Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Lehrstuhl für Angewandte Mathematik
und Numerische Mathematik

Preprint BUW-AMNA 09/02

Patrick Deuß

Measuring the Value at Risk of a Stock Portfolio
The Copula Approach

Working Paper

January 2009

http://www-num.math.uni-wuppertal.de/

Contents

List of Figures

1 Introduction

As the title indicates, the aim of this paper is to determine the value at risk of a stock portfolio. We will present two approaches to solve this challenge: the standard correlation approach (COR) and the copula approach (COP). We want to show the short-coming of COR concerning the mapping of multivariate dependency structures. This problem makes COR underestimating the **portfolio risk** whereas COP overcomes this obstacle.

This text is meant to be as a guide to implement the above mentioned approaches. It lacks mathematical finesse and profoundness to keep it as readable as possible. For the pure-maths background we refer to the bibliography.

The paper is organised as follows: in chapter 2 and 3 we describe COR and COP as well as their implementation. Chapter 4 outlines the handling with the operating figure of the value at risk, applies the two approaches to a stock portfolio and depicts numerical results. We finish with a conclusion.

2 The Standard Correlation Approach (COR)

In our model setup we assume that our stock portfolio comprises I assets and each stock price S_t^i, $i = 1, \dots, I$, evolves according to the following stochastic differential equation

$$dS_t^i = \mu_i(S_t, t)dt + \sigma_i(S_t, t)dW_t^i \qquad \text{for } t \in [0, T] \text{ and } T < \infty$$

in which μ_i and σ_i are technical adequate functions (e.g. integrable) and W_t^i is a Brownian motion for $i = 1, \dots, I$.

Note that S_t^i is driven by just **one** Brownian motion. We come to that later when we deal with the whole portfolio.

Although we know that stock price returns deviate from the (log-)normal distribution (such returns often have heavy-tailed distributions as many empirical analysis show), we believe that the incorporated **risk** of a stock portfolio can be grasped via its dependence structure and its marginal distributions (i.e. the copula) and not solely through the stock price model.

That's why we choose a tractable model and regard the functions $\mu_i(S_t, t)$ and $\sigma_i(S_t, t)$ as constant. This leads to the following, log-normal distributed stock price model and its solution.

$$\left(\begin{array}{l} dS_t^i = \mu_i S_t^i dt + \sigma_i S_t^i dW_t^i \quad \text{with initial condition } S_0^i \text{ for } i = 1, \dots, I \\ S_t^i = S_0^i \exp\left(\underbrace{(\mu_i - \frac{1}{2}\sigma_i^2)}_{:=\alpha_i} t + \sigma_i W_t^i \right) \quad \Leftrightarrow \quad \ln\left(\frac{S_t^i}{S_0^i}\right) = \alpha_i t + \sigma_i W_t^i \end{array} \right) \tag{2.1}$$

We denote α_i as the *mean or expected rate of return* and σ_i as the *standard deviation* of asset i, $i = 1, \ldots, I$.

2.1 Data Fitting - Stock Prices

Suppose we have $N+1$ observed closing prices S_t^i, $i = 1, \ldots, I$ and $t = 1, \ldots, N+1$. As we deal with a log-normal model (2.1), we have to take the logarithm of these prices:

$$\gamma_i(t) := \ln\left(\frac{S_t^i}{S_{t-1}^i}\right)$$

See that we have now N observed daily log-returns $\gamma_i(t)$ and we remember that

$$E\left[\ln\left(\frac{S_t^i}{S_0^i}\right)\right] = \alpha_i t \quad \text{or} \quad E\left[\ln\left(\frac{S_t^i}{S_{t-1}^i}\right)\right] = \alpha_i$$

due to the properties of the Brownian motion. In order to estimate the expected rate of return γ_i we have compute the expression

$$\gamma_i := \frac{1}{N}\sum_{n=1}^{N}\gamma_i(n)$$

We set $\gamma = (\gamma_1, \ldots, \gamma_I)^t$. Moreover, we calculate the variance η_i^2 of asset i.

$$\eta_i^2 := \frac{1}{N-1}\sum_{n=1}^{N}(\gamma_i(n) - \gamma_i)^2 \quad \text{for } i = 1, \ldots, I$$

Thus, setting $\alpha_i = \gamma_i$ and $\sigma_i = +\sqrt{\eta_i^2}$ we have fitted model (2.1):

$$S_t^i = S_0^i \exp\left(\alpha_i t + \sigma_i W_t^i\right) \tag{2.2}$$

2.2 The Multidimensional Model

However, we want to set up a multidimensional model. Let's denote

i) $S_t := (S_t^1, \ldots, S_t^I)^t$

ii) $\alpha := (\alpha_1, \ldots, \alpha_I)^t$

iii) $W_t := (W_t^1, \ldots, W_t^I)^t$

iv) $D := \mathrm{diag}(\sigma_1, \ldots, \sigma_I)$.

Note that each stock price is driven by **one** Brownian motion which leads to the vector $W_t = (W_t^1, \ldots, W_t^I)^t$. It is also possible to model **one** stock price by means of **several** Brownian motions. We omit this case here.

With this notation we are able to write (2.1) or (2.2), respectively, in the following multidimensional form

$$\begin{aligned} S_t &= S_0 \exp\left(\alpha t + DW_t\right) \\ &= \begin{pmatrix} S_0^1 \\ \vdots \\ S_0^I \end{pmatrix} \cdot \exp\left[\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_I \end{pmatrix} \cdot t + \begin{pmatrix} \sigma_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_I \end{pmatrix} \cdot \begin{pmatrix} W_t^1 \\ \vdots \\ W_t^I \end{pmatrix} \right] \end{aligned} \tag{2.3}$$

In order to simplify the implementation of the model (2.3) the Brownian motions W_t^i, $i = 1, \ldots, I$, should be drawn independently. The problem is that the observation of the data set might reveal a certain correlation between the price behaviour of our assets. Hence, the Brownian motions have to be connected to each other according to a possibly arisen linear dependence structure. It is consequently not sufficient to model the randomness of the stock price behaviour via the expression $D \cdot W_t$.

We determine the covariance matrix Σ of the observation matrix

$$\Gamma := (\gamma_{i,n}) = \gamma_i(n) \quad \text{with } i = 1, \ldots, I \text{ and } n = 1, \ldots, N.$$

The i,j-th entry of $\Sigma \in \mathbb{R}^{I \times I}$ is the covariance between asset i and asset j, the diagonal elements are the variances of each asset. Our aim is now to generate random variables which are multivariate-normal distributed with the very covariance matrix Σ. Let's have a closer look at Σ.

$$\begin{aligned} \Sigma &= \begin{pmatrix} \rho_{1,1} \cdot \sigma_1 \cdot \sigma_1 & \ldots & \rho_{1,I} \cdot \sigma_1 \cdot \sigma_I \\ \vdots & \ddots & \vdots \\ \rho_{I,1} \cdot \sigma_I \cdot \sigma_1 & \cdots & \rho_{I,I} \cdot \sigma_I \cdot \sigma_I \end{pmatrix} \\ &= \begin{pmatrix} \sigma_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_I \end{pmatrix} \begin{pmatrix} 1 & & \rho(i,j) \\ & \ddots & \\ \rho(j,i) & & 1 \end{pmatrix} \begin{pmatrix} \sigma_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_I \end{pmatrix} \\ &= D \cdot P \cdot D \end{aligned}$$

The matrix P denotes the correlation matrix and $\rho(i,j)$ indicates the correlation between security i and j. P and Σ are symmetric by nature and they are positive definite as a rule.

It is known, that the expression $\Sigma \cdot W_t = DPD \cdot W_t$ is not multivariate-normal distributed, i.e. $\Sigma \cdot W_t \nsim N(\mathbf{0}, \mathbf{\Sigma t})$. To eliminate this problem we initially consider the I-dimensional Brownian motion W_t.

2.2.1 A Guide to Implementation

As implementation always means to discretise things, we regard the time horizon $[0,T]$, $T < \infty$, divide it into M sub-intervals of the length Δt, i.e. $\Delta t = \frac{T}{M}$ and denote the discretized time steps $t_m := m\Delta t$, $m = 1, \ldots, M$.

Furthermore, we know that the increments of a Brownian motion $Z_{\Delta t} = W_{t_m} - W_{t_{m-1}}$, $m = 1, \ldots, M$, are $N(\mathbf{0}, \mathbf{\Delta t})$-distributed and we thus consider the I-dimensional random variable $Z = (Z_1, \ldots, Z_I)^t$ for each time step t_m, where we

skip the index for the sake of readability. Note that the Z_i, $i = 1, \ldots, I$, are drawn independently.

If we now scale a standard normal-distributed random variable Y with the standard deviation $\sqrt{\Delta t}$ of Z, we receive the very variate $Z = \sqrt{\Delta t} \cdot Y$, $Y \sim N(\mathbf{0}, \mathbf{1})$.

However, the expression $\sqrt{\Delta t} \cdot \Sigma \cdot Y$ is still not multivariate-normal distributed. We have to revise the matrix Σ.

As it can be found in many textbooks on numerics and / or generation of multivariate random variables, we compute the Cholesky decomposition L of Σ (possible as $x^t \Sigma x > 0$ for $x \neq 0$):

$$\Sigma = LL^t$$

in which L is a lower triangular matrix.

Please note that we are also able to use the correlation matrix P for the Cholesky decomposition. We just have to add the diagonal matrices D:

$$P = \tilde{L}\tilde{L}^t \quad \Rightarrow \quad \Sigma = DPD = D\tilde{L}\tilde{L}^t D = D\tilde{L}\tilde{L}^t D^t = D\tilde{L}(D\tilde{L})^t = LL^t$$

Together with the random variable $Y \sim N(\mathbf{0}, \mathbf{1})$, the matrix L creates

$$L \cdot Y \sim N(\mathbf{0}, \mathbf{\Sigma}).$$

Putting these considerations together, we receive

$$V := \alpha \Delta t + \sqrt{\Delta t} \cdot L \cdot Y \sim N(\alpha \mathbf{\Delta t}, \mathbf{\Sigma \Delta t}).$$

Thus, V is the random variable we searched for and which constitutes linear dependence among our traded securities.

$$S_{t_m} = S_{t_{m-1}} \exp(V) = S_{t_{m-1}} \exp\left(\alpha \Delta t + \sqrt{\Delta t} \cdot L \cdot Y\right) \tag{2.4}$$

Expression (2.4) can now easily be implemented for each time step t_m, $m = 1, \ldots, M$. Notice that in each step one has to draw I independent standard normal variables Y_i, $i = 1, \ldots, I$. Of course, if we set $\Delta t = T$, we receive directly the simulated final quotation S_T.

Some MATLAB functions will simplify the implementation. The calculation of γ for instance is solved by the command `mean(Γ, 2)`. Moreover, `mvnrnd(α, Σ, M)` creates M $N(\alpha, \mathbf{\Sigma})$-distributed random variables. Hence, we don't have to perform a Cholesky decomposition of Σ, the covariance matrix of Γ.

2.2.2 The Continuous-Time Setup

This section is included for the sake of completeness and just specifies the multidimensional stock price evolutions incorporating the linear dependence structure.

As $W_t \sim N(\mathbf{0}, \mathbf{t})$ and hence $\tilde{V} := \alpha t + L \cdot W_t \sim N(\alpha \mathbf{t}, \mathbf{\Sigma t})$, the continuous-time setup (2.3) changes to

$$S_t = S_0 \exp(\alpha t + L \cdot W_t) \tag{2.5}$$

2.3 Basing Upon and Forecasting a Period

We are endowed with the mean rate of return α and the covariance matrix Σ. Hence, we have fitted model (2.4). See that we estimated α and Σ on daytime basis. Implementing this setup for $m = 1$ means that we forecast the performance of our portfolio for the **next** day. Thus, calculating $m = 1, \ldots, M$ steps the price behaviour for M days is simulated.

Suppose now that we want to do the forecasting for several days $k = 1, \ldots, K$, for $K < N$. E.g., $K = 5$ would be a trading week, $K = 10$ two weeks and $K = 20$ symbolises a month. We are interested in howfar the portfolio evolves during this period. On the one hand this can be done in the above mentioned way in setting $m = k$. On the other hand we can adapt the fitting which has the advantage to reduce complexity.

Assume that $K \ll N$, i.e. the regarded period K is much smaller than the observed data N. For simplicity reasons we claim that $N \mod K \equiv 0$. Otherwise, some daily data has to be skipped. We are interested in fitting the model to a period of K trading days.

Consequently, we change estimations in the following way: we sum up the log-returns for each K trading days

$$\gamma_{i,k} = \sum_{n=1}^{K} \gamma_i(k \cdot K + n) \quad \text{for } k = 0, 1, 2, \ldots, \frac{N-1}{K}$$

and compute the mean with the abbreviation $\xi(K) := \frac{N-1}{K}$

$$\gamma_i^K = \frac{1}{\xi(K)} \sum_{k=0}^{\xi(K)} \gamma_{i,k}$$

and denote

$$\gamma^K := (\gamma_1^K, \ldots, \gamma_I^K)^t \quad \text{and} \quad \alpha^K := \gamma^K.$$

The covariance matrix Σ^K is calculated from the observation matrix $\Gamma^K = \gamma_{i,k}$, $i = 1, \ldots, I$ and $k = 0, 1, \ldots, \xi(K)$.

Digression

As we have

$$\begin{aligned}\gamma_{i,k} &= \sum_{n=1}^{K} \gamma_i(k \cdot K + n) = \gamma_i(k \cdot K) + \ldots + \gamma_i((k+1) \cdot K) \\ &= \ln\left(\frac{S_{kK}^i}{S_{kK-1}^i}\right) + \ln\left(\frac{S_{kK+1}^i}{S_{kK}^i}\right) + \ldots + \ln\left(\frac{S_{(k+1)K}^i}{S_{(k+1)K-1}^i}\right) = \ln\left(\frac{S_{(k+1)K}^i}{S_{kK-1}^i}\right)\end{aligned}$$

we can also deal with the raw data and just calculate the log-returns of every period of K trading days. We leave this decision to the reader. In our point of view it is easier to implement the above mentioned procedure.

Of course, it might be strange to some readers why we fit the setup to K trading days instead of the normally used one-year period. We will see later that we

are interested in forecasting a certain time period of $K = 1, 5, 10$ or 20 trading days. Fitted to this very period, simulations will allow us to use a normalised time step of $\Delta t = 1$. This implementation will provide the inherited risk (value at risk) of the portfolio.

2.4 Monte Carlo Simulation - Part I

Suppose we regard a period of K trading days. For the sake of simplicity we set the initial values $S_0^i := 1$, $i = 1, \ldots, I$, and assume that all assets in our I stock portfolio are equally weighted. Calibrating setup (2.4) by means of historical data provides us with the expressions α^K and Σ^K. As these terms are fitted on basis of a K-trading-days-period and we want to simulate the **next** time step (i.e. the next K trading days), it yields $\Delta t = 1$.

In the continuous-time model (2.5) this fact leads to

$$S_1 = S_0 \exp(\alpha^K + L^K \cdot W_1)$$

which is in this case equal to the implementation presetting (2.3)

$$S_{t_1} = S_{t_0} \exp(\alpha^K + L^K \cdot Y).$$

In other words, for forecasting the next period we have to produce $N(\alpha^{\mathbf{K}}, \mathbf{\Sigma^K})$-distributed random variables and put it into the exponential function.

Now, we would like to perform R Monte Carlo simulations, i.e. we estimate the outcome (this is the expected return) for the next K trading days by simulating the return of our portfolio R times and calculating the mean.

First, as mentioned in subsection 2.2.1 the expression

$$y = \texttt{mvnrnd(}\alpha^K\texttt{,}\ \Sigma^K\texttt{,}\ R\texttt{)}$$

produces a $R \times I$ matrix y which entries are $N(\alpha^{\mathbf{K}}, \mathbf{\Sigma^K})$-distributed.

Thus, the expected stock price $\hat{S}_r^i$, $i = 1, \ldots, I$ and $r = 1, \ldots, R$, of the i-th asset and the r-th simulation run is therefore

$$\hat{S}_r^i = \underbrace{S_0^i}_{=1} \exp(y_{r,i}).$$

Summing up the columns of $\hat{S}_r^i$ gives us the expected return for the next period of K trading days of the r-th simulation run (remember that all stocks are equally weighted).

$$\hat{S}_r = \frac{1}{I} \sum_{i=1}^{I} \hat{S}_r^i \quad \text{for } r = 1, \ldots, R$$

We summarise the R different Monte Carlo outcomes:

$$\hat{S} = \frac{1}{R} \sum_{r=1}^{R} \hat{S}_r \tag{2.6}$$

$\hat{S}$ is hence the simulated or expected portfolio return for the next K trading days.

Remark - $K \ll N$

One notes that K should be much smaller than N to hold a certain consistency as the data is divided into $\frac{N}{K}$ sets.

3 The Copula Approach (COP)

In the previous chapter we examined expected stock price behaviour in the Merton setup (2.1). When we modelled the multidimensional case (equation (2.5)) we took into account that stock prices are correlated, i.e. that they are linear dependent. But considering multidimensional returns there might occur different dependencies which are possibly not grasped via this model.

3.1 Multivariate Distributions

For a multivariate distribution function F the important theorem of Sklar (1959) shows that this distribution F can be separated into its marginal distribution functions F_i, $i = 1, \ldots, I$, often called *margins* and their dependency structure, the copula function C.

3.1 Theorem (Sklar)
Let F be a continuous multivariate distribution and $F_1, \ldots, F_I$ continuous margins. Then, there exists a unique, I-dimensional copula function C which holds the following equivalent equations

$$F(z_1, \ldots, z_I) = C(F_1(z_1), \ldots, F_I(z_I)) \tag{3.1}$$

$$F(F_1^{-1}(v_1), \ldots, F_I^{-1}(v_I)) = C(v_1, \ldots, v_I) \tag{3.2}$$

in which $z_i \in \mathbb{R}$, $v_i \in [0, 1]$, $v_i = F_i^{-1}(z_i)$, $i = 1, \ldots, I$, and F_i^{-1} is the inverse function of the marginal distribution F_i.

For further details see the excellent text book of Nelsen [7].

The Problem

Every investor holding a stock portfolio is exposed to the market risk of asset prices and wants to estimate it. The question might arise whether extreme outcomes of a single asset will have significant influence on the whole portfolio. If yes, can this dependence be modelled via the linear correlation? Or, in other words, is the stock price setup able to capture the actual risk structure? We do not think so.

As we see in equation (3.1) we need the marginal distribution functions F_i, $i = 1, \ldots, I$. In our case it is nothing else than the distribution functions of the

log-returns of every single asset in the portfolio. There are several different ways of estimating and / or fitting these marginals. An example for a parametric approach would be the maximum likelihood estimator. We restrict our analysis to the empirical distribution function. This is often called *non-parametric estimation* and only possible if the data set is very substantial which will be fact for our data.

3.2 About Empirical Distribution Functions and Ranks

We denote $x_{i,n}$ as the n-th log-return of the i-th asset. Consistent with the former notation it is $i = 1, \dots, I$ and $n = 1, \dots, N$ and we get the following *observation matrix*:

$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{I,1} & \dots & x_{I,N} \end{pmatrix}$$

3.2 Definition (Empirical Distribution Function)
The *empirical distribution function* of asset i is defined as

$$F_{i,N}(x) := \frac{1}{N+1} \sum_{n=1}^{N} \mathbf{1}_{\{x_{i,n} \leq x\}}(x) \tag{3.3}$$

The factor $\frac{1}{N+1}$ effects that $F_{i,N} \in [0, 1)$. Evaluations of distribution functions at the boundary 1 might cause problems. That's why we keep the distribution function artificially smaller than 1.

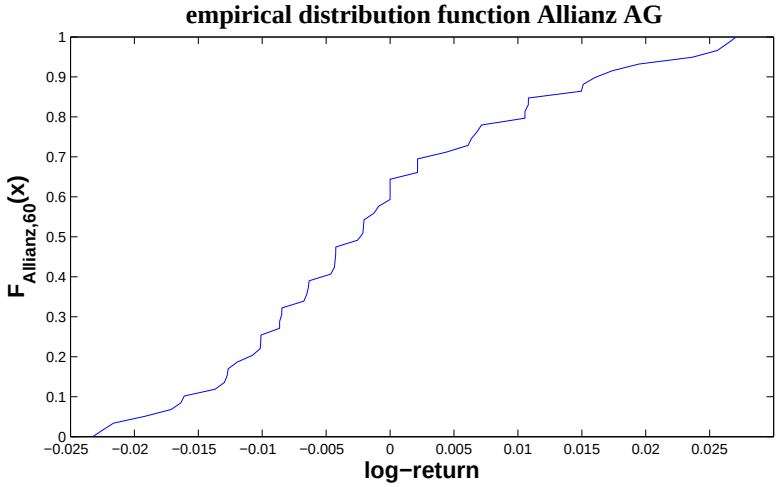


Figure 3.1: Allianz AG, $N = 60$ log-returns from 18/10/07 to 16/01/08

Figure 3.1 shows the empirical distribution function for Allianz AG obtained by an analysis of $N = 60$ log-returns from 18 October 2007 to 16 January 2008.

Instead of implementing equation (3.3) it is equivalent to compute the ranks of our observation data and divide them by $N+1$. We just have to sort the rows of matrix X, i.e. we find

$$\frac{rank(x_{i\cdot})}{N+1} \triangleq F_{i,N}(x_{i\cdot})$$

which simplifies the implementation and saves computing time.

Implementation in MATLAB - computing ranks

i) temp1 = $[x(i,:)' \quad (1:N)']$;

ii) temp2 = sortrows(temp1, 1),

iii) temp3 = [temp2 $(1:N)'./(N+1)$];

iv) temp4 = sortrows(temp3, 2);

v) rank(i,:) = temp4(:, 3);

In the first step we set a marker for remembering the original position of each $x_{i,n}$, $i = 1, \ldots, I$ and $n = 1, \ldots, N$. Step ii) sorts the $x_{i,n}$ according to their values. Step iii) adds the probabilities $(1:N)'./(N+1)$ ($\rightarrow$ empirical distribution function). Step iv) sets everything back to its original order with help of the indicator step i). In the last step we read out the ranks for every $x_{i,n}$.

With the aid of this algorithm we compute the following matrix:

$$\begin{aligned} U &= \begin{pmatrix} F_{1,N}(x_{1,1}) & \ldots & F_{1,N}(x_{1,N}) \\ \vdots & \ddots & \vdots \\ F_{I,N}(x_{I,1}) & \ldots & F_{I,N}(x_{I,N}) \end{pmatrix} \\ &= \begin{pmatrix} rank(x_{1,1}) & \ldots & rank(x_{1,N}) \\ \vdots & \ddots & \vdots \\ rank(x_{I,1}) & \ldots & rank(x_{I,N}) \end{pmatrix} = \begin{pmatrix} u_{1,1} & \ldots & u_{1,N} \\ \vdots & \ddots & \vdots \\ u_{I,1} & \ldots & u_{I,N} \end{pmatrix} \end{aligned}$$

It holds $u_{i,n} \in [0,1)$. The matrix and its entries are often called *pseudo-observations*.

3.3 Structure and Simulation of Standard Copulas

By Sklar's Theorem 3.1 we know the following representation for a multivariate distribution function F

$$C(F_1(z_1), \ldots, F_I(z_I)) = F(z_1, \ldots, z_I)$$

We have already determined the marginal distribution functions F_i, $i = 1, \ldots, I$, for each asset - these are the empirical distribution functions $F_{i,N}$ which we will make use of later. So, we have to try out in howfar they are connected to each other. We need to specify the multivariate distribution function F. Suppose

the margins are continuous. Then, by inverting the margins (F_i^{-1}, $i = 1, \ldots, I$), the dependence structure (i.e. the copula) is revealed.

$$\begin{aligned} & F_i(z_i) := v_i \quad \Rightarrow \quad F_i^{-1}(v_i) = z_i \\ \Rightarrow \quad & C(v_1, \ldots, v_I) = F(F_1^{-1}(v_1), \ldots, F_I^{-1}(v_I)) \end{aligned}$$

The vector $v = (v_1, \ldots, v_I)^t$ with such a representation is distributed according to copula C.

However, we would like to **simulate** this very vector $v = (v_1, \ldots, v_I)^t$ which is distributed according to a copula C. To do this we have to reverse the order of our approach. First, we simulate random variables which are distributed according to the multivariate distribution function F and get a vector $z = (z_1, \ldots, z_I)^t$. Now, we apply the margins F_i to each component of z and receive

$$(F_1(z_1), \ldots, F_I(z_I))^t = (v_1, \ldots, v_I)^t = v.$$

By the representation of Sklar we know that v is distributed according to the copula C. This copula C has to be defined by the risk manager. She or he determines the dependency structure, the company wants to implement. In the next two subsections we present those copulas which are frequently used in financial risk application and we want to deal with.

3.3.1 The Gaussian Copula

For setting up the Gaussian copula we have to choose $F_i^{-1} = \phi^{-1}$ the standard normal distribution function. A **standard** distribution function is a distribution function with **mean zero** and **variance one**. Moreover, applying $F = \Phi^I(0, P) := \Phi_P^I$ the I-dimensional standard normal distribution function with mean $\mathbf{0}$ and correlation matrix P we receive the Gaussian copula.

$$\begin{aligned} C_P^{Ga}(v_1, \ldots, v_I) &= \Phi_P^I(\phi^{-1}(v_1), \ldots, \phi^{-1}(v_I)) \\ &= \int_{-\infty}^{\Phi^{-1}(v_1)} \cdots \int_{-\infty}^{\Phi^{-1}(v_I)} \frac{1}{(2\pi)^{I/2}|P|^{1/2}} \exp\left(-\frac{1}{2} z^t P^{-1} z\right) dz_1 \ldots dz_I \end{aligned} \tag{3.4}$$

in which $|P|$ is the determinant of P.

Please note that in case of $P = \mathbf{1}_I$ - the identity matrix - we get the *independence* or *product* copula

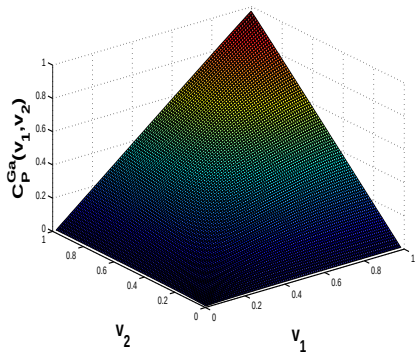
$$C(v_1, \ldots, v_I) = \prod_{i=1}^{I} v_i$$

which represents a complete independence of each component.

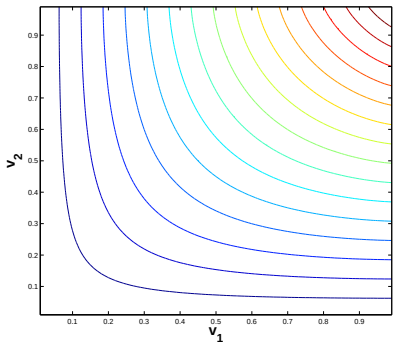
Due to some computation one obtains the density of the Gaussian copula

$$c_P^{Ga}(v_1, \ldots, v_I) = |P|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} z^t \left(P^{-1} - \mathbf{1}_I\right) z\right) dx_1 \ldots dx_I$$

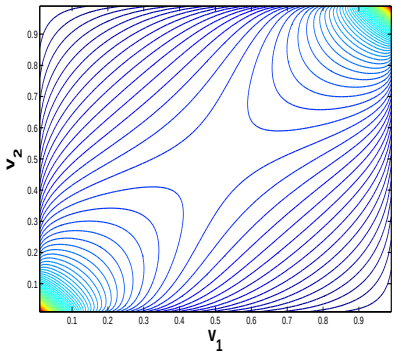
in which $z = (z_1, \ldots, z_I)^t = (\phi^{-1}(v_1), \ldots, \phi^{-1}(v_I)^t)^t$.



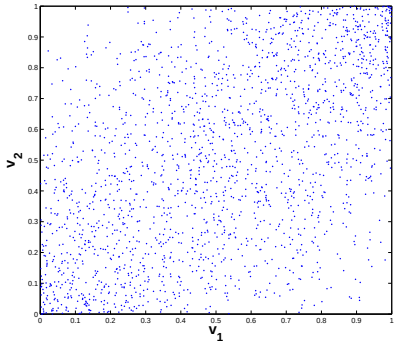
(a) perspective of the distribution function



(b) contour plot of the distribution function



(c) contour plot of the density function



(d) 2000 simulated points for C_P^{Ga}

Figure 3.2: Gaussian copula with correlation $\rho = 0,5$

To get an idea about this equation and the behaviour of the Gaussian copula we specify distribution and density function as well as 2000 simulated points of the 2-dimensional Gaussian copula with correlation matrix $P = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$. These properties are shown in figure 3.2.

For the simulation of a vector v which is distributed according to the Gaussian copula as in subfigure 3.2d we have the following algorithm.

3.3 Algorithm (Simulation of the Gaussian copula)

i) Generate $Z \sim \Phi^I(0, P)$ and receive $z = (z_1, \ldots, z_I)^t$.

ii) Apply ϕ (standard normal distribution function) to each component of z and receive $v = (v_1, \ldots, v_I)^t = (\phi(z_1), \ldots, \phi(z_I))^t$.

iii) v is distributed according to C_P^{Ga}.

Further explanations and comments on algorithm 3.3 can be found in McNeil et al. [1] and references therein.

3.3.2 The t-Copula

As well as the Gaussian copula the Student t-copula or just t-copula belongs to the family of elliptical copula functions which are often used in financial application due to their analytical tractability.

Compared to the Gaussian one, the t-copula has one additional parameter - ν the degrees of freedom. This problem will challenge us later, see section 3.4 Fitting Copulas. The margins of this copula are the univariate standard t-distributions, i.e. $F_i^{-1} := t_\nu^{-1}$ with ν indicating the degrees of freedom. Moreover, the multi-dimensional structure is given by the multivariate standard t-distribution with mean $\mathbf{0}$ and correlation matrix P, $F := t_\nu^I(0, P) := t_{\nu,P}^I$.

$$C_{\nu,P}^t(v_1, \ldots, v_I) = t_{\nu,P}^I\left(t_\nu^{-1}(v_1), \ldots, t_\nu^{-1}(v_I)\right) \tag{3.5}$$
$$= \int_{-\infty}^{t_\nu^{-1}(v_1)} \cdots \int_{-\infty}^{t_\nu^{-1}(v_I)} \frac{\Gamma\left(\frac{\nu+I}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{(\nu\pi)^I|P|}} \left(1 + \frac{z^t P^{-1} z}{\nu}\right)^{\frac{\nu+I}{2}} dz_1 \ldots dz_I$$

in which Γ denotes the usual Gamma function.

One small note: the t-copula converges to the Gaussian for $\nu \to \infty$. The density for (3.5) is given by

$$c_P^t(v_1, \ldots, v_I) = |P|^{-\frac{1}{2}} \left(\frac{\Gamma\left(\frac{\nu+I}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\right) \left[\frac{\Gamma\left(\frac{\nu}{2}\right)}{\Gamma\left(\frac{\nu+1}{2}\right)}\right]^I \frac{\left(1 + \frac{z^t P^{-1} z}{\nu}\right)^{-\frac{\nu+I}{2}}}{\prod_{i=1}^I \left(1 + \frac{z_i^2}{\nu}\right)^{-\frac{\nu+1}{2}}}$$

in which $z = (z_1, \ldots, z_I)^t = (t_\nu^{-1}(v_1), \ldots, t_\nu^{-1}(v_I)^t)^t$.

As we did for the Gaussian copula we visualise some basic properties of the bivariate t-copula in figure 3.3. Again, we have chosen $\rho = 0,5$, i.e. $P = \begin{pmatrix} 1 & 0,5 \\ 0,5 & 1 \end{pmatrix}$ and set $\nu = 3$.

Similar to the simulation of $v \sim C_P^{Ga}$, we present an algorithm simulating a vector which are distributed according to the t-copula.

3.4 Algorithm (Simulation of the t-copula)

i) Generate $Z \sim t_{\nu,P}^I$ and receive $z = (z_1, \ldots, z_I)^t$.

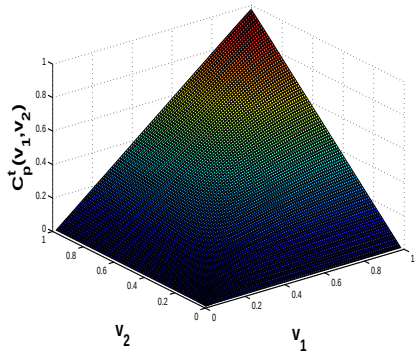
ii) Apply t_ν (standard univariate t-distribution function) to each component of z and receive $v = (v_1, \ldots, v_I)^t = (t_\nu(z_1), \ldots, t_\nu(z_I))^t$.

iii) v is distributed according to $C_{\nu,P}^t$.

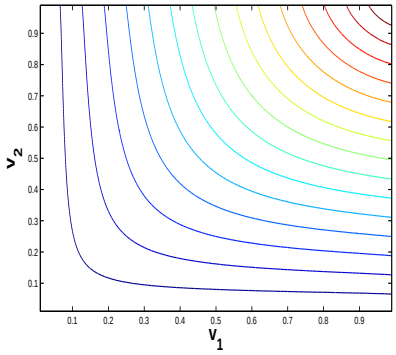
Here too, compare McNeil et al. [1] and references therein. From the simulation of copulas we turn to their calibration.

3.4 Fitting Copulas

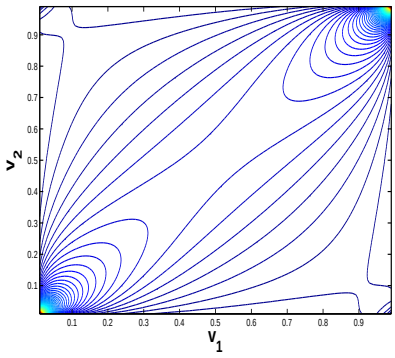
In this section we want to present an outline for the problem of how to calibrate copula functions. We will apply the so-called *pseudo-loglikelihood estimation*,



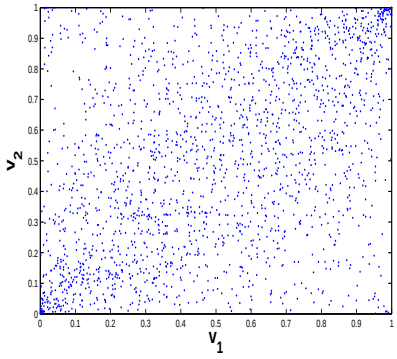
(a) perspective of the distribution function



(b) contour plot of the distribution function



(c) contour plot of the density function



(d) 2000 simulated points for C_P^t

Figure 3.3: t-copula with correlation $\rho = 0,5$ and $\nu = 3$ degrees of freedom

an approach which in our opinion fits best for our model setup.

As seen, copulas contain certain parameters. Therefore, we rewrite equation (3.1) to indicate the parametric dependence of the copula.

$$F(z_1, \dots, z_I, \theta_1, \dots, \theta_I, \theta) = C(F_1(z_1, \theta_1), \dots, F_I(z_I, \theta_i), \theta) \tag{3.6}$$

We see that on the one hand every margin F_i includes the parameters $\theta_i = (\theta_i^1, \theta_i^2, \dots)$ depending on the structure of F_i itself. On the other hand the copula embodies the parameters $\theta = (\theta^1, \theta^2, \dots)$ which is as well up to the choice of the copula C.

3.5 Example (Parameters of the Gaussian Copula)

$$C_P^{Ga}(v_1, \dots, v_I) = \Phi_P^I(\phi^{-1}_{\mu_1,\sigma_1}(v_1), \dots, \phi^{-1}_{\mu_I,\sigma_I}(v_I))$$

i) As $F_i = \Phi(\mu_i, \sigma_i)$ is chosen as univariate normal distribution function, the margins of the Gaussian copula contain the means μ_i and the standard deviation σ_i, $i = 1, \dots, I$. If, of course, F_i is given as the standard univariate distribution function, the parameters are zero and one (mean and variance) a priori.

ii) Moreover, as $F = \Phi_P^I$ holds for the Gaussian copula, we see that the copula parameter θ is the to be estimated correlation matrix P. Remember that the coherence of the covariance matrix Σ and the correlation matrix is shown in subsection 2.2.1. Additionally, if one wants the Gaussian copula not be centred, the mean vector μ increases the parameter family to $\theta = (\mu, \Sigma)$. We restrict to the standard case, i.e. we only need to estimate the correlation matrix P.

iii) All in all, we have to fit and optimise on the basis of $\frac{I(I-1)}{2}$ variables in the standard case.

The calibration of these parameters is done by maximisation of the log-likelihood function for the copula model (3.6)

$$l(\theta_1, \dots, \theta_I, \theta, x) = \sum_{n=1}^{N} \ln \underbrace{c(F_1(x_{1,n}, \theta_1), \dots, F_I(x_{I,n}, \theta_I), \theta) \cdot \prod_{i=1}^{I} f_i(x_{i,n}, \theta_i)}_{=:\eta}$$

in which c

$$c(v_1, \dots, v_I, \theta) = \frac{\partial^d C(v_1, \dots, v_I, \theta)}{\partial v_1 \cdots \partial v_I}$$

is the density of copula C and f_i, $i = 1, \dots, I$, are the marginal densities of the distribution functions F_i. See that the term η is the density of the copula model (3.6). For more information we refer to Dias [6] and Tappe [2].

However, we have already determined the empirical distribution functions $F_{i,N}$ and the pseudo-observation matrix U. This means we are able to skip the margin parameters $(\theta_1, \dots, \theta_I)$ as they are implied in the empirical distribution

functions - the $F_{i,N}$, $i = 1, \ldots, I$, do not depend on parameters. Thus, we have to maximise the so-called *pseudo log-likelihood function*

$$\hat{l}(\theta, x) = \sum_{n=1}^{N} \ln c(F_{1,N}(x_{1,n}), \ldots, F_{I,N}(x_{I,n}), \theta)$$

Summarised, we are searching for

$$\theta^* = \arg\max_{\theta \in \Theta} \sum_{n=1}^{N} \ln c(u_{1,n}, \ldots, u_{I,n}, \theta)$$

in which Θ indicates the admissible set of model parameters. For a detailed prescription concerning log-likehood estimators and calibrating issues compare e.g. Dias [6].

3.4.1 Calibrating the Gaussian Copula

In case of the Gaussian copula the remaining parameter to fit is the correlation matrix P. McNeil et al. [1] (compare example 5.53) showed how to find the maximum likelihood estimator (MLE) for that problem. It turns out to be sufficient to transform the pseudo observation matrix U by means of the inverse function of the standard univariate distribution Φ^{-1}, i.e.

$$\Phi^{-1}(U) = \begin{pmatrix} \Phi^{-1}(u_{1,1}) & \ldots & \Phi^{-1}(u_{1,N}) \\ \vdots & \ddots & \vdots \\ \Phi^{-1}(u_{I,1}) & \ldots & \Phi^{-1}(u_{I,N}) \end{pmatrix}$$

Then take the correlation of this matrix

$$\hat{P} = \rho(\Phi^{-1}(U))$$

which is the searched MLE for the Gaussian copula. Now, we are able to apply algorithm 3.3 to simulate random variables distributed according to $C^{Ga}_{\hat{P}}$.

3.6 Algorithm (Calibration of the Gaussian Copula)

i) Determine the empirical distribution functions $F_{i,N}$, $i = 1, \ldots, I$.

ii) Apply $F_{i,N}$ to the log-return data matrix X and receive the observation matrix U.

iii) Transform matrix U by means of the inverse of the standard normal distribution $\Phi^{-1}(U)$.

iv) Compute the correlation matrix $\hat{P} \in [-1, 1]^{I \times I}$ of matrix $\Phi^{-1}(U)$.

As we have calibrated the copula, we have set up a model. Hence, we are ready to apply algorithm 3.3 to simulate random variables distributed according the Gaussian copula.

3.4.2 Calibrating the t-copula

Regarding the t-copula we see that besides the correlation matrix P we have to fit one additional parameter, ν the degrees of freedom. McNeil et al. [1] propose to calibrate the correlation matrix $\hat{P}$ by means of Kendall's tau and the degrees of freedom by means of the pseudo log-likelihood estimation, see examples 5.54 and 5.59 and compare Tappe [2].

First, we determine the rank correlation (i.e. Kendall's tau) of matrix U and transform this matrix $\tau(U) \in \mathbb{R}^{I \times I}$:

$$\hat{P} = \sin\left(\frac{\pi\tau(U)}{2}\right)$$

to get the estimated correlation matrix $\hat{P}$. Now, the pseudo log-likelihood function $\hat{l}$ only depends on the degrees of freedom ν. A maximisation can be effected numerically.

3.7 Algorithm (Calibration of the t-copula)

i) Determine the empirical distribution functions $F_{i,N}$, $i = 1, \ldots, I$.

ii) Apply $F_{i,N}$ to the log-return data matrix X and receive the observation matrix U.

iii) Calculate $\tau(U)$, Kendall's rank correlation coefficients.

iv) Transform $\hat{P} = \sin\left(\frac{\pi\tau(U)}{2}\right)$ to receive the estimated correlation matrix.

v) Maximise the log-likelihood function $\hat{l}(\nu, \hat{P}, U)$ with respect to ν numerically.

Again, we are able to simulate the fitted t-copula according to algorithm 3.4. For a workaround of this section we refer to McNeil et al. [1], Dias [6] and Tappe [2].

3.5 Monte Carlo Simulation - Part II

By now, we know how to calibrate as well as to simulate the Gaussian and the Student t-copula. As in section 2.4 we would like to analyse a period of K trading days, i.e. we want to know - or better simulate - how our stock portfolio evolves in the next period.

We remember the observation matrix X in which the entries $x_{i,n}$, $i = 1, \ldots, I$ and $n = 1, \ldots, N$, denote the n-th daily log-return of asset i. Again for simplicity reason we assume that $N \mod K = 0$ and define $\delta^K := \frac{N}{K} \in \mathbb{N}$, i.e. we receive a shortened observation matrix

$$\begin{aligned} X^K &= \begin{pmatrix} x_{1,1} & \ldots & x_{1,\delta^K} \\ \vdots & \ddots & \vdots \\ x_{I,1} & \ldots & x_{I,\delta^K} \end{pmatrix} \in \mathbb{R}^{I \times \delta^K} \\ &= (x_{i,n}) \qquad \text{with } i = 1, \ldots, I \text{ and } n = 1, \ldots, \delta^K. \end{aligned}$$

Consequently, the pseudo-observation matrix U changes to

$$U^K = \begin{pmatrix} F_{1,\delta^K}(x_{1,1}) & \dots & F_{1,\delta^K}(x_{1,\delta^K}) \\ \vdots & \ddots & \vdots \\ F_{I,\delta^K}(x_{I,\delta^K}) & \dots & F_{I,\delta^K}(x_{I,\delta^K}) \end{pmatrix} = \begin{pmatrix} u_{1,1} & \dots & u_{1,\delta^K} \\ \vdots & \ddots & \vdots \\ u_{I,1} & \dots & u_{I,\delta^K} \end{pmatrix}$$

Please mind that the empirical distribution functions $F_{i,N}$ have to be revised to F_{i,δ^K}, $i = 1, \dots, I$, and remember that it must hold $N \gg K$ to have enough data.

Algorithm 3.6 for the Gaussian and algorithm 3.7 for the t-copula are applied to this new data setup to calibrate both copulas. Algorithm 3.3 and algorithm 3.4 simulate vectors $v = (v_1, \dots, v_I) \in [0,1]^I$ distributed according to the Gaussian and t-copula, respectively.

To receive a simulated outcome for our portfolio we have to invert the empirical distribution functions F_{i,δ^K} and apply F^{-1}_{i,δ^K} to vector v. Keep in mind that it holds

$$F^{-1}_{i,\delta^K}(u_{i,n}) = (x_{i,n}) \quad \text{for } i = 1, \dots, I \text{ and } n = 1, \dots, \delta^K.$$

This yields the following problem: the inverse empirical distribution function is (of course) not continuous. Thus, the simulated entries v_i of v will not match the entries $u_{i,n}$ of U, $i = 1, \dots, I$ and $n = 1, \dots, \delta^K$.

If δ^K is big enough (what we assumed), F_{i,δ^K} is almost like being continuous and we can find an entry $u_{i,n_i^*} \approx v_i$. We think that the error is neglectable. If an interpolation yields a better result can be researched.

In fact we are searching for the index $n_i^* \in \{1, \dots, \delta^K\}$ which holds

$$\min_{n \in \{1,\dots,\delta^K\}} |u_{i,n} - v_i| = n_i^* \quad \text{for } i = 1, \dots, I.$$

Now, we change vector $v = (v_1, \dots, v_I)^t$ to vector $u^* = (u_{1,n_1^*}, \dots, u_{I,n_I^*})^t$ which can be transformed by means of F^{-1}_{i,δ^K}

$$F^{-1}_{i,\delta^K}(u_{i,n_i^*}) = (x_{i,n_i^*}) \quad \text{for } i = 1, \dots, I$$

and we receive

$$x^* = (x_{1,n_1^*}, \dots, x_{I,n_I^*})^t$$

The vector x^* contains the "simulated" log-returns x_{i,n_i^*} for each asset i, $i = 1, \dots, I$, for the next period, i.e. the next K trading days.

To get the return of our portfolio (assuming that $S_0^i = 1$ and that the portfolio is equally weighted) we compute

$$\lambda^K = \frac{1}{I} \sum_{i=1}^{I} S_0^i \exp(x_{i,n_i^*}) = \frac{1}{I} \sum_{i=1}^{I} \exp(x_{i,n_i^*}).$$

For a Monte Carlo simulation the above described procedure has to be repeated, say, R times. Naturally, it is sufficient to calibrate the copula only once. This changes the formulation as follows. Algorithms 3.3 and 3.4 have to be rerun R times which generates a matrix $V \in [0,1]^{I \times R}$

$$V = \begin{pmatrix} v_{1,1} & \dots & v_{1,R} \\ \vdots & \ddots & \vdots \\ v_{I,1} & \dots & v_{I,R} \end{pmatrix} = (v_{i,r}) \quad \text{with } i = 1, \dots, I \text{ and } r = 1, \dots, R.$$

Searching the index is expanded to

$$\min_{n \in \{1,\dots,\delta^K\}} |u_{i,n} - v_{i,r}| = n_i^* \quad \text{for } i = 1, \dots, I \text{ and } r = 1, \dots, R \tag{3.7}$$

and creates a matrix U^*

$$U^* = \begin{pmatrix} u_{1,n_1^*} & \dots & u_{1,n_R^*} \\ \vdots & \ddots & \vdots \\ u_{I,n_1^*} & \dots & u_{I,n_R^*} \end{pmatrix} = (u_{i,n_r^*}) \quad \text{with } i = 1, \dots, I \text{ and } r = 1, \dots, R.$$

The transformation by means of the inverse function of the empirical distribution function yields

$$X^* = \begin{pmatrix} x_{1,n_1^*} & \dots & x_{1,n_R^*} \\ \vdots & \ddots & \vdots \\ x_{I,n_1^*} & \dots & x_{I,n_R^*} \end{pmatrix} = (x_{i,n_r^*}) \quad \text{with } i = 1, \dots, I \text{ and } r = 1, \dots, R.$$

Again, the matrix X^* consists of the "simulated" log-returns. We revise to

$$\begin{aligned} \Lambda = \exp(X^*) &= \begin{pmatrix} \exp(x_{1,n_1^*}) & \dots & \exp(x_{1,n_R^*}) \\ \vdots & \ddots & \vdots \\ \exp(x_{I,n_1^*}) & \dots & \exp(x_{I,n_R^*}) \end{pmatrix} \\ &= (\lambda_{i,r}) \quad \text{with } i = 1, \dots, I \text{ and } r = 1, \dots, R. \end{aligned}$$

Summing up the columns generates the portfolio return λ_r^K of the r-th Monte Carlo simulation run

$$\lambda_r^K = \frac{1}{I} \sum_{i=1}^{I} \lambda_{i,r} \quad \text{for } r = 1, \dots, R$$

Taking the mean of all simulated portfolio returns λ_r^K induces the expected or simulated portfolio return for the next K trading days:

$$\lambda^K = \frac{1}{R} \sum_{r=1}^{R} \lambda_r \tag{3.8}$$

Important Remarks

1. $K \ll N$. As the data is divided into $\frac{N}{K} = \delta^K$ sets and these are used for the empirical distribution F_{i,δ^K} and its inverse F^{-1}_{i,δ^K}, respectively, it is obvious that N should be much bigger than K. Thus, the error of equation (3.7) is kept small and its approximation is adequate.
2. The computation of matrix U^* is very time-consuming as equation (3.7) has to be run over three indices $i = 1, \ldots, I$, $n = 1, \ldots, \delta^K$ and $r = 1, \ldots, R$.

4 Controlling the Risk

For most financial companies it is nice to have a good approximation for the expected portfolio return for the next K trading days. However, what is much more important to know - especially for departments such as controlling or risk management - is the inherited risk of their (stock) portfolio. The basic mathematical tools for measuring risk are the value at risk (VaR) and the expected shortfall (ES). The ES is a sort of expectation of the VaR. Many companies restrict their operating risk figures to the VaR, we follow suit.

4.1 Value at Risk

4.1 Definition (Value at Risk)
Let $\alpha \in (0, 1)$ be some confidence level. The *VaR* of a (stock) portfolio at confidence level α is given by the smallest number l such that the probability that the portfolio return PFR is below or equal to l is not larger than $(1 - \alpha)$. Mathematically,

$$VaR_\alpha := \inf\{l \in \mathbb{R} : \mathbb{P}[PFR \leq l] \leq 1 - \alpha\}.$$

Thus, the VaR is the quantile function of the return distribution:

$$q_\alpha = F^{-1}_{PFR}(\alpha) = \inf\{l \in \mathbb{R} : F_{PFR}(l) \leq 1 - \alpha\}$$

in which F_{PFR} is the distribution function of the portfolio return. As confidence level generally $\alpha = 0,95$ and $\alpha = 0,99$ are chosen.

We can put the definition in a different, more comprehensive way: With a probability of 99 % ($\alpha = 0,99$) the company can be sure that the portfolio return PFR is bigger than l, i.e. the VaR_α.

4.2 Application

Assume that we should determine the VaR of a stock portfolio of an insurance company. Suppose that this company is interested in their portfolio risk for the

next two weeks ($K = 10$ trading days) and the next month ($K = 20$ trading days). We are on actuarial fields, thus the chosen confidence level is $\alpha = 0,99$.

Somehow we need a simulated distribution function for the portfolio return. By means of chapters 2 and 3 this is easy to achieve. Instead of calculating the mean - equations (2.6) and (3.8) - we can build a distribution function out of the vectors $\hat{S}_r$ and λ_r. This can be done via the MATLAB command `hist` which produces a density function. The result is then used to receive the quantile function from which the VaR is derived.

In MATLAB it is possible to specify the VaR directly with the expression $\texttt{quantile}(\hat{S}_r, 1-\alpha)$ and $\texttt{quantile}(\lambda_r, 1-\alpha)$, respectively.

Therefore, the VaR for COP is

$$VaR_\alpha^C = \texttt{quantile}(\lambda_r, 1-\alpha)$$

and the VaR for COR is

$$VaR_\alpha^C = \texttt{quantile}(\hat{S}_r, 1-\alpha).$$

4.3 Numerical Results

We analyse a stock portfolio of 13 DAX-quoted corporations. Collecting $N = 1788$ daily log-returns of these companies from January 2001 to January 2008, we compute the VaR by means of both COP and COR for $K = 10$ and $K = 20$ trading days. Concerning the structure of the portfolio we have to point out that the assets were not equally weighted. Of course, it is neither possible to set $S_0^i = 1$. For a Monte Carlo simulation of $R = 100.000$ we receive the following results.

VaR computations **-** $R = 100.000$

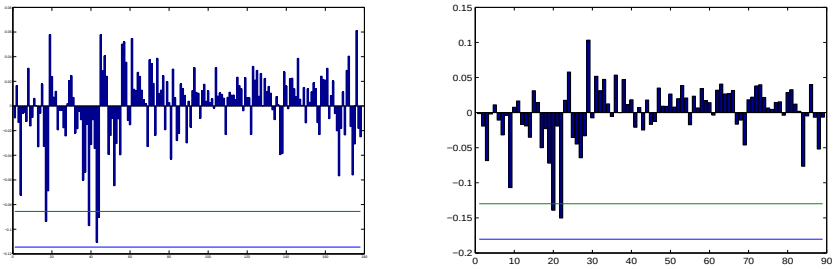
basis	**confidence level**	**COP**	**COP**	**COR**
		Gaussian copula	**t-copula**	
$K = 10$	$\alpha = 0.99$	0,8930	0,8855	0,9144
$K = 20$	$\alpha = 0.99$	0,8295	0,8232	0,8709

We see that the t-copula delivers the lowest VaR on the two and four week basis, closely followed by the results of the Gaussian copula. The difference between COR and COR is remarkable especially in the case of the monthly basis $K = 20$.

4.4 Backtesting

The results are not very meaningful without any comparisons. That's why we compare the 2-weeks and monthly returns from January 2001 to January 2008 with the VaR of COR (green line) and the VaR of COP by means of the t-copula (blue line). That procedure is often called *backtesting*.

It is significant that COR underestimates the risk as well for the 2 week outcomes as for the monthly returns.

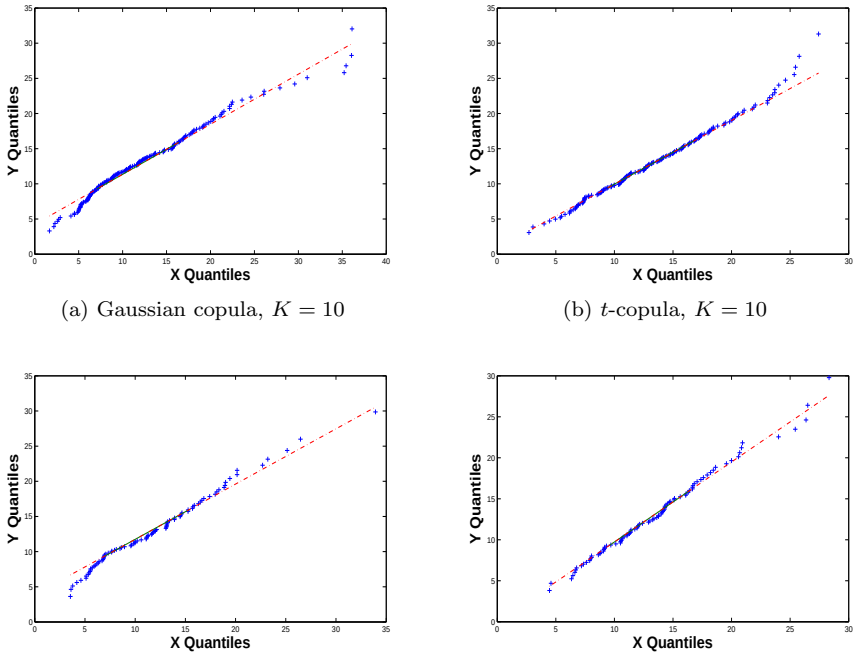


(a) VaR and returns for $K = 10$ (b) VaR and returns for $K = 20$

Figure 4.1: VaR compared to portfolio returns

4.5 Goodness-of-Fit Test

We would like to spare the details and just indicate that we applied a goodness-of-fit test for COP. In brief, this test shows with help of a χ^2-test if the calibrated copula is really suitable for our data. For the Gaussian copula the hypothesis is rejected whereas the hypothesis for the t-copula is supposed to be correct.



(a) Gaussian copula, $K = 10$ (b) t-copula, $K = 10$

(c) Gaussian copula, $K = 20$ (d) t-copula, $K = 20$

Figure 4.2: quantile-quantile-plots for the goodness-of-fit test

In figure 4.2 we see the graphical result of the goodness-of-fit tests. In case of a perfect fit one estimates that all crosses would be on the dotted line. Especially the Gaussian copula does not match the line in the tails (see 4.2a and 4.2c). This is the mainly reason why the hypothesis is rejected for the Gaussian copula.

5 Conclusion

Figure 4.1 is significant. Concerning a backtest, COR would have underestimated the portfolio risk whereas COP captures the risk structure.

Moreover, the Gaussian copula fails the goodness-of-fit test which turns out the t-copula to be an appropriate candidate for the estimation of our stock portfolio risk on a confidence level of $\alpha = 99\%$. It is a popular perception that financial returns are heavy-tailed and often modelled by t-distributions concerning the univariate case: the risk of asset return lies in the lower tails its distribution. Now, for the multivariate case (the portfolio return) correlation is not able not grasp this risk because it does not model the dependency structure in an adequate way. The fact that asset returns are highly mutually dependent especially in times of bear markets is not captured by means of COR. The recent years - in times of financial crisis - showed that multivariate dependency structures must be adhere to.

The COP is particularly suitable for this fact, at least it minds the dependency structure better than the standard approach. The short-coming for financial companies employing COP is obvious: a lower VaR means to allocate more risk capital which cannot be used for other investments.

Bibliography

[1] A. McNeil, R. Frey, P. Embrechts: *Quantitative Risk Management*, Princeton University Press, 2005

[2] K. Tappe: *Ordinary and Lévy Copulas in Finance*, Dissertation, August 2008

[3] D. Brigo, F. Mercurio: *Interest Rate Models - Theory and Practice*, Springer, 2006

[4] M. Günther, A. Jüngel: *Finanzderivate mit MATALB*, Vieweg, 2003

[5] S.E. Shreve: *Stochastic Calculus for Finance II, Continuous-Time Models*, Springer, 2004

[6] A. da Costa Dias: *Copula Inference for Finance and Insurance*, Doctoral Thesis ETH No. 15283, 2004

[7] R. B. Nelsen, *An Introduction to Copulas*, Springer, 2006