



New Prospects on Vines

Dominique Guegan, Pierre-André Maugis

► To cite this version:

| Dominique Guegan, Pierre-André Maugis. New Prospects on Vines. 2009. halshs-00348884v2

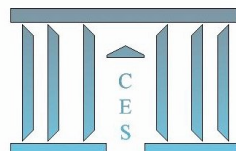
HAL Id: halshs-00348884

<https://shs.hal.science/halshs-00348884v2>

Submitted on 30 Nov 2009 (v2), last revised 27 May 2010 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



New Prospects on Vines

Dominique GUEGAN, Pierre André MAUGIS

2008.95

Version révisée



CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

New Prospects on Vines

D. Guegan^{*,1}, P.A. Maugis^{1,1}

^a*CES-MSE, Université Paris 1 Panthéon-Sorbonne, 106 boulevard de l'Hopital 75647
Paris Cedex 13, France*

Abstract

In this paper, we present a new method to use vine copulas that aims to make optimal use of their diversity. First, we introduce a new algorithm that can produce $\frac{n!}{2} \prod_{i=1}^{n-3} i!$ different vine copulas in dimension n . Then, to handle the ensuing computational burden of estimating all of them, we introduce a computationally efficient selection algorithm. Without such algorithms to build and select vine copulas, statistically efficient use of them would be computationally intractable, especially in higher dimensions. Another advantage of our methodology is that it offers great flexibility to the practitioner, offering a choice of the pair-copulas used, the selection test-used and the vine copulas used.

Key words: Vines, Multivariate copulas, Model Selection

1. Introduction

For almost ten years now, copulas have been used in econometrics and finance. They became an essential tool for pricing complex products, manag-

*Corresponding author, Tel: 33 (0) 1.44.07.82.93 - Fax: 33 (0) 1.44.07.83.01

URL: dguegan@univ-paris1.fr (D. Guegan), pmaugis@dpthmaths.ens-cachan.fr (P.A. Maugis)

¹This work has already been presented at the Sorbonne Finance Seminar (Paris) and at the MIT Econometric Seminar (Cambridge). We thank the participants of the seminars for their helpful comments. P.A. Maugis thank Arun Chandrasekhar, Victor Chernozhukov, Matthieu Latapy and Miriam Sofronia for their helpful comments.

ing portfolios and evaluating risks in banks and insurance companies. For instance, they can be used to compute VaR (Value at Risk) and ES (Expected shortfall). Moreover, copulas appear to be a very flexible tool, allowing for semi-parametric estimation, fast parameter optimisation and time varying parameters. These advantages make them a very interesting tool, although one major shortcoming is their use in high dimensions.

Recently, [Aas et al. \(2009\)](#) produced a seminal paper presenting a method to build high dimension copulas using pair-copulas as building blocks. Such decompositions are called vine copulas. To build these copulas, the authors recursively decomposed the dependence of the variables as done in [Joe \(1997\)](#) and [Bedford and Cooke \(2002, 2001\)](#). Another possible way to build multivariate copulas is to use "nested copulas." Here, we only consider vine copulas, which appear richer than "nested copulas" for multivariate analysis ([Berg and Aas, 2009](#)).

In this paper, we focus on building upon [Aas et al. \(2009\)](#)'s methodology by taking advantage of the diversity of vine copulas. A vine copula could be described as a decomposition of an n -variate vector's density into a combination of bivariate densities. Such decomposition is not unique, and many different vine copulas exist in each dimension. Previously cited papers use $n!$ decompositions in dimension n . We show through an example that each decomposition is linked to a specific dependence between the variables, and that more varied vine copulas could allow for the representation of more varied dependences. To increase the number of usable vine copulas, we developed a simple algorithm that produces $\frac{n!}{2} \prod_{i=1}^{n-3} i!$ different vine copulas. The size of the set of possible vine copulas makes use of an efficient selection algorithm necessary to find the best one among all of them in a reasonable time, as estimating and testing all of them would be computationally intractable. We present an algorithm that has the advantage of selecting the best vine copula within a set according to a predefined criterion without the requirement of having to test all the copulas, using an underlying lattice structure inside the set of vine copulas.

The paper is organised as follows: in Section 2, we introduce a new method to build vine copulas. Section 3 provides some characteristics of vine copulas that motivated our work. In Section 4, we describe the model selection procedure, relying on a lattice structure on the set of vine copulas. Section

5 presents a numerical application and Section 6 concludes.

2. The Vine Set

In this section, we introduce a new algorithm to build vine copulas. It consists of a step-by-step factorisation of the density function in a product of bivariate copulas. Our approach is different from the methods developed by [Aas et al. \(2009\)](#) and [Bedford and Cooke \(2002, 2001\)](#) and is not related to them in any obvious way. It has the advantage of being able to coherently describe a large set of vine copulas – $\frac{n!}{2} \prod_{i=1}^{n-3} i!$ in dimension n – while also being a simple recursive algorithm.

2.1. Formula

Let us consider a vector $\mathbf{X} = (X_1, X_2, \dots, X_n)$ of random variables characterised by a joint distribution function $F_{\mathbf{X}}$ that we assume has a density function $f_{\mathbf{X}}$. We introduce the following notations:

- $\mathbf{X}_{-\alpha} = (X_1, \dots, X_{\alpha-1}, X_{\alpha+1}, \dots, X_n)$ is the set of variables except the α -th.
- We denote f_{α} as the density of X_{α} . In the same fashion $f_{\alpha|\beta}$ is the density of $(X_{\alpha}|X_{\beta})$, $f_{-\alpha}$ is the density of $\mathbf{X}_{-\alpha}$ and $f_{\alpha|-\beta}$ is the density of $\mathbf{X}_{\alpha}|\mathbf{X}_{-\beta}$.
- $c_{\alpha,\beta|\gamma} = c_{X_{\alpha}, X_{\beta}|X_{\gamma}}(F_{X_{\alpha}|X_{\gamma}}(X_{\alpha}|X_{\gamma}), F_{X_{\beta}|X_{\gamma}}(X_{\beta}|X_{\gamma}))$ is the copula density of $(X_{\alpha}, X_{\beta}|X_{\gamma})$ as defined in [Sklar \(1959\)](#)'s theorem. Similarly $c_{\alpha,\beta|-\alpha,\beta} = c_{X_{\alpha}, X_{\beta}|\mathbf{X}_{-\alpha,\beta}}(F_{X_{\alpha}|\mathbf{X}_{-\alpha,\beta}}(X_{\alpha}|\mathbf{X}_{-\alpha,\beta}), F_{X_{\beta}|\mathbf{X}_{-\alpha,\beta}}(X_{\beta}|\mathbf{X}_{-\alpha,\beta}))$.

Our objective is to compute $c_{1,\dots,n}$, the copula density associated with the vector \mathbf{X} . This will be done by factorizing $f_{\mathbf{X}}$ in the following form:

$$f_{\mathbf{X}} = \prod_{i=1,\dots,n} f_i \cdot c_{1,\dots,n}.$$

By construction for $n = 2$ we have: $f_{\alpha,\beta} = f_{\alpha} \cdot f_{\beta} \cdot c_{\alpha,\beta}$. Using this property we consider the following factorisation of the joint density $f_{\mathbf{X}}$:

$$\forall \alpha, \beta \in \llbracket 1, n \rrbracket^2, \alpha \neq \beta$$

$$\begin{aligned} f_{\mathbf{X}} &= f_{-\alpha} \cdot f_{\alpha|-\alpha} = f_{-\alpha} \cdot \frac{f_{\alpha, \beta|-(\alpha, \beta)}}{f_{\beta|-(\alpha, \beta)}} = f_{-\alpha} \cdot \frac{f_{\alpha|-(\alpha, \beta)} \cdot f_{\beta|-(\alpha, \beta)}}{f_{\beta|-(\alpha, \beta)}} \cdot c_{\alpha, \beta|-\alpha, \beta} \\ &= \frac{f_{-\alpha} \cdot f_{-\beta}}{f_{-(\alpha, \beta)}} \cdot c_{\alpha, \beta|-(\alpha, \beta)}. \end{aligned} \tag{1}$$

This formula allows the computation of an n -variate density with a bivariate copula, two $(n-1)$ -and one $(n-2)$ -variate densities. Using this factorisation recursively, insuring that the denominators cancel at each step, we produce a factorisation of the n -variate density as a product of univariate and bivariate copula densities.

2.2. Example

In this example, we illustrate the unwinding of the previous algorithm for $n = 4$, providing the joint density function $f_{1,2,3,4}$. Our aim is to compute $c_{1,2,3,4}$: the joint copula density. We describe the two steps of the algorithm using the previous notations.

- First step:

$$f_{1,2,3,4} = \frac{f_{1,2,3} \cdot f_{1,2,4}}{f_{1,2}} \cdot c_{3,4|1,2}. \tag{2}$$

- Second step: we apply the relationship (1) to $f_{1,2,3}$ and $f_{1,2,4}$:

$$f_{1,2,3} = \frac{f_{1,2} \cdot f_{1,3}}{f_1} \cdot c_{2,3|1} = f_1 \cdot f_2 \cdot f_3 \cdot c_{1,2} \cdot c_{1,3} \cdot c_{2,3|1}, \tag{3}$$

and

$$f_{1,2,4} = \frac{f_{1,2} \cdot f_{2,4}}{f_2} \cdot c_{1,4|2} = f_1 \cdot f_2 \cdot f_4 \cdot c_{1,2} \cdot c_{2,4} \cdot c_{1,4|2}. \tag{4}$$

By merging formulas (2),(3) and (4), we obtain the following factorisation:

$$f_{1,2,3,4} = f_1 \cdot f_2 \cdot f_3 \cdot f_4 \cdot c_{1,3} \cdot c_{1,2} \cdot c_{2,4} \cdot c_{2,3|1} \cdot c_{1,4|2} \cdot c_{3,4|1,2}.$$

We have now factorised the density $f_{1,2,3,4}$ into a product of four univariate densities and six bivariate copula densities. By construction, this means that the copula density of (X_1, X_2, X_3, X_4) can be factorised as follows:

$$c_{1,2,3,4} = c_{1,3} \cdot c_{1,2} \cdot c_{2,4} \cdot c_{2,3|1} \cdot c_{1,4|2} \cdot c_{3,4|1,2}.$$

The unwinding of this algorithm can yield other vine copulas if other parameters are used. For instance the following formula is another possible vine copula:

$$c_{1,2,3,4} = c_{2,3} \cdot c_{3,4} \cdot c_{1,4} \cdot c_{2,4|3} \cdot c_{1,3|4} \cdot c_{1,2|3,4}.$$

3. Vine Copulas' Slip

Using the previous algorithm for an n -variate density, we can build $\frac{n!}{2} \prod_{i=1}^{n-3} i!$ different vine copulas in dimension n . In the following, by using a real data example, we show how each vine copula can yield different representations of the same dataset.

We estimate two different trivariate vine copulas from the same dataset and compare them graphically². The dataset is the daily Morgan Stanley evaluation of the French, German and British price indexes from 01/01/06 to 01/12/08 (Datastream). We denote them respectively as X , Y and Z . We use the same estimation procedure as describe in [Aas et al. \(2009\)](#). For each bivariate copula in the estimation procedure, we use the Gumbel copula. All computations are done in *MatLab*. In Figure 1, we plot the estimated copulas with uniform margins in three cases: $(X, Y|z = 0.1)$, $(X, Y|z = 0.5)$ and $(X, Y|z = 0.9)$. By examining this graph, one can see the importance of the choice of the vine copula in representing the "true" behaviour of the dataset.

Taking this into account, we decided to consider the whole set of possible vine copulas and selected the best one according to a specific criterion. However, as there are $\frac{n!}{2} \prod_{i=1}^{n-3} i!$ different vine copulas in dimension n , estimating and testing all of them is computationally intractable, thus we needed to develop an efficient search strategy. In the next section, we describe our construction of such an algorithm using a lattice structure.

²The two copulas used are: *Vine 1* = $c_{1,2} \cdot c_{1,3} \cdot c_{2,3|1}$ and *Vine 2* = $c_{1,3} \cdot c_{2,3} \cdot c_{1,2|3}$

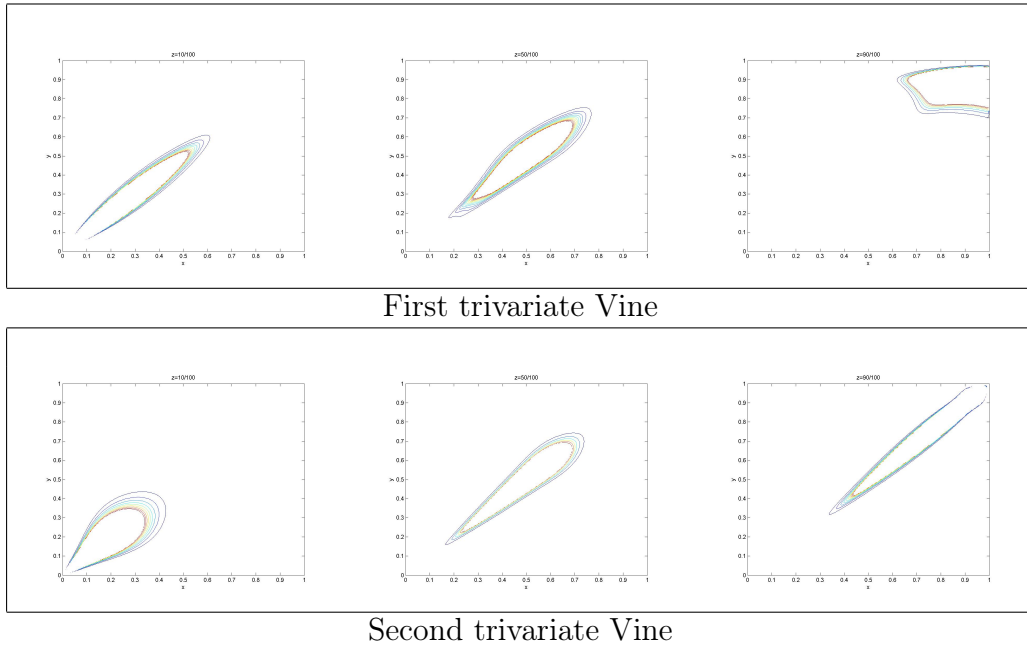


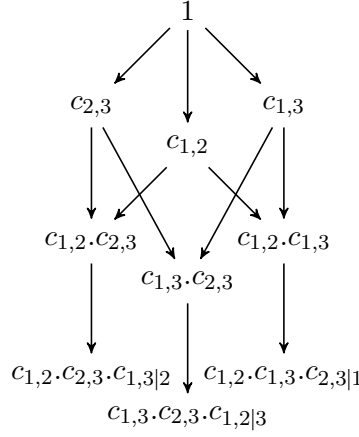
Figure 1: Two vines estimated on the same data set: slices at $z = 10\%$, 50% , 90%

4. Model Selection

In this section, we describe an efficient search algorithm that selects one among all possible vine copulas by introducing a lattice, which is a partial order on the set of vine copulas. We proceed in two steps: we first introduce the implemented lattice structure, and then we describe the search algorithm. We illustrate our approach with an example for $n = 3$.

The lattice structure we consider from the set of all possible vines is based on increasing independence hypotheses. For instance, in the trivariate case when the random variables $(X_1|X_2)$ and $(X_3|X_2)$ are independent, the vine copula $c_{1,2,3} = c_{1,2} \cdot c_{2,3} \cdot c_{1,3|2}$ becomes $c_{1,2,3} = c_{1,2} \cdot c_{2,3}$. Thus, we will consider $c_{1,2,3} = c_{1,2} \cdot c_{2,3}$ as being a vine copula and insert it into the lattice structure.

In dimension three we get the following lattice structure:



"1" is the root of the lattice, and the arrow points from a general vine copula to a more specific one. We say that the vine copula " $c_{1,3}$ " is more general than the vine copula " $c_{1,2} \cdot c_{1,3}$ " because the former vine copula defines a simpler relation between the variables and requires fewer parameters than the latter one. This kind of specification of the vine copulas is different from that introduced by [Czado et al. \(2009\)](#), which is based on a Bayesian selection framework.

We proceed now to the search algorithm based on the lattice structure in the general case. The use of lattices to organise large sets of models is not new, but lattice selection has never been used on vine copulas. [Gabriel \(1969\)](#) developed the principles and provided the theoretical groundwork for such a method. It relies on a test to decide whether a model is accepted or rejected (we specify a possible test in Section 5) and on the coherence hypothesis: "one ought not to accept a model while rejecting a more general model." Thus, when we test a model, if it is accepted we can accept all generalisations of this model, and if it is rejected we can reject all specifications of this model. In our case, the term "model" refers to vine copulas, and the coherence hypothesis could be restated as follows: "the gain in increased specification does not compensate for the loss of requiring more parameters."

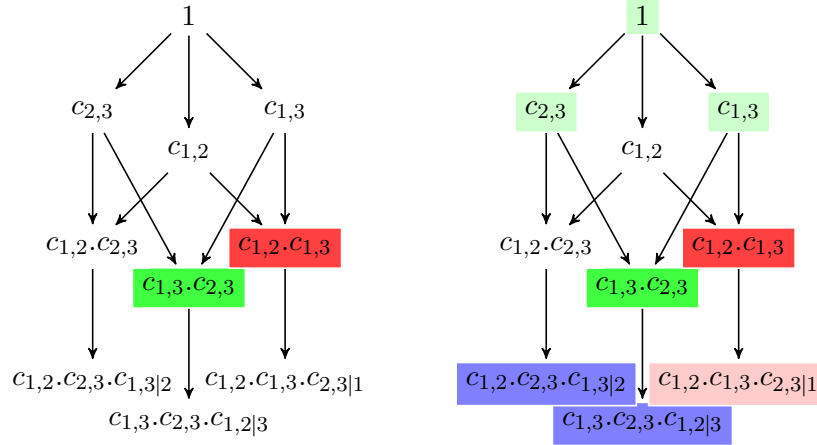
We now develop the search algorithm based on the coherence hypothesis. Within the set of models, it selects the subset composed of the most complex models accepted by the test [Edwards and Havranek \(1987\)](#):

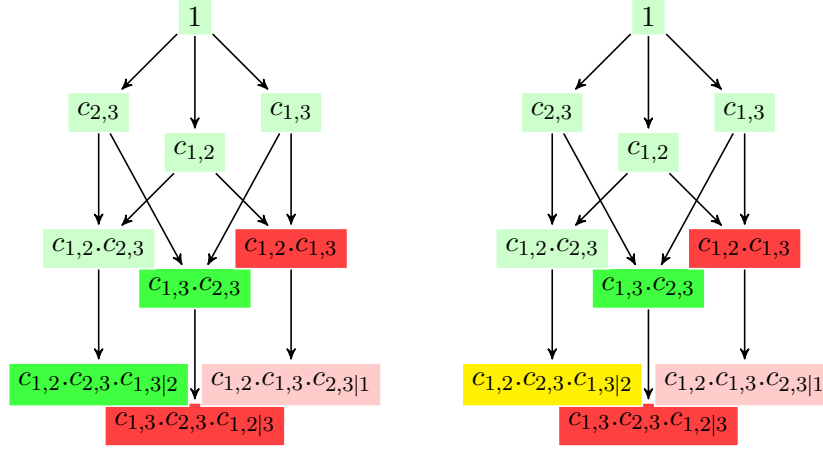
1. We randomly build a set S of possible models and classify them into

two sets according to the test: A , the accepted models and R , the rejected models.

2.
 - i If $A \neq \emptyset$: let D_A be the set composed of the most complex models that are not in A or R . If $D_A = \emptyset$ we stop; otherwise we test the models in D_A . If we do not retain any models, we stop. Otherwise, we update A and R accordingly and iterate.
 - ii If $A = \emptyset$: let D_R be the set composed of the most simple models that are not in A or R . If $D_R = \emptyset$ we stop; otherwise we test the models in D_R . If we accept all models, we stop. Otherwise, we update A and R accordingly and iterate.

We give a step-by-step illustration of the previous algorithm in the case of our trivariate example. First, two vine copulas are selected randomly to be tested: $c_{1,3} \cdot c_{2,3}$ and $c_{1,2} \cdot c_{1,3}$. The former one is accepted (in green) while the latter one is rejected (in red), as can be seen in the upper left diagram.





The consequence of $c_{1,3} \cdot c_{2,3}$ being accepted according to the coherence hypothesis is that three more vine copulas are accepted: 1, $c_{2,3}$ and $c_{1,3}$ (in weak green in the upper right diagram). Similarly, the consequence of $c_{1,2} \cdot c_{1,3}$ being rejected is that $c_{1,2} \cdot c_{1,3} \cdot c_{2,3|1}$ is also rejected (in weak red in the upper right diagram). The next step of the algorithm is to test the most complex vine copulas that are not accepted or rejected: $c_{1,3} \cdot c_{2,3} \cdot c_{1,2|3}$ and $c_{1,2} \cdot c_{2,3} \cdot c_{1,3|2}$ (in blue in the upper right diagram). The former copula is refused, and the latter one is accepted (in the lower left diagram). By the coherence hypothesis, we accept: $c_{1,2}$ and $c_{1,2} \cdot c_{2,3}$. In summary, the research algorithm keeps the most complex among the accepted vine copulas, which is $c_{1,2} \cdot c_{2,3} \cdot c_{1,3|2}$ (in yellow in the lower right diagram).

The advantage of this lattice structure is that it presents a high connectivity inside the set of all possible vine copulas. For instance, $c_{1,2}$ is connected directly or indirectly to five other vine copulas. This means that each step of the algorithm is very useful for the search, as it will allow us to accept or reject many other copulas by using the coherence hypothesis.

5. Application on the CAC40

Here we estimate the joint density of the five main assets composing the CAC40, the French leading index, using the methodology described above, and we use this estimated joint density to compute the *VaR*.

The dataset is taken from Datastream, daily quotes from Total, BNP - Paribas, Sanofi - Synthelabo, EDF - GDF and France Telecom from 25/4/08 to 21/11/08. This period is marked by the 2008 crisis, and our purpose is to test the resilience of our model to this shock. We estimate the parameters and select the vine copula based on the data ranging from 25/4/08 to 9/12/08, while the VaR is computed from the remaining dates.

Each univariate process is modelled with a $GARCH(p, q)$ model selected using the AIC criterion. We estimated the vine copula on the residual of the $GARCH(p, q)$ estimation using maximum likelihood (Aas et al., 2009). The parametric families used in order to estimate the bivariate copulas are the Clayton, Gumbel, Student and Gaussian copulas. The test we used is the second one described in Chen et al. (2004).

As we work in dimension five, there are 8536 vine copulas to choose from. Using our lattice-based algorithm in order to select the "best" vine copula, we only needed to test 1423 vine copulas (less than 17% of them). In fine, the retained vine copula is³:

$$c_{1,2,3,4,5} = c_{2,5} \cdot c_{4,5} \cdot c_{1,3} \cdot c_{1,5} \cdot c_{1,5|4} \cdot c_{1,2|5}.$$

Our final objective is to use this estimated vine copula density to compute the 10% VaR ⁴. We computed it from 9/12/08 to 21/11/08, and we plotted it in Figure 2. We compared it to a $GARCH(p, q)$ model-based estimate of the VaR computed directly on the $CAC40$ time series. To discriminate between the two approaches, we use the Kupiec test (Kupiec, 1995). The vine copula- VaR has a p -value of 0.96, so it is accepted as a true VaR , while the $GARCH$ - VaR has a p -value of 0.00 and is rejected according to this test⁵. Nevertheless, the vine copula approach fails to predict the major drop during the crisis, but the prediction remains solid before and after the crisis. These results make the vine copula methodology we described an interesting approach for risk management in order to estimate the multivariate density

³The computation took one hour on a 1.5Ghz processor computer.

⁴Given a random variable X , the 10% VaR of X is the value $VaR(X)$ such that:

$$P(X < VaR(X)) = 0.1.$$

⁵The number of failing: Vines: 5; $GARCH$: 21.

of a portfolio and to compute its associated VaR .

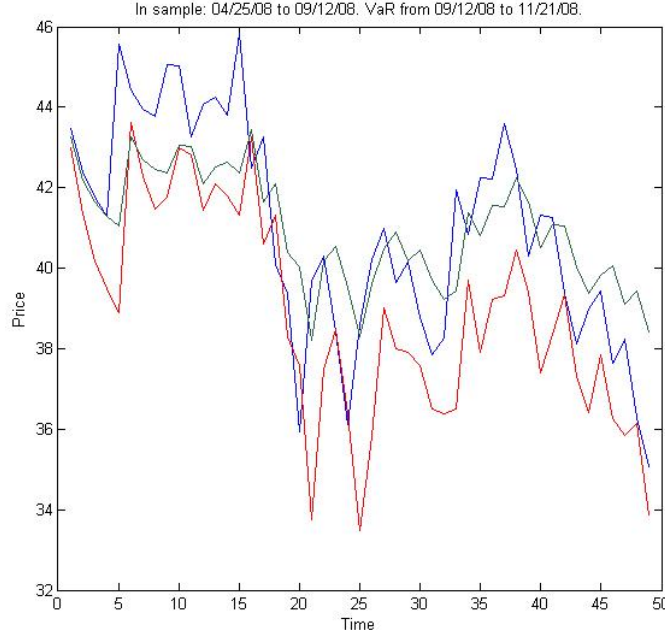


Figure 2: In Blue: The CAC40, In Red: Vine VaR Estimation, In Green: GARCH VaR Estimation.

6. Conclusion

In this paper, we estimated multivariate distributions in high dimensions using a new vine copula-based approach that builds upon the work of [Aas et al. \(2009\)](#) by taking advantage of the diversity of vine copulas. We introduced an algorithm utilising a large number of vine copulas from which we were able to retain the "best" one according to a predefined test, using a lattice structure, which makes the research simple and fast. The methodology we developed offers great flexibility to the practitioners in terms of the choice of the set of vine copulas, choice of bi-variate copulas, usefulness of the lattice structure, and the decision rule for the selection procedure.

It appears necessary to have a great number of vine copulas to be able to take into account the most important features of the dataset to be anal-

ysed: indeed, through a simple example we have highlighted the importance of finding the vine copula that takes into account most of the information contained in the dataset, particularly the behaviour of the tails. Finally, the *VaR* computation based on our methodology provides interesting results compared with the classical parametric approach.

However, some questions remain. First is the problem of devising a test to decide whether a vine copula represents a dataset dependence correctly or not. [Chen et al. \(2004\)](#) and [Chen and Fan \(2006\)](#) produced interesting tests, but for an optimum use of vine copulas a simpler and more efficient test needs to be built. Another question concerns the quality of the estimates, which we did not focus on in this paper. Instead, we applied a classical maximum likelihood estimation approach as developed for instance in [Aas et al. \(2009\)](#) to estimate the vine copula parameters. Results on the efficiency of the estimates have to be established. We address this question in a companion paper [Guégan and Maugis \(2009\)](#).

References

- Aas, K., Czado, C., Frigessi, A., Bakken, H., 04 2009. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics* 44, 182–198.
- Bedford, T., Cooke, R., 2001. Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial Intelligence* 32, 245–268.
- Bedford, T., Cooke, R., 2002. Vines: A New Graphical Model for Dependent Random Variables. *The Annals of Statistics* 30 (4), 1031–1068.
- Berg, D., Aas, K., 2009. Models for construction of multivariate dependence. Forthcoming in *The European Journal of Finance*.
- Chen, X., Fan, Y., 2006. Estimation of copula-based semi-parametric time series models. *Journal of Econometrics* 130, 307–335.
- Chen, X., Fan, Y., Patton, A., 2004. Simple Tests for Models of Dependence Between Multiple Financial Time Series, with Applications to U.S. Equity

- Returns and Exchange Rates. London Economics Financial Markets Group Working Paper (483), london, UK.
- Czado, C., Gartner, F., Min, A., 2009. Analysis of Australian electricity loads using joint Bayesian inference of D-Vines with autoregressive margins. Zentrum Mathematik Technische Universitat Munchen: Working PaperMunchen, Germany.
- Edwards, D., Havranek, T., 1987. A Fast Model Selection Procedure for Large Families of Models. *Journal of the American Statistical Association* 82 (397), 205–213.
- Gabriel, K. R., 1969. Simultaneous Test Procedures - Some Theory of Multiple Comparisons. *Annals of Mathematics Statistics* 40 (1), 224–250.
- Guégan, D., Maugis, P. A., 2009. Dealing With Vines Issues. Work in Progress.
- Joe, H., 1997. *Multivariate Models and Dependence Concepts*. Chapman & Hall, London, UK.
- Kupiec, P. H., 1995. Techniques for verifying the accuracy of risk measurement models. Board of Governors of the Federal Reserve System (U.S.), Washington (95-24).
- Sklar, A., 1959. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de L' Université de Paris* 8, 229–231.