"New prospects on vines"

AUTHORS	P.A. Maugis D. Guégan
ARTICLE INFO	P.A. Maugis and D. Guégan (2010). New prospects on vines. *Insurance Markets and Companies*, *1*(1)
RELEASED ON	Wednesday, 26 May 2010
JOURNAL	"Insurance Markets and Companies"
FOUNDER	LLC "Consulting Publishing Company "Business Perspectives"

NUMBER OF REFERENCES	NUMBER OF FIGURES	NUMBER OF TABLES
0	**0**	**0**

Pierre-Andre Maugis (France), Dominique Guegan (France)

New prospects on vines

Abstract

In this paper, we present a new methodology based on vine copulas to estimate multivariate distributions in high dimensions, taking advantage of the diversity of vine copulas. Considering the huge number of vine copulas in dimension n, we introduce an efficient selection algorithm to build and select vine copulas with respect to any test T. Our methodology offers a great flexibility to practitioners to compute *VaR* associated to a portfolio in high dimension.

Keywords: vines, multivariate copulas, model selection.

Introduction

For almost ten years now, copulas have been used in econometrics and finance. They became an essential tool for pricing complex products, managing 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) (Artzner et al., 1997). Moreover, copulas appear to be a very flexible tool, allowing for semi-parametric estimation, fast parameter optimization and time varying parameters. These advantages make them a very interesting tool, although one major shortcoming is their use in high dimensions. Indeed, elliptical copulas can be expended to higher dimension, but they are unable to represent financial tail dependences (Patton, 2009), and the Archimedean copulas are not satisfactory as models to describe multivariate dependence in dimensions higher than 2 (Joe, 1997). In this paper, we present a solution to this problem by using the vine copulas.

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. This last notion is defined as follows: if $C_1(.,.)$ and $C_2(.,.)$ are two bivariate copulas, then $C_1(C_2(.,.),.)$ is a trivariate nested copula. In this paper, we only consider vine copulas, which appear richer than nested copulas for multivariate analysis, following Berg and Aas (2009), who show that nested copulas are less efficient than vine copulas to estimate densities and risk measures such as the *VaR*.

Our purpose is to build all possible vine copulas in order to capture as much information as possible from a dataset. Indeed, a practitioner aims at finding the vine copula that characterizes correctly the behavior of the dataset under study: we call this true

copula C^0. However, finding C^0 is almost impossible for any real dataset. On the other hand, a vine copula could be described as a decomposition of an n-variate vector's density using bivariate copula densities as building blocks. Such vine decomposition is not unique, and many different vine copulas exist in each dimension. Through an example in Section 3 we show that, given a dataset, each vine decomposition estimates specific dependence between the variables.

This result highlights the importance of defining a strategy to find the vine copula that is closest to the true copula C^0, according to any criterion retained by the practitioner. To do so we generated a large number of vine copulas using a simple algorithm that produces $N = n!/2 \prod_{i=1}^{n-3} i!$ different vine copulas in dimension n. Previously cited papers use $n!$ vine copulas, but as our purpose is to find a copula closest to the copula C^0, increasing the number of vine copulas under consideration is crucial. The size of the set of possible vine copulas necessitates a computationally efficient selection algorithm. We present an algorithm that has the advantage of selecting the best vine copula within a set of vine copulas, according to any criterion chosen by the practitioner without the requirement of having to test all the copulas. This algorithm is based on an underlying lattice structure inside the set of vine copulas.

The paper is organized as follows: in Section 1, we introduce a new method to build vine copulas. Section 2 provides some characteristics of vine copulas that motivated our work. In Section 3, we describe the model selection procedure, relying on a lattice structure on the set of vine copulas. Section 4 presents a numerical application and the last section concludes.

1. The vine set

In this section, we introduce a new algorithm to build vine copulas. It consists in a step-by-step factorization of the density function in a product of bivariate copulas. Our approach is different from the

methods developed by Bedford and Cooke (2002, 2001) and is not related to them in any obvious way. Indeed, in Bedford and Cooke (2002, 2001) vine copulas are introduced as decomposition of the multivariate random vector density based on a type of graph structured called "vines", from which their denomination comes. Our approach has the advantage of being able to coherently describe a large set of vine copulas while also being a simple recursive algorithm.

1.1. Formula. Let us consider a vector $X=(X_1, X_2,..., X_n)$ of random variables characterized by a joint distribution function F_x that we assume has a density function f_x. We introduce the following notations:

- $X_{-\alpha} = (X_1,..., X_{\alpha-1}, X_{\alpha+1},...,X_n)$ is the set of variables except the α-th.

- We denote f_α 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 $X_{-\alpha}$ and $f_{\alpha|-\beta}$ is the density of $X_\alpha|X_{-\beta}$.

- $c_{\alpha,\beta|\gamma} = c_{X_\alpha, X_\beta|X_\gamma}\left(F_{X_\alpha|X_\gamma}\left(X_\alpha \mid X_\gamma\right), F_{X_\beta|X_\gamma}\left(X_\beta \mid X_\gamma\right)\right)$ is the copula density of $(X_\alpha, X_\beta|X_\gamma)$ as defined in Sklar's (1959) theorem. Similarly: $c_{\alpha,\beta|-\gamma} = c_{X_\alpha, X_\beta|X_{-\gamma}}\left(F_{X_\alpha|X_{-\gamma}}\left(X_\alpha \mid X_{-\gamma}\right), F_{X_\beta|X_{-\gamma}}\left(X_\beta \mid X_{-\gamma}\right)\right)$.

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

$$f_X = \prod_{i=1,...n} f_i \cdot c_{1,...,n}.$$

By construction for $n = 2$ we have: $f_{\alpha,\beta} = f_\alpha f_\beta . c_{\alpha,\beta}$. Using this property we consider the following factorization of the joint density f_X: $\forall \alpha, \beta \in \{1,...,n\}^2, \alpha \neq \beta$.

$$f_X = f_{-\alpha} \cdot f_{\alpha|-\alpha} = f_{-\alpha} \cdot \frac{f_{\alpha,\beta|-(\alpha,\beta)}}{f_{\beta|-(\alpha,\beta)}} =$$

$$= f_{-\alpha} \frac{f_{\alpha|-(\alpha,\beta)} \cdot f_{\beta|-(\alpha,\beta)}}{f_{\beta|-(\alpha,\beta)}} \cdot c_{\alpha,\beta|-(\alpha,\beta)} = \frac{f_{-\alpha} \cdot f_{-\beta}}{f_{-(\alpha,\beta)}} \cdot c_{\alpha,\beta|-(\alpha,\beta)}. \quad (1)$$

Formula (1) allows the computation of an n-variate density with a bivariate copula, two $(n-1)$- and one $(n-2)$-variate densities. Using this factorization recursively, insuring that the denominators cancel at each step, we produce a factorization of the n-variate density as a product of univariate and bivariate copula densities. Using this algorithm we can produce all N possible vine copulas (Napoles, 2007)[1,2].

---

[1] N is the number of "vine" type graphs with n nodes, which is also the number of vine copulas (see Bedford and Cooke, 2002, 2001). The proof of the formula relies heavily on the graph structure of vines.
[2] Our algorithm can produce more varied decomposition, however, we do not consider them as they are not efficient estimators (Bedford and Cooke, 2002, 2001).

1.2. 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:

$$f_{1,2,3,4} = \frac{f_{1,2,3} \cdot f_{1,2,4}}{f_{1,2}} \cdot c_{3,4|1,2} = \frac{f_{1,2,3} \cdot f_{1,2,4}}{f_1 \cdot f_2 \cdot c_{1,2}} \cdot c_{3,4|1,2}. \quad (2)$$

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

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

$$= f_1 \cdot f_2 \cdot f_3 \cdot c_{1,2} \cdot c_{1,3} \cdot c_{2,3|1}, \quad (3)$$

and

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

$$= f_1 \cdot f_2 \cdot f_4 \cdot c_{1,2} \cdot c_{2,4} \cdot c_{1,4|2}. \quad (4)$$

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

$$f_{1,2,3,4} = f_1 \cdot f_2 \cdot f_3 \cdot f_4 \cdot c_{1,3} \cdot c_{1,2} \cdot c_{2,4} \cdot c_{2,3|1} \cdot c_{1,4|2} \cdot c_{3,4|1,2}. \quad (5)$$

We have now factorized 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 factorized 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 copula density $c_{1,2,3,4}$ has also the following decomposition:

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

2. The vine copula estimator

Using the previous algorithm for an n-variate density, we can construct the N different vine copulas in dimension n. In the following, using a real dataset as example, we show how each vine copula can yield different representations of the same dataset. We proceed in the following way.

We estimate two different trivariate vine copulas from the same dataset and compare the tail dependence of the two estimated densities by observing in which quadrant of $[0, 1]^3$ the density

evolves[1]. 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 as X, Y and Z, respectively. In a first step, for X, Y and Z, we estimate separately GARCH (1, 1) models using pseudo-maximum likelihood methods (Ghysels et al., 1995). In a second step, using the residuals, we estimate the parameters of the vine copula using maximum likelihood (Chen and Fan, 2006). For each bivariate copula in the estimation procedure, we select among the Gaussian, Student, Clayton and Gumbel copulas, the one with higher likelihood. We choose these copulas because they characterize the main features detected in economic and financial time-series (Patton, 2009)[2]. 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 $(X,Y|z=0,9)$ (from left to right).

By examining Figure 1, we observe that the two vine copulas describe different tail dependences.

According to the first vine copula, there is strong upper tail dependence (top-right plot) but no lower dependence (top-left plot). However, the second vine copula describes strong lower tail dependence (bottom left plot) but no upper tail dependence (bottom right plot). This difference in terms of tail dependence is solely caused by the vine copula, since in both cases we used the same estimation procedure and the same dataset.

Taking this fact into account and to attain our purpose of finding the copula that is closest to the true one C^0, according to a criterion chosen by the practitioner, we decided to consider the whole set of possible vine copulas. However, as there are N 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 such an algorithm using a lattice structure. We denote T the test retained by the practitioner to make the selection.
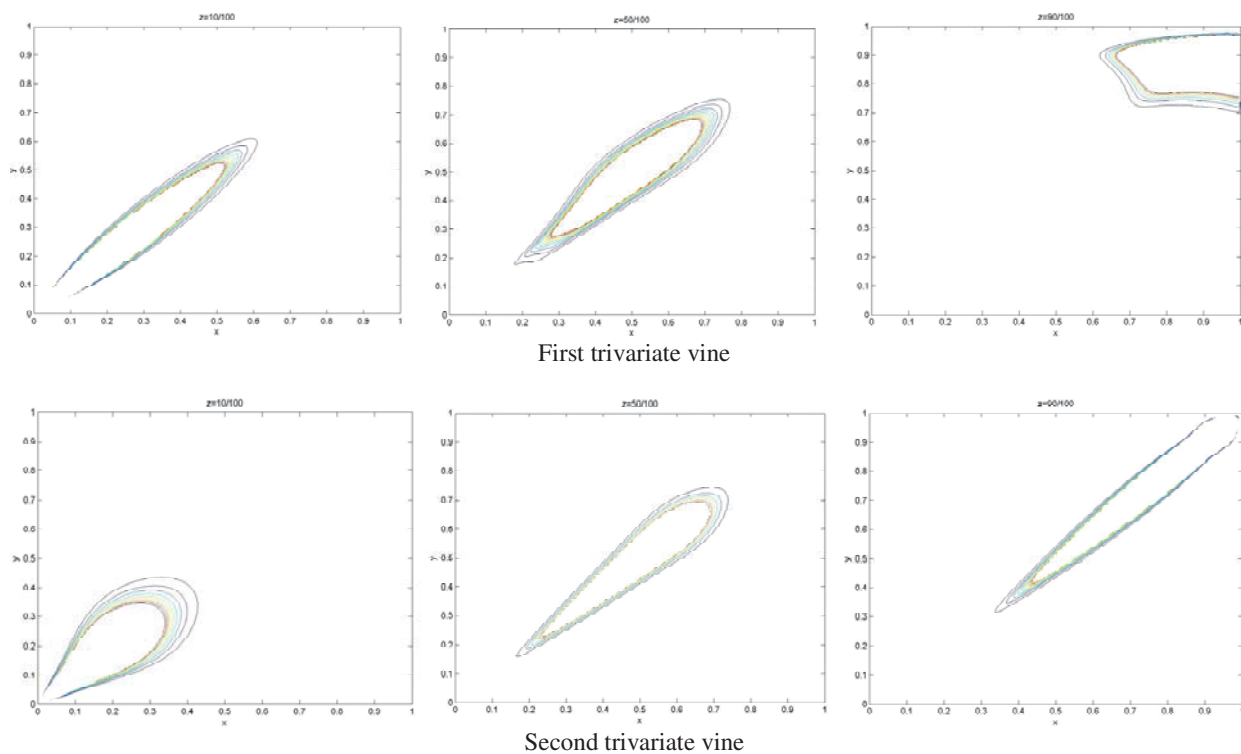

First trivariate vine

Second trivariate vine

Fig. 1. Two vines estimated on the same data set: slices at z = 10%, 50%, 90%

3. Model selection

In this section, we describe the search algorithm that finds the set of vine copulas accepted by the test T among all possible vine copulas. To this end, we introduce a lattice which is a partial order on the set of vine copulas. We proceed in two steps: we first introduce lattice theory and the implemented lattice structure, and then we describe the search algorithm. We illustrate our approach with an example in dimension $n = 3$.

Lattice theory: A lattice is a set and a partial order. A partial order is a binary relation, for instance, denoted $\leq$, that is reflexive, antisymmetric and transitive, i.e for all a, b, and c, we have that:

♦   a ≤ a (reflexivity);

♦   if a ≤ b and b ≤ a then a = b (antisymmetric);

---

[1] The two copulas used are: *Vine* 1 = $c_{1,2}$, $c_{1,3}$, $c_{2,3}$ and *Vine* 2 = $c_{1,3}$, $c_{2,3}$, $c_{1,2|3}$
[2] All computations are done in *MatLab*.

♦ if a ≤ b and b ≤ c then a ≤ c (transitive).

For instance, the real numbers with binary relation ≤ is a lattice.

The lattice structure we build 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[1].

In dimension three we get the following lattice structure:
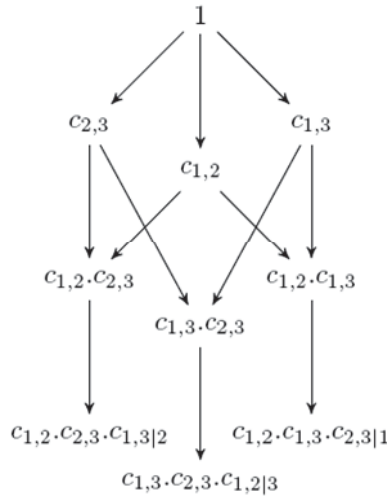
"1" is the root of the lattice, and the arrow points from a general vine copula to a more specific one. We say that the vine copula "$c_{1,3}$" is more general than the vine copula "$c_{1,2} \cdot c_{1,3}$" because the former vine copula defines a simpler relation between the variables and requires fewer 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 construct the search algorithm based on the lattice structure in the general case. The use of lattices to organize 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 two tools, first, a test T to decide whether a model is accepted or rejected[2], second, on the rule of coherence: "one ought not to accept a model while rejecting a more general model". When we apply the test T on a model, if the null of the model being the true model is retained then we accept all generalizations of this model, and if the null is rejected we reject all specifications of this model. In our case, the term "model" refers to vine copulas, and the rule of coherence could be restated as follows: "the gain in increased specification does not compensate for the loss of requiring more parameters".
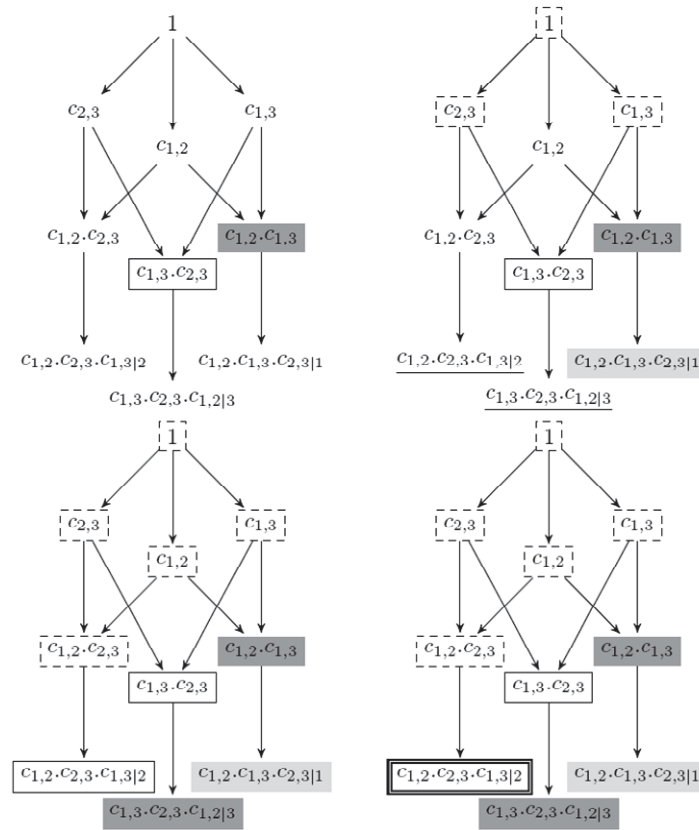
We now develop the search algorithm based on the rule of coherence. Within the set of models, it selects the subset of all possible vine copulas composed of the most complex models accepted by a test T:

1. We randomly build a set S of possible models and classify them into two sets according to the test T: A, the set of accepted models, and R, the set of 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 $D_A \neq \emptyset$ we stop; otherwise, we test with T the models in D_A. If we do not retain any model, we stop. Otherwise, we update A and R accordingly and iterate.
   ii) If $A \neq \emptyset$: let D_R be the set composed of the most simple models that are not in A or R. If $D_R \neq \emptyset$ we stop; otherwise, we test with T the models in D_R. If we accept all models, we stop. Otherwise, we update A and R accordingly and iterate.

Edwards and Havranek (1987) study a similar algorithm and prove that the algorithm reaches its purpose efficiently. This means that given a set of models of size p it will only require $o(log(p))$ steps to find the optimum.

We give a step-by-step illustration of how the previous algorithm could unwind in the case of our trivariate example. Let X be a trivariate dataset and T a test. 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 is accepted (solid line) while the latter is rejected (in black) by the test T, as can be seen in the upper left diagram.

---

[1] This implies that the number of copulas we consider is greater than *N*.
[2] We do not specify *T* in this section, and instead describe the procedure for any test chosen by the practitioner, but we specify a possible test in Section 4.

$$1$$
$$c_{2,3} \qquad c_{1,3}$$
$$c_{1,2}$$
$$c_{1,2}\cdot c_{2,3} \qquad c_{1,2}\cdot c_{1,3}$$
$$c_{1,3}\cdot c_{2,3}$$
$$c_{1,2}\cdot c_{2,3}\cdot c_{1,3|2} \qquad c_{1,2}\cdot c_{1,3}\cdot c_{2,3|1}$$
$$c_{1,3}\cdot c_{2,3}\cdot c_{1,2|3}$$

Upper right diagram:

$$1$$
$$c_{2,3} \qquad c_{1,3}$$
$$c_{1,2}$$
$$c_{1,2}\cdot c_{2,3} \qquad c_{1,2}\cdot c_{1,3}$$
$$c_{1,3}\cdot c_{2,3}$$
$$\underline{c_{1,2}\cdot c_{2,3}\cdot c_{1,3|2}} \qquad c_{1,2}\cdot c_{1,3}\cdot c_{2,3|1}$$
$$\underline{c_{1,3}\cdot c_{2,3}\cdot c_{1,2|3}}$$

Lower left diagram:

$$1$$
$$c_{2,3} \qquad c_{1,3}$$
$$c_{1,2}$$
$$c_{1,2}\cdot c_{2,3} \qquad c_{1,2}\cdot c_{1,3}$$
$$c_{1,3}\cdot c_{2,3}$$
$$c_{1,2}\cdot c_{2,3}\cdot c_{1,3|2} \qquad c_{1,2}\cdot c_{1,3}\cdot c_{2,3|1}$$
$$c_{1,3}\cdot c_{2,3}\cdot c_{1,2|3}$$

Lower right diagram:

$$1$$
$$c_{2,3} \qquad c_{1,3}$$
$$c_{1,2}$$
$$c_{1,2}\cdot c_{2,3} \qquad c_{1,2}\cdot c_{1,3}$$
$$c_{1,3}\cdot c_{2,3}$$
$$c_{1,2}\cdot c_{2,3}\cdot c_{1,3|2} \qquad c_{1,2}\cdot c_{1,3}\cdot c_{2,3|1}$$
$$c_{1,3}\cdot c_{2,3}\cdot c_{1,2|3}$$

The consequence of $c_{1,3}\cdot c_{2,3}$ being accepted according to the rule of coherence is that three more vine copulas are accepted: 1, $c_{2,3}$ and $c_{1,3}$ (dashed lines 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 grey in the upper right diagram). The next step of the algorithm is to test using T the most complex vine copulas that are not accepted or rejected: $c_{1,3}\cdot c_{2,3}\cdot c_{1,2|3}$ and $c_{1,2}\cdot c_{2,3}\cdot c_{1,3|2}$ (underlined in the upper right diagram). The former copula is refused, and the latter one is accepted (in the lower left diagram) by the T-test. By the rule of coherence, 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 $c_{1,2}\cdot c_{2,3}\cdot c_{1,3|2}$ (double lines 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, because it allows us to accept or reject many other copulas by using the rule of coherence.

4. Application to the CAC40 index

Here we estimate the joint density of the five main assets composing the *CAC*40, the French leading index, using the methodology described above, and we use this estimated joint density to compute the *VaR* of a portfolio composed of the five assets.

The dataset is taken from Datastream, daily quotes from Total, BNP – Paribas, Sanofi – Synthelabo, GDF-Suez 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 and change of regimes. 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. The portfolio we consider is as follows: Total (33%), BNP – Paribas (20%), Sanofi – Synthelabo (20%), GDF-Suez (14%) and France Telecom (13%).

For each dataset a *GARCH (p,q)* process is selected using the *AIC* criterion (Akaike, 1974) and estimated using pseudo likelihood. On the residual we estimated the vine copula parameters using maximum likelihood. The parametric copula families used in this exercise are the Clayton, Gumbel, Student and Gaussian copulas.

The selection process has been described in Section 3. The test T we used is an Anderson-Darling (A-D) type test modified for use with copulas in high dimension. It is described in Chen et al. (2004) and shown through Monte-Carlo simulation of possessing significant power even in high dimension. We recall that this test associates to each estimated density a χ^2 sample. A complete presentation of the test is detailed in the Appendix.

In dimension five, the lattice structure contains 8536 different vine copulas. Using our lattice-based algorithm in order to select the best vine copula according to the A-D test, we only needed to test 1423 vine copulas (less than 17% of them). In fine, the retained vine copula is[1]:

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

The fact that this vine copula was selected implies that the vine copulas presented in Aas et al. (2009) were rejected by the test, and hence were unable to represent the dependence between the five assets properly. This example shows the necessity of using the whole set of vine copulas for time series applications.

Our final objective is to use this estimated vine copula density to compute the 10% VaR[2]. We computed it from 9/12/08 to 21/11/08 using Monte-Carlo based integration and optimization (see Figure 5). We compared it to a univariate model-based estimate of the VaR computed directly on the portfolio value time series. To discriminate between the two approaches, we use the Kupiec test (Kupiec, 1995). The Kupiec statistic is the number Q of time the out-sample time-series is below the predicted 10% VaR. Under the null of the prediction being a true 10% VaR the sampling distribution of the statistic follows a binomial distribution of parameter 0.1.

In our example the vine copula-VaR has a p-value of 0.96 for the Q statistic, 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 test[3]. 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 (n > 2) density of a portfolio and to compute its associated VaR.
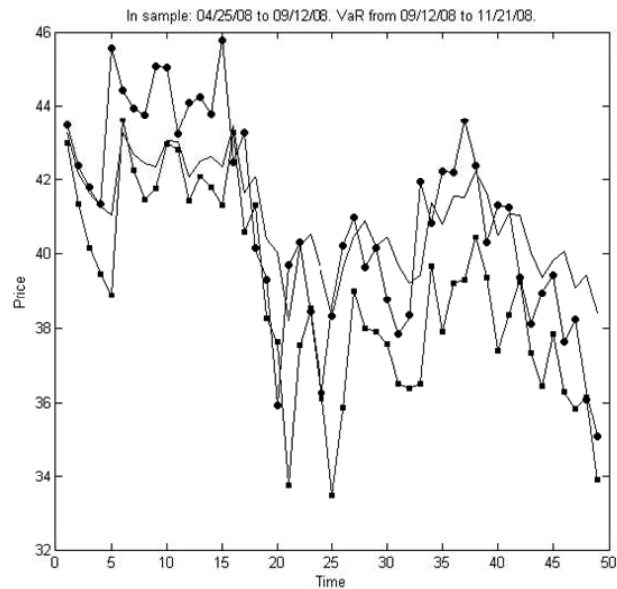

Fig. 2. Dotted line: the CAC40, squared line: Vine VaR estimation, continuous line: GARCH VaR estimation

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 using a large number of vine copulas from which we were able to retain the "best" one according to a predefined test T, using a lattice structure, which makes the research simple and fast. The methodology we developed offers great flexibility for 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 test T 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 analyzed: 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 behavior 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, on which we did not focus 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.

---

[1] The computation took one hour on a *1,5Ghz* processor computer.

[2] Given a random variable *X*, the 10% *VaR* of *X* is the value such that: $P(X < VaR(X)) = 0.1$.

[3] *Q* for the vine copula-*VaR* is equal to 5 and for the *GARCH-VaR* is equal to 21.

Results on the efficiency of the estimates have to be established. We address this question in a companion paper, (Guégan and Maugis, 2009).

References

1. Aas, K., Czado, C., Frigessi, A., Bakken, H. (2009). Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44, 182-198.
2. Akaike, H. (1974). A new look at the statistical model identification. IEEE Trans. Automatic Control AC-19, 716-723.
3. Artzner, P., Delboen, F., Eber, J., Heath, D. (1997). Thinking coherency. Risk 10, 68-71.
4. 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.
5. Bedford, T., Cooke, R. (2002). Vines: A New Graphical Model for Dependent Random Variables. The Annals of Statistics 30 (4), 1031-1068.
6. Berg, D., Aas, K. (2009). Models for construction of multivariate dependence. Forthcoming in The European Journal of Finance.
7. Chen, X., Fan, Y. (2006). Estimation of copula-based semi-parametric time series models. *Journal of Econometrics* 130, 307-335.
8. 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.
9. 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 Paper Munchen, Germany.
10. 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.
11. Gabriel, K.R. (1969). Simultaneous Test Procedures – Some Theory of Multiple Comparisons. Annals of Mathematics Statistics 40 (1), 224-250.
12. Ghysels, E., Harvey, A., Renault, E. (1995). Stochastic volatility. Papers, Toulouse - GREMAQ.
13. Guégan, D., Maugis, P.A. (2009). Dealing With Vines Isues.Work in Progress.
14. Joe, H. (1997). Multivariate Models and Dependence Concepts. Chapman & Hall, London, UK.
15. 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).
16. Napoles, O.M. (2007). Number of vines. 1st Vine Copula Workshop Working Paper.
17. Patton, A. (2009). Handbook of Financial Time Series. Springer Verlag, Ch. Copula-Based Models for Financial Time Series, pp. 767-781.
18. 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.

Appendix

The modified Anderson-Darling test (Chen et al. (2004)

The test is the following: we are testing the null hypothesis H_0 against the alternative hypothesis H_1:

$$H_0 : \Pr\big(C(U_1,...,U_n) = C_0(U_1,...,U_n)\big) = 1,$$

$$H_1 : \Pr\big(C(U_1,...,U_n) = C_0(U_1,...,U_n)\big) < 1,$$

where C is the true copula and C_0 is the estimated copula. We define $\{Z_i\}_{i<n}$ and W as:

$$Z_i = F\big(U_i \mid U_1,...,U_{i-1}\big) \text{ and } W = \sum_1^n \big[\Phi^{-1}(Z_j)\big]^2,$$

where Φ is the standard normal distribution function. Then W follows a χ_d^2 distribution; our test will be based on this result. For this purpose, we use the univariate boundary kernel $K_h(x, y)$:

$$
K_h(x, y) = \begin{cases} k\left(\dfrac{x-y}{h}\right) \Big/ \int_{-\frac{x}{h}}^{1} k(u)\,du, & \text{if } x \in [0, h), \\[2ex] k\left(\dfrac{x-y}{h}\right), & \text{if } x \in [h, 1-h], \\[2ex] k\left(\dfrac{x-y}{h}\right) \Big/ \int_{-\frac{x}{h}}^{1} k(u)\,du, & \text{if } x \in [1-h, 1], \end{cases}
$$

in which $k(\cdot)$ is the quartic kernel: $k(u) = \dfrac{15}{16}\left(1 - u^2\right)^2 1_{|u|<1}$ and h is fixed using the rule of thumb: $h = \sqrt{Var(W)}\, n^{-1/5}$.

Then we define g_W as a kernel estimation of the inverse density of W:

$$
g_W(\omega) = \frac{1}{n.h} \sum_{t}^{T} K_h\!\left(\omega, F_{\chi_d^2}(W_t)\right).
$$

Finally, the test is based on the statistics J_n:

$$
J_n = \int_0^1 \left[g_W(\omega) - 1\right]^2 d\omega.
$$

Then under the null we have that:

$$
Stat_n = \frac{\left(nh^{1/2} J_n - r_n\right)}{\sigma} \to N(0,1) \text{ in distribution},
$$

where:

$$
r_n = h1/2\left[\left(h^{-1} - 2\right)\int_{-1}^{1} k^2(\omega)\,d\omega + 2\int_0^1 \int_{-1}^{z} k_z^2(y)\,dy\,dz\right],
$$

$$
\sigma^2 = 2\int_{-1}^{1}\left[\int_{-1}^{1} k(u+v)k(v)\,dv\right]^2 du,
$$

$$
k_z(y) = k(y) \Big/ \int_{-1}^{z} k(u)\,du.
$$