$, and integrating over x_n yields 1. Therefore, integrating (3.1) over x_n yields

$$f_{1,\ldots,n-1} = f_1 \ldots f_{n-1} \prod_{e \in \mathcal{V}(n-1)} c_{e_1,e_2;D(e)}$$

where $\mathcal{V}(n-1)$ is the regular vine obtained by eliminating all terms containing n. $\square$

The *Thumb Rule* says that if a density is represented on two regular vines whose top nodes share a common element, then the product of all (conditional) copula containing that element on one regular vine is equal to the product of all (conditional) copula containing that same element in the other vine.

The thumb rule is illustrated with a D-vine with first tree 1-2-3-4-5 and R-vine in Figures 1 and 2.
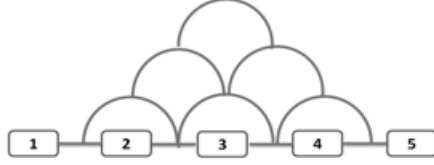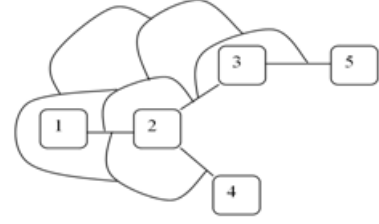
Figure 1: *D*-vine .

Figure 2: *R*-vine

If the same density is represented with (conditional) copulas on these two vines then the marginal densities on nodes $(1,\ldots,4)$ must be equal, and therefore, by the Thumb Rule the product of all copula containing 5 must also be equal.

$$c_{4,5}c_{3,5;4}c_{2,5;4,3}c_{5,1;4,3,2} = c_{3,5}c_{2,5;3}c_{1,5;2,3}c_{4,5;1,2,3}.$$

The density for the *R*-vine above may be written:

$$
\begin{aligned}
f_{12345}(x_1, x_2, x_3, x_4, x_5) &= f_1(x_1)f_2(x_2)f_3(x_3)f_4(x_4)f_5(x_5)c_{4,2}c_{2,1}c_{2,3} \\
&\quad c_{3,5}c_{4,1;2}c_{1,3;2}c_{2,5;3}c_{4,3;1,2}c_{1,5;2,3}c_{4,5;1,2,3} \\
&= f_5(x_5)c_{3,5}c_{2,5;3}c_{1,5;2,3}c_{4,5;1,2,3}f_4(x_4) \\
&\quad c_{2,4}c_{1,4;2}c_{3,4;1,2}f_3(x_3)c_{2,3}c_{1,3;2}f_2(x_2)c_{2,1}f_1(x_1)
\end{aligned}
$$

where the last expression simply re-arranges terms in the previous expression. Since

$$\frac{f_{12345}(x_1, x_2, x_3, x_4, x_5)}{f_{1234}(x_1, x_2, x_3, x_4)} = f_{5|1234}(x_5|x_1, x_2, x_3, x_4)$$

we have

$$
\begin{aligned}
f_{5|1234}(x_5|x_1, x_2, x_3, x_4) &= f_5(x_5)c_{3,5}c_{2,5;3}c_{1,5;2,3}c_{4,5;1,2,3}, \\
f_{4|123}(x_4|x_1, x_2, x_3) &= f_4(x_4)c_{2,4}c_{1,4;2}c_{3,4;1,2}, \\
f_{3|12}(x_3|x_1, x_2) &= f_3(x_3)c_{2,3}c_{1,3;2}, \\
f_{2|1}(x_2|x_1) &= f_2(x_2)c_{2,1}.
\end{aligned}
$$

This corresponds [4] to the familiar decomposition

$$f_{12345}(x_1, x_2, x_3, x_4, x_5) = f_{5|1234}(x_5|x_1, x_2, x_3, x_4)f_{4|123}(x_4|x_1, x_2, x_3)f_{3|12}(x_3|x_1, x_2)f_{2|1}(x_2|x_1)f_1(x_1).$$

Note that these conditional densities can be expressed as products of copula densities on the *R*-vine and one dimensional margins of the regular vine decomposition and they correspond to a sampling order $x_1 \to x_2 \to x_3 \to x_4 \to x_5$ in which each variable is sampled conditional on its predecessors in the ordering.

Definition 3.1 (Sampling order implied by a regular vine)**.** *A sampling order for n variables is a sequence of conditional densities in which the first density is unconditional, and the density*

6

for other variables are conditioned on the preceding variables in the ordering. A sampling order is implied by an R-vine representation of the density if each conditional density can be written as a product of copula densities in the vine and one dimensional margins.

Not all sampling orders are implied by an R-vine representation of the density. Following sections show which sampling orders are implied by a given R-vine representation. First, the edging up algorithm is presented which samples an R-vine starting with an arbitrary variable.

4   Edging Up Sampling Algorithm

This section describes an algorithm for sampling a regular vine starting at an arbitrary element. It does not require the constant conditional copula assumption. This is also an algorithm for conditionalizing on a set of variables which is an initial segment of a sampling order. An illustration is better than an abstract description; we illustrate with the R-vine in Figure 2. The first two steps are shown in Figures 3 and 4. The edges are labeled in Figure 3 and omitted in subsequent steps of the algorithm. We start with five independent samples of a $[0, 1]$ uniform variable $u_1, u_2, \ldots, u_5$.

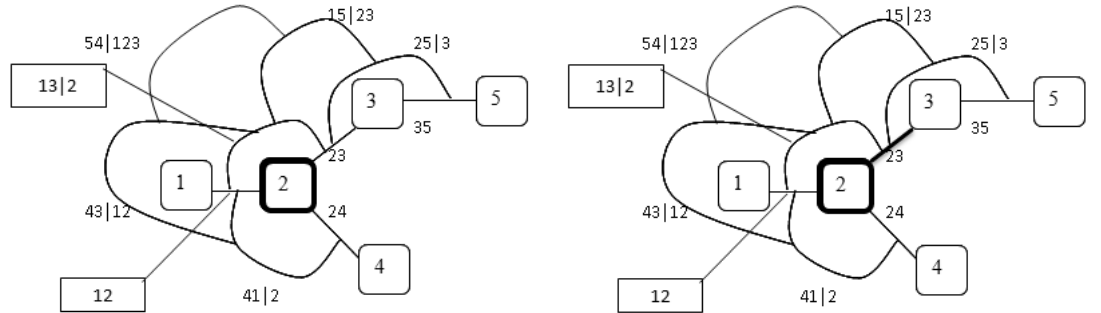1. Pick a variable, say 2 and sample it as:

$$x_2 = F_2^{-1}(u_2).$$

Figure 3: Edging up step 1

Figure 4: Edging up step 2

2. Pick an edge abutting the previously chosen element (see step 2 in Figure 4). A single element in the constraint set of the last chosen edge is un-sampled, sample it as:

$$x_3 = F_3^{-1} C_{3|2}^{-1}(u_3).$$

By proposition 2.1 this samples $(X_3|x_2)$. $x_3 = F_3^{-1}C_{3|2}^{-1}(u_3)$ is shorthand for $x_3 = F_3^{-1}C_{3|x_2}^{-1}(u_3)$. Similar shorthand notation is used below.

3. Pick an edge abutting the previously chosen element, the choice is shown in Figure 5:
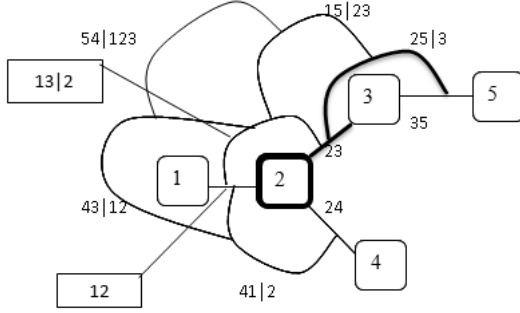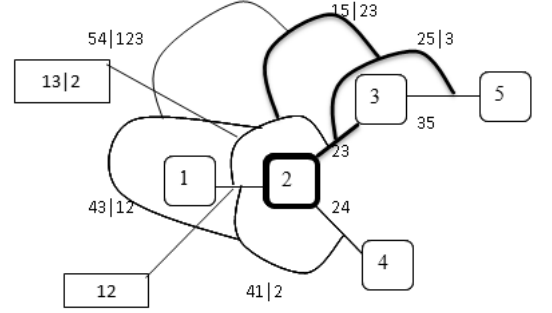

Figure 5: Edging up step 3

Figure 6: Edging up step 4

A single element in the constraint set of the last chosen edge is un-sampled, sample it as:

$$x_5 = F_5^{-1}C_{5|3}^{-1}C_{5|32}^{-1}(u_5).$$

By proposition 2.1 this samples $(X_5|x_3, x_2)$.

4. Pick an edge abutting the previously chosen element. A single element in the constraint set of the last chosen edge is un-sampled, sample it as:

$$x_1 = F_1^{-1}C_{1|2}^{-1}C_{1|32}^{-1}C_{1|532}^{-1}(u_1).$$

By proposition 2.1 this samples $(X_1|x_5, x_3, x_2)$.

5. Pick an edge abutting the previously chosen element. A single element in the constraint set of the last chosen edge is un-sampled, sample it as:

$$x_4 = F_4^{-1}C_{4|2}^{-1}C_{4|12}^{-1}C_{4|312}^{-1}C_{4|5312}^{-1}(u_4).$$

By proposition 2.1 this samples $(X_4|x_1, x_2, x_3, x_5)$.

For the R-vine in Figure 2 there are seven possible sampling orders starting with 2:

$$2 \to 3 \to 5 \to 1 \to 4$$
$$2 \to 3 \to 1 \to 5 \to 4$$
$$2 \to 3 \to 1 \to 4 \to 5$$
$$2 \to 1 \to 3 \to 4 \to 5$$
$$2 \to 1 \to 3 \to 5 \to 4$$
$$2 \to 1 \to 4 \to 3 \to 5$$
$$2 \to 4 \to 1 \to 3 \to 5.$$

8

An algorithm for generating implied sampling orders is presented in the next section.

5   Sampling Order and Merging Theorems

This section establishes a general result on the number of implied sampling orders for a regular vine. This is important when we need to conditionalize the distribution represented as a regular vine on observations. Conditionalizing such a distribution is very easy when the conditioning variables appear as an initial segment of a sampling order for this vine. To sample the conditional distribution it is enough to plug the values of the conditioning variables into the sampling algorithm. If there is no sampling order for this vine that keeps all conditioning variables as an initial segment one would need to transform the vine to another one. Such transformations are in general cumbersome.

Theorem 5.1. *Any regular vine on n variables implies 2^{n-1} sampling orders.*

Proof: The proof proceeds by establishing a bijection between implied sampling orders and increasing paths in a triangular array (an example is given in Figure 7). The array consists of n echelons. At the bottom are the individual variables. For $i = 2, \ldots, n$, echelon i corresponds to tree $i - 1$ in the R-vine, and shows the conditioned and conditioning sets of the copulas on tree $i - 1, i = 2, \ldots, n$. Draw lines representing the membership relations. Each application of the edging up algorithm corresponds to a unique increasing path in the triangular array: at each echelon there is one and only one variable in the conditioned set which is not present in a conditioned set lower on the path. The proof is concluded by counting the number of increasing paths in such a triangular array. Because each node in echelon i is an edge in tree $i - 1; i = 2, \ldots, n$, the number of upward edges leaving a node is the degree of the node in the vine, whereas the number of edges entering the bottom of each node above the first row is always two. All paths terminate in the top node (echelon n) through one of the two edges entering the top node. Each such path enters echelon $n - 1$ through one of two paths. Counting from the top node down, at echelon i the number of paths increases by a factor $2, i = n, \ldots, 2$. Hence the number of increasing paths is 2^{n-1}.□

With this proof in hand, it is easy to see that the following is a procedure for generating all sampling orders. The algorithm is as follows

- Choose one variable from the top node's conditioned variables, call it a_n.

- Truncate the vine by integrating a_n out. That is, remove all nodes containing a_n.

- The remaining sub-vine has two conditioned variables in its top node. Pick one call it a_{n-1}, and integrate it out etc.

The sequence of variables removed in this way in reverse order $(a_1, \ldots, a_n)$ is a sampling order. At each step except the last there were 2 choices, yielding the 2^{n-1} sampling orders.

The triangular array for the R-vine in Figure 2 and the implied sampling order used to illustrate the edging up algorithm are shown in Figure 7.
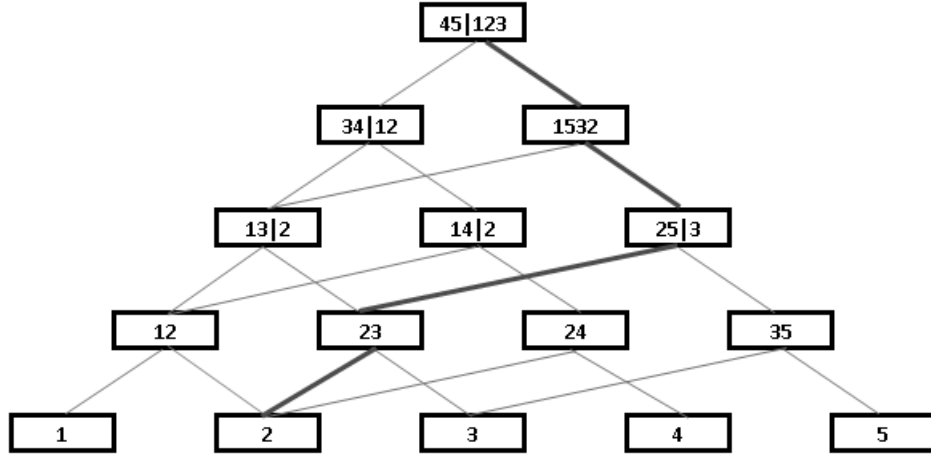

Figure 7: Triangular array for R-vine with sample order

It is easy to count the number of sampling orders emanating from each variable. In Figure 7 instead of writing the variables in each cell of the triangular array, write '1' in the top cell. In each other cell, write the sum of the numbers immediately above that cell. The results for the D-vine and R-vine in Figures 1 and 2 are shown in Figures 8 and 9 where the variables' indices are written below the cells n in the first row. A simple function counts the implied sampling orders starting with variable x. If t is the top cell, put $h(t) = 1$ and for cell i,

$$h(i) = \sum_{j \text{ directly above } i} h(j), \tag{5.1}$$

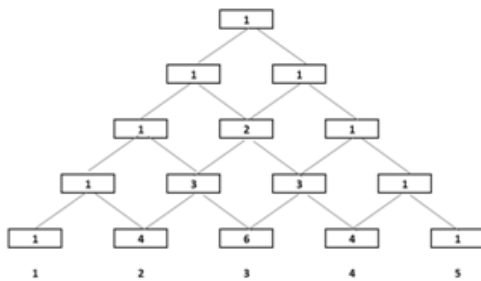where $h(x)$ is the number of sampling orders starting with variable x.

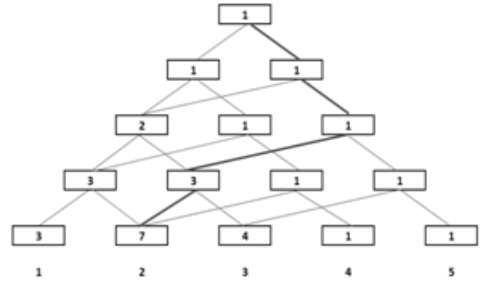Figure 8: Numbers of implied sample orders for D-vine of Figure 1

Figure 9: Numbers of implied sample orders for R-vine of Figure 2

We address a problem of merging two vines that are not overlapping into one larger vine. Merging may be useful if we need to combine vine models of distinct subsets of random variables. The number of possible such mergers is found. This result provides at once a shorter, more intuitive and more general result of the theorem in [6] stating the number of regular vines.

Definition 5.1 (Merger). *A regular vine $\mathcal{V}(n+m)$ with $n+m$ variables is a* merger *of regular vines $\mathcal{A}(n)$ on n variables and $\mathcal{B}(m)$ on m variables distinct from those in $\mathcal{A}(n)$ if $\mathcal{A}(n)$ and $\mathcal{B}(m)$ are sub-vines of $\mathcal{V}(n+m)$.*

Theorem 5.2. *For regular vines $\mathcal{A}(n)$ on n variables and $\mathcal{B}(m)$ on m variables distinct from those in $\mathcal{A}(n)$, there are 2^{n+m-2} mergers.*

Proof: Let $\mathcal{A}(n)$ have top conditioned variables a_n, a_{n-1}, and $\mathcal{B}(m)$ have top conditioned variables b_m, b_{m-1}. Suppose they are merged. The conditioned variables of the merged top node are one of a_n, a_{n-1} and one of b_m, b_{m-1}. Suppose they are a_n, b_m. Since a_n is conditioned with every other a_i, in $\mathcal{A}(n)$, which goes from echelon 1 to n, a_n must be conditioned with variables from vine $\mathcal{B}(m)$ in echelons $n+1, \ldots, n+m$.

Integrate out variable b_m. The top node in vine $\mathcal{B}(m)$ is removed leaving a $\mathcal{B}(m-1)$-subvine with a new top node. The top node of the new merged subvine has a variable from $\mathcal{B}(m-1)$ in its conditioned set, and this variable must occur in the new top node of the $\mathcal{B}(m-1)$-subvine. Call this variable b_{m-1}. In the new merged subvine b_{m-1} is partnered with a_n. Now integrate b_{m-1} out of the merged subvine. A new variable, call it b_{m-2}, appears conditioned with a_n and b_{m-2} in the resulting merged subvine.

Proceeding in this way, the sequence $(b_m, b_{m-1}, \ldots, b_1)$ occurs as top conditioned variables in successive subvines of $\mathcal{B}(m)$, and also as partners of a_n in decreasing echelons. It is a reverse sampling order for $\mathcal{B}(m)$. By parity of reasoning, the same holds for the sequence $(a_n, a_{n-1}, \ldots, a_1)$ which is a reverse sampling order of $\mathcal{A}(n)$.

Going back to the merged subvine after integrating out b_m, its top node's conditioned variables are a_n, b_{m-1}. Now integrate out successively the a_i's. By the above argument applied to this subvine, the sequence of partners of b_{m-1} obtained is a reverse sampling order. Denote this sampling order $(a_n, a_x, \ldots, a_y)$. We claim that $(a_n, a_x, \ldots a_y) = (a_n, a_{n-1}, \ldots, a_1)$. Indeed, the node with conditioned variables a_x, b_{m-1} must be a child of b_m, a_{n-1} from which it follows that $a_x = a_{n-1}$, and similarly for $b_{m-2}, b_{m-3}, \ldots, b_1$.

We have shown, for any i, that the partners of a_i are a reverse sampling order starting with the reverse sampling order $(b_m, b_{m-1}, \ldots, b_1)$ followed by the reverse sampling order of $\mathcal{A}(n)$ after integrating out $a_n, \ldots, a_{i+1}$. Mutatis mutandis for $\mathcal{B}(m)$. This shows that a merger of $\mathcal{A}(n)$ and $\mathcal{B}(m)$ defines a unique sampling order of each of $\mathcal{A}(n)$ and $\mathcal{B}(m)$. Also, given a sampling order of $\mathcal{A}(n)$ and of $\mathcal{B}(m)$, it is easy to verify that we can construct a merger by choosing partners of a_i and b_j from the sampling order of $\mathcal{B}(m)$ and $\mathcal{A}(n)$ respectively; whereby the conditioning variables are the variables of $\mathcal{B}(m)$ and $\mathcal{A}(n)$ which must be integrated out to render a given node a top node in a merged subvine. $\square$

For $m = 1$, this result says that there are 2^{n-1} extensions of a regular vine on n variables. [6] showed that for a naturally ordered regular vine on n elements, there are 2^{n-2} naturally ordered regular vines on $n+1$ elements which extend this regular vine. As there are two natural orderings, these two results agree. The present proof is much shorter. The number of naturally ordered regular vines on n variables is $2^{\binom{n-2}{2}}$, the number of regular vines on n variables is $\binom{n}{2}(n-2)!2^{\binom{n-2}{2}}$.

In the Table 1 eight possible mergers of two D-vines $2 - 1 - 3$ and $4 - 5$ are given. We present conditioned and conditioning sets of nodes in the mergers that do not belong to the D-vines.

NR	echelon 5	echelon 4	echelon 3	echelon 2
1	24|135	25|13, 34|15	35|1, 14|5	15
2	24|135	25|13, 14|35	15|3, 34|5	35
3	25|134	24|13, 35|14	34|1, 15|4	14
4	25|134	24|13, 15|34	14|3, 35|4	34
5	34|125	35|12, 14|25	15|2, 24|5	25
6	34|125	35|12, 24|15	25|1, 14|5	15
7	35|124	34|12, 15|24	14|2, 25|4	24
8	35|124	34|12, 25|14	24|1, 15|4	14

Table 1: Conditioning and conditioned sets of the mergers that do not belong to D-vines $2-1-3$ and $4 - 5$.

6   Plug in Conditionalization and Sample Order Proximity

We can conditionalize on any initial segment of any sampling order by simply plugging in values. Referring to the R-vine in Section 4, we could plug x_2, x_3 and x_1 into the edging up example and conditionalize on these three values. Suppose we want to conditionalize on variables 2 and 5; none of the sampling orders implied by this R-vine have $\{2, 5\}$ as an initial segment. To enable plug in conditionalizing we must transform the R-vine in such a way that variables 2 and 5 are contiguous. Consider the R-vine obtained by switching positions of variables 2 and 3. The sampling order $2 \to 5 \to 3 \to 1 \to 4$ becomes available. The conditional densities are

$$f_2, f_{5|2}, f_{3|5,2}, f_{1|3,5,2}, f_{4|1,3,5,2}.$$

Note that $f_{1|3,5,2}, f_{4|1,3,5,2}$ are already available as products of copula densities from the original R-vine; they don't need to be re-computed. To compute $f_{5|2}, f_{3|5,2}$ we apply the Thumb Rule to $2, 3, 5$:

$$c_{2,5;3}c_{3,5} = c_{5,3;2}c_{2,5}.$$

The terms on the left hand side are available from the original R-vine. We need to compute $c_{2,5}$ and then solve for $c_{5,3;2}$. From $c_{2,5}$ we obtain $f_{5|2}$ and $f_{3|5,2} = c_{5,3;2}c_{2,3}f_3$.

In searching for 'neighboring' R-vines to enable plug in conditionalization, or in searching the space of R-vines efficiently, it may be useful to have a notion of proximity.

Definition 6.1 (Sample order proximity). *The sample order proximity of two R-vines $\mathcal{A}(n), \mathcal{B}(n)$ on n variables is the number of common sampling orders implied by $\mathcal{A}(n)$ and $\mathcal{B}(n)$.*

Figure 10 shows for each of the 120 permutations of the 5 variables in the R-vine in section 4, the number of common sampling orders with the original R-vine. The permutations are shown as the result of permuting the lexigraphical order; thus $(1, 2, 3, 4, 5)$ is the identity permutation, $(1, 2, 3, 5, 4)$ is the permutation switching 5 and 4, etc.

For each permutation in Figure 10, the number of implied sampling orders is $2^{n-1} = 16$. The top permutation $(1, 2, 3, 4, 5)$ corresponding to the original R-vine has 16 common sample orders, other permutations have fewer. The total number of common sampling orders shown in Figure 10 is $2^{2(n-1)} = 2^8 = 256$, and this seems to hold for any regular vine and any n, though a proof is not available at this point. There are 52 permutations of the R-vine which have no common sampling orders with the initial R-vine. The following proposition gives a lower bound on the number of permutations with zero common sampling orders with a given R-vine, which does not depend on the R-vine being permuted.

Proposition 6.1. *Let $(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ be permutations of $(1, \ldots, n)$. Let $\mathcal{A}(a_1, \ldots, a_n)$ be an R-vine on n variables where the variables are ordered according to the highest echelon in which they appear as conditioned variables: $\{a_1, a_2\}$ are conditioned variables in the top node (echelon n), a_3 and (possibly) a_4 are conditioned variables in echelon $(n-1)$, etc. Let $\mathcal{B}(b_1, \ldots, b_n)$ be another R-vine on n variables, similarly ordered. Then the number of permutations $\pi \in n!$ for which $\mathcal{B}(\pi^{-1}(b_1), \ldots, \pi^{-1}(b_n))$ has no common sampling orders with $\mathcal{A}(a_1, \ldots, a_n)$ is greater or equal to $(n-2)(n-3)(n-2)! + 4(n-3)^2$.*

Proof: There can be no common sampling orders for permutations under which the conditioned variables in the top node of $\mathcal{B}$ do not occur in the top node of $\mathcal{A}(a_1, \ldots, a_n)$. This number is

$$\#\{\pi \in n! | \pi^{-1}(b_1) \notin \{a_1, a_2\}, \pi^{-1}(b_2) \notin \{a_1, a_2\}\} = (n-2)(n-3)(n-2)!$$

If a permutation preserves one of the top conditioned variables, say x, but has no common element in echelon $(n-1)$ for the node not containing x, then there can be no common sample orders. If a_3, a_4 are both in echelon $n-1$ then this number is

$$\#\{\pi \in n! | \pi^{-1}(b_1) \in \{a_1, a_2\}, \pi^{-1}(b_2) \notin \{a_1, a_2, a_4\}, \pi^{-1}(b_4) \notin \{a_1, a_2, a_4\}\} \geq 2(n-3)(n-3) \tag{6.1}$$

$$\#\{\pi \in n! | \pi^{-1}(b_2) \in \{a_1, a_2\}, \pi^{-1}(b_1) \notin \{a_1, a_2, a_3\}, \pi^{-1}(b_3) \notin \{a_1, a_2, a_3\}\} \geq 2(n-3)(n-3)$$

If a_4 is not in echelon $n-1$, then a_3 replaces a_4 in Equation 6.1, without affecting the number of permutations. □

In the case $n = 5$, the above estimate is exact, and the number of permutations of $\mathcal{B}$ having no common sampling orders with $\mathcal{A}$ is $36 + 8 + 8 = 52$. Figure 11 shows the number of common sampling orders with the original R-vine, for all permutations of a C vine on n variables. The distribution of numbers of common sampling orders is different from that in Figure 10, though the total numbers are the same as is the number of permutations with zero common sample orders.

The space of regular R-vines is very large. In searching for a vine which optimally fits a multivariate data set, heuristics are employed to restrict the search space. The sampling order proximity may provide a useful heuristic, as we may wish to restrict the initial search to vines which have a low number of common sampling orders in common. If we are searching to optimize some scalar function of R-vines and if this function is related to the vine structure as captured in sampling orders, then a low proximity search heuristic may yield good starting points for searches in a small number of neighborhoods.

7   Illustrative Example

We consider the financial data first analysed using vines in [13]. The data represent four time series of daily data: (1) the SSBWG hedged bond index, (2) the MSCI world stock index, (3) the Norwegian stock index (TOTX) and (4) the Norwegian bond index (BRIX), for the period from 04.01.1999 to 08.07.2003. Prior to fitting a vine to these data, the conditional mean is modelled using a first order autoregression and the conditional variance is modelled using a GARCH(1, 1) process. The remaining analysis is then run on the residuals. These initial steps follow exactly those given in [13].

The Kendall rank correlations between each of the variables are given in Table 2. We see that there are both positive and negative associations in the data, with the strongest being between the MSCI and TOTX indexes and SSBWG and BRIX indexes. Both of these relationships are positive.

	SSBWG(1)	MSCI(2)	TOTX(3)	BRIX(4)
SSBWG(1)	1.00	-0.16	-0.13	0.22
MSCI(2)	-0.16	1.00	0.33	-0.04
TOTX(3)	-0.13	0.33	1.00	-0.11
BRIX(4)	0.22	-0.04	-0.11	1.00

Table 2: The Kendall rank correlations between the 4 variables.

Initially we fit a D-vine to the 4 time series using the VineCopula [14] and CDVine [15] packages in R [5]. To do so we specify a structure and then bivariate copulas are selected from over 30 different possibilities for each relevant pair of variables by choosing the one with the

highest AIC value. The parameters of these copulas are estimated using maximum likelihood. The initial structure chosen is arbitrary in order to allow us to consider changes to the structure to improve the fit using the methods in the paper.

The chosen vine structure, bivariate copulas and fitted Kendall correlation values are given in Figure 12.

We see the structure of the vine. In the first tree the copula with the highest AIC between SSBWG and TOTX is the Normal copula, between TOTX and BRIX is the rotated 90 degrees Gumbel copula and between BRIX and MSCI is the Frank-copula. The t-copula is chosen between all variables in the second and third trees.

We can use the edging up algorithm to sample from the vine. To do so we use the method in Section 4. Suppose we wish to sample from the vine in the order (BRIX,TOTX,SSBWG,MSCI)= $(4, 3, 1, 2)$. Suppose we have sampled 4 uniform random variables (u_1, u_2, u_3, u_4). We can then use the edging up algorithm to sample;

$$
\begin{aligned}
x_4 &= F_4^{-1}(u_4), \\
x_3 &= F_3^{-1}\left(C_{3|4}^{-1}(u_3)\right), \\
x_1 &= F_1^{-1}\left(C_{1|3}^{-1}\left(C_{1|34}^{-1}(u_1)\right)\right), \\
x_2 &= F_2^{-1}\left(C_{2|4}^{-1}\left(C_{2|34}^{-1}\left(C_{2|134}^{-1}(u_2)\right)\right)\right).
\end{aligned}
$$

We take 1000 samples. To show that the edging up algorithm is retaining the vine structure we need to check that the marginal distributions of the four variables are still uniform and the Kendall correlations are being preserved. We give histograms of the four variables in Figure 13.

We see that all four of the samples from the marginal distributions of the variables show uniform distributions on $[0, 1]$. Thus, marginally, the edging up algorithm is doing a good job.

We consider the dependence between the different variables in the sample. Bivariate scatter plots between each pair of variables are given in Figure 14.

From the scatterplots we see the positive correlations between MSCI and TOTX and SSBWG and BRIX as remarked on the correlations table from the data (Table 2) and we also see negative correlations between SSBWG and TOTX and BRIX and TOTX.

The correlation between the simulated values for MSCI and TOTX is 0.35, between SSBWG and BRIX is 0.21 and between SSBWG and TOTX is -0.14. These values all show good agreement with the values from the data.

We consider the optimal selection of a vine structure for these data. The log-likelihood for the vine considered is 278.31 and the AIC is -538.63. A search of all of the possible D-vine structures, and performing the sequential estimation of the copulas within each vine using the VineCopula package, results in the optimal vine measured using both log-likelihood and AIC being that of order (4,3,2,1) in the first tree. The copulas in this vine are a mix of Gaussian,

t-, rotated Joe and rotated Gumbel. The log-likelihood of this optimal vine and the AIC are 292.56 and -567.13 respectively.

Let us return to the vine in Figure 12. There are $2^{4-1} = 8$ possible sampling orders for this vine. They are:

$$
\begin{array}{ccccccc}
2 & \to & 4 & \to & 3 & \to & 1 \\
4 & \to & 2 & \to & 3 & \to & 1 \\
4 & \to & 3 & \to & 2 & \to & 1 \\
4 & \to & 3 & \to & 1 & \to & 2 \\
3 & \to & 4 & \to & 2 & \to & 1 \\
3 & \to & 2 & \to & 4 & \to & 1 \\
3 & \to & 2 & \to & 1 & \to & 4 \\
4 & \to & 3 & \to & 2 & \to & 1 \\
\end{array}
$$

We wish to improve the fit of our vine. To do so we consider the permutations of the vine with a low number of sampling orders in common with this vine. There are two D-vines with 0 common sampling orders with this vine. The first tree of each of these vines, using the first letter as shorthand for each variable, along with the log-likelihood and AIC of each vine, are given in Table 3. The log-likelihood and AIC of the original vine and optimal vine are given below for comparison.

Vine	Log-likelihood	AIC
(3,1,2,4)	284.37	-550.74
(3,2,1,4)	292.22	-566.45
(2,4,3,1) Original	278.31	-538.63
(4,3,2,1) Optimal	292.56	-567.13

Table 3: The log-likelihood and AIC of each of the D-vines with no common sampling orders to the original vine.

We see that both of the vines with 0 sampling orders in common with the original vine have better fit, in terms of both log-likelihood and AIC, than the original vine. The second of these vines, with first tree (3,2,1,4), has a log-likelihood and AIC which are very close to those of the optimal vine: just 0.34 and 0.68 lower respectively. Thus by considering permutations with low proxoimity to the original vine we have found a vine which is very close to optimal. The optimal vine has just 2 sampling orders in common with the original vine.

The conjecture is that a "low proximity search" would give an efficient coarse grained way of searching the space of regular vines. This example is really too small to offer much support, indeed there are only 12 distinct $D-$vines on 4 variables. Nonetheless, we compare two search strategies, (1) choose 3 random permutations, compute the AIC of each and pick the minimum, and (2) pick one random permutation, compute its AIC and the AIC's of the two permutations with no common sampling order with the randomly chosen permutation, take the minimum. Using the data in Table 4 and a little simulation, we find that strategy (2) has a slightly better

chance (62%) of reaching a lower AIC. Table 4 gives a proximity matrix and the AIC values for each permutation. Of course, the efficacy of this heuristic depends on the relation between the scalar function we wish to optimize and the vine structure as captured in the proximity relation. Using the loglikelihood instead of the AIC would give no advantage to the proximity heuristic.

| Vine | 1324 | 1243 | 1324 | 1342 | 1423 | 1432 | 2134 | 2143 | 3124 | 2413 | 3214 | 2314 |
	-567.13	-563.17	-541.33	-538.63	-560.86	-547.73	-555.28	-556.23	-550.74	-552.27	-566.45	-543.27
1324	8	2	4	2	2	4	2	0	2	0	4	2
1243	2	8	2	4	4	2	0	2	4	2	2	0
1324	4	2	8	2	4	2	2	0	2	0	2	4
1342	2	4	2	8	2	4	4	2	0	2	0	2
1423	2	4	4	2	8	2	0	2	2	4	2	0
1432	4	2	2	4	2	8	2	4	0	2	0	2
2134	2	0	2	4	0	2	8	2	4	2	2	4
2143	0	2	0	2	2	4	2	8	2	4	4	2
3124	2	4	2	0	2	0	4	2	8	2	4	2
2413	0	2	0	2	4	2	2	4	2	8	2	4
3214	4	2	2	0	2	0	2	4	4	2	8	2
2314	2	0	4	2	0	2	4	2	2	4	2	8

Table 4: The AIC of each of the D-vines with no common sampling orders to the original vine.

8 Conclusions

Regular vines are a powerful modeling tool for data analysis and uncertainty propagation. To be used in "live decision support" regular vines must handle large data sets and must enable rapid conditionalization. This paper shows that there are promising lines of attack on these problems. Vine space grows very rapidly in the number of variables and the notion of sampling proximity may enable searching this space efficiently by stepping over vines that are close in this sense. Rapid conditionalization can be based on the edging up algorithm if the conditioned variables form an initial segment of an implied sample order. When that is not the case, the thumb rule may help limit the number of calculations required to place the conditioned variables in an initial segment of an implied sampling order.

References

[1] Harry Joe. Multivariate extreme-value distributions with applications to environmental data. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 22(1):pp. 47–64, 1994.

[2] R.M. Cooke. Markov and entropy properties of tree and vine-dependent variables. In *Proceedings of the ASA Section on Bayesian Statistical Science,*, 1997.

[3] Cooke R.M., Joe H., and Aas K. Vines arise. In *Dependence Modeling: Handbook on Vine Copulae*, 2010.

[4] Aas K.and Czado C.and Frigessi A. and Bakken H. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198, 2009.

[5] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2014.

[6] Oswaldo Moralees Napoles. *PhDF Thesis Bayesian Belief Nets and Vines in Aviation Safety and Other Applications.* Department of Mathematics, Delft University of Technology, Delft, 2009.

[7] T.J. Bedford and R.M. Cooke. Probability density decomposition for conditionally dependent random variables modelled by vines. *Annals of Mathematics and Artificial Intelligence*, 32(4):245–268, 2001.

[8] Bedford T.J. and Cooke R.M. Vines - a new graphical model for dependent random variables. *Ann. Statist*, 30:1031–1068, 2002.

[9] Harry Joe. *Dependence Modeling with Copulas.* Chapman & Hall, CRC., 2014.

[10] Jakob Stoeber, Harry Joe, and Claudia Czado. Simplified pair copula constructions, limitations and extensions. *Journal of Multivariate Analysis*, 119:101 – 118, 2013.

[11] I. Hobaek Haff, K. Aas, and A. Frigessi. On the simplified pair-copula construction - simply useful or too simplistic? *Journal of Multivariate Analysis*, 101:1296–1310, 2010.

[12] D. Kurowicka and R.M. Cooke. *Uncertainty Analysis with High Dimensional Dependence Modelling.* Wiley, 2006.

[13] K. Aas, K. C. Czado, A. Frigessi, and H. Bakken. Pair-copula constructions of multiple dependence. *Insurance, Mathematics and Economics*, 44:182–198, 2009.

[14] Ulf Schepsmeier, Jakob Stoeber, Eike Christian Brechmann, and Benedikt Graeler. *VineCopula: Statistical inference of vine copulas*, 2014. R package version 1.3.

[15] Eike Christian Brechmann and Ulf Schepsmeier. Modeling dependence with c- and d-vine copulas: The R package CDVine. *Journal of Statistical Software*, 52(3):1–27, 2013.

9   Appendix: Regular Vines

Definition 9.1 (Regular vine). *$\mathcal{V}$ is a regular vine on n elements if*

1. *$\mathcal{V} = \{T_1, \ldots, T_{n-1}\}$,*

2. *T_1 is a connected tree with nodes $N_1 = \{1, \ldots, n\}$, and edges E_1;*
   for $i = 2, \ldots, n-1$ T_i is a tree with nodes $N_i = E_{i-1}$.

3. **(proximity)** *for $i = 2, \ldots, n-1$, $\{a, b\} \in E_i, |a \triangle b| = 2$ where $\triangle$ denotes the symmetric difference and $|A|$ denotes the number of elements in set A.*

*A D-**vine** is a regular vine in which each node in each tree has degree at most 2; a C-**vine** is a regular vine in which each tree has one node of maximal degree, other nodes having degree 1.*

Definition 9.2 (constraint, conditioning, conditioned sets).

1. *For $e \in E_i, i \leq n-1$, the **constraint set** associated with e is the **complete union** U_e^* of e, that is, the subset of $\{1, \ldots, n\}$ reachable from e by the membership relation.*

2. *For $i = 1, \ldots, n-1, e \in E_i$, if $e = \{j, k\}$ then the **conditioning set** associated with e is*

$$D_e = U_j^* \cap U_k^*$$

   *and the **conditioned set** associated with e is*

$$\{C_{e,j}, C_{e,k}\} = \{U_j^* \setminus D_e, \ U_k^* \setminus D_e\}.$$

3. *The order of an edge $e \in E_j$ is the cardinality of its conditioning set, and is $j - 1$.*

4. *A node belongs to **echelon** ξ if ξ is the cardinality of its constraint set. Individual variables belong to the first echelon.*

Note that a node in tree k is an edge in tree $k-1$, $k > 1$; therefore referencing a node by its tree can be ambiguous. The echelon designation removes this ambiguity. With n variables, there are n echelons, the top echelon (echelon n) contains one element, namely the unique node in echelon n that is an unique edge in tree $n-1$. Note that for $e \in E_1$, the conditioning set is empty and its order is 0. For $e \in E_i, i \leq n-1, e = \{j, k\}$ we have $U_e^* = U_j^* \cup U_k^*$.

Definition 9.3 (child, descendent). *If node e is an element of node f, we say that e is an **child** of f; similarly, if e is reachable from f via the membership relation: $e \in e_1 \in \ldots \in f$, we say that e is an **descendent** of f.*

Lemma 9.1. *[12] For any node M of order $k > 0$ in a regular vine, if element i is a member of the conditioned set of M, then i is a member of the conditioned set of exactly one of the children of M, and the conditioning set of an child of M is a subset of the conditioning set of M.*

Definition 9.4 (parents, siblings)**.** *If element a occurs with element b as conditioned variables in echelon $k > 1$, then a and b are termed* **k-partners***. Nodes A and B are* **siblings** *if they are children of a common parent. A* **consanguineous grandchild** *of node $e \in E_i, i > 1$ is a node which is a child of two children of e.*

Definition 9.5 (natural order)**.** *A* **natural order** *of the elements of a regular vine $V(n)$ on n elements is a sequence of numbers $NO(V(n)) = (A_n, A_{n-1}, ..., A_1)$ where each A_i is an integer not greater than n obtained as follows: Take one conditioned element of echelon n of a regular vine and assign it position n; assign its n-partner position $(n-1)$. Element A_{n-1} occurs with an $(n-1)$-partner in the conditioned set of one child of the top node. Give this $(n-1)$-partner position $(n-2)$. The $(n-2)$ partner of element A_{n-2} is assigned position $(n-3)$. Iterate this process until all elements have been assigned a position. Position 1 is assigned to the $2-$partner of the variable in position 2.*

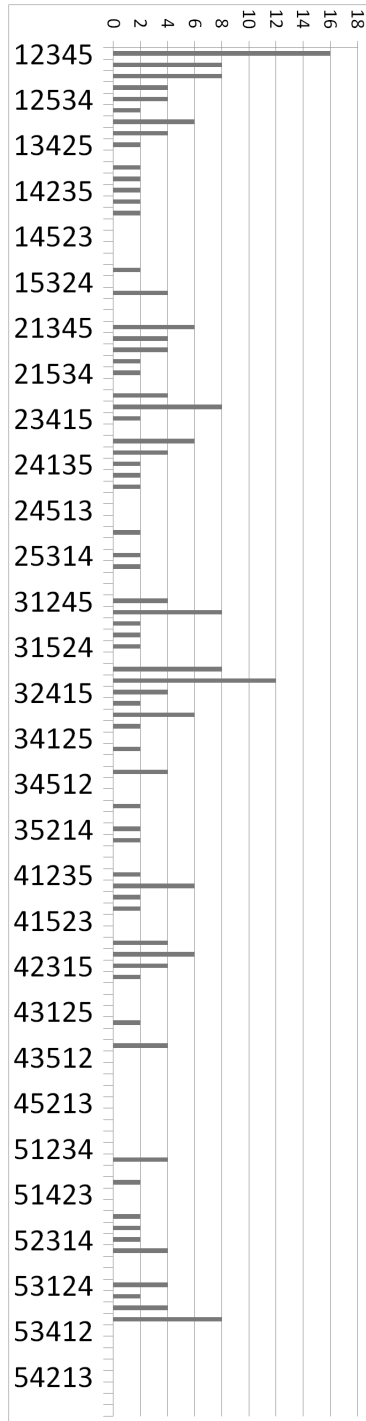Observe that there are two natural orders for every regular vine.

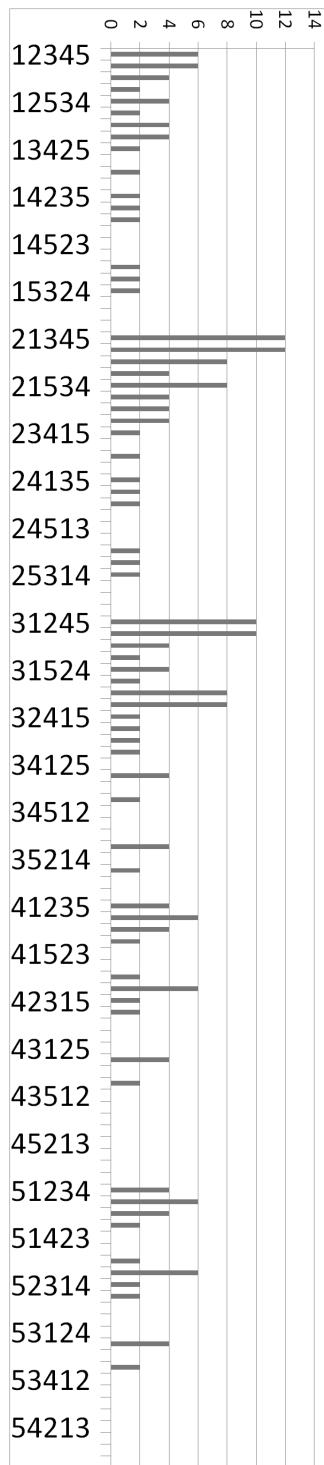Figure 10: Number of common sampling orders for each permutation of the R-vine

Figure 11: Number of common sampling orders with the R-vine for each permutation of a C-vine
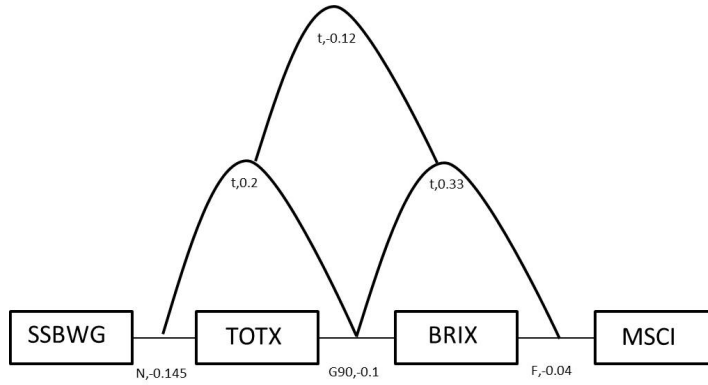
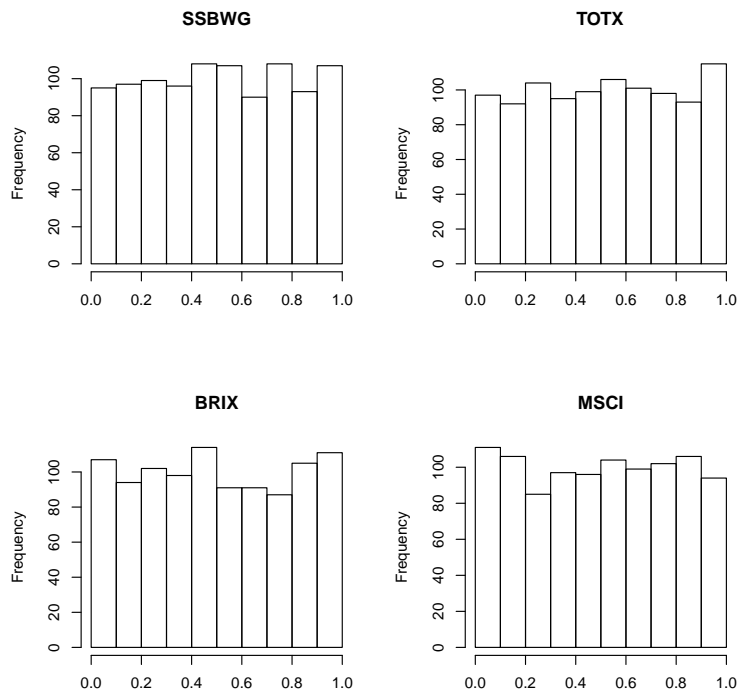Figure 12: The *D*-vine fitted to the Norwegian stock market data.

Figure 13: Histograms of the samples from the four variables showing the preservation of the marginal uniform distributions.
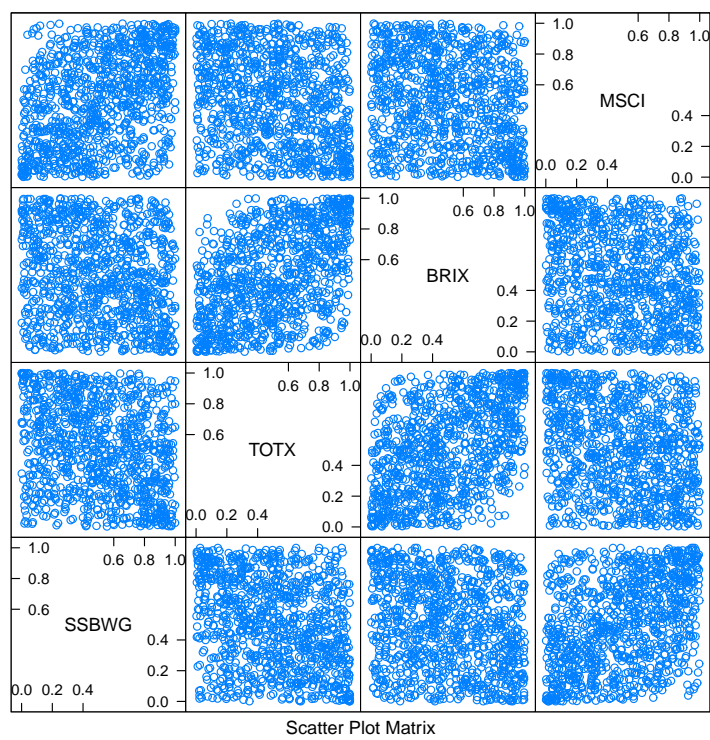
23

Figure 14: Bivariate Scatter plots between each of the four variables showing the dependence in the sample from the vine.