Austrian Journal of Statistics
March 2016, Volume 45, 25–44.
http://www.ajs.or.at/
doi:10.17713/ajs.v45i1.87

GCPM: A Flexible Package to Explore Credit Portfolio Risk

Kevin Jakob
University of Augsburg

Matthias Fischer
University of Erlangen-Nuremberg

Abstract

In this article, we introduce the novel **GCPM** package, which represents a **g**eneralized **c**redit **p**ortfolio **m**odel framework. The package includes two of the most popular modeling approaches in the banking industry, namely the CreditRisk$^+$ and the CreditMetrics model, and allows to perform several sensitivity analyses with respect to distributional or functional forms assumptions. Therefore, besides the pure quantification of credit portfolio risk, the package can be used to explore certain aspects of model risk individually for every arbitrary credit portfolio. The way the package is implemented combines a high level of flexibility and performance together with a maximum of usability. Furthermore, the package offers the possibility to apply simple pooling techniques to speed up calculations for large portfolios as well as the opportunity to combine simulation models with a user specified importance sampling approach. The article concludes with a comprehensive example demonstrating the flexibility of the package.

Keywords: credit risk, portfolio model, model risk, R, Monte Carlo simulation, pooling, CreditRisk$^+$, CreditMetrics.

1. Introduction

Banks apply credit portfolio models in order to quantify the amount of economic capital which must be withheld in order to cover unexpected losses caused by credit defaults. As the financial crisis had shown very impressively, the use of quantitative models is always accompanied by a certain amount of model risk which has to be taken into account whenever decisions or price evaluations are based on them. Nowadays, banks are explicitly requested by supervisors to validate their quantitative models and to quantify model risk (see Board of Governors of the Federal Reserve System 2011). Ignoring model risk can lead to wrong management decisions and an underestimation of the true risk. The **GCPM** package addresses both of these issues – quantification of credit risk and an analysis of the underlying model risk.

A great advantage of **GCPM** over other available packages for R (R Core Team 2014), like **QRM** (Pfaff and McNeil 2014) or **CreditMetrics** (Wittmann 2007), is that it utilizes an object oriented approach, where one object consists of a specified model together with all portfolio information and risk figures (once the portfolio loss distribution was estimated). Therefore, it is easy to handle different models (or portfolios) simultaneously without jeopardizing their consistency. As the example in Section 5 will show, performing comparison or sensitivity studies is very simple. In addition, the package is able to deal with large portfolios. On the one hand, portfolios with several thousands of counterparties can be used, whereas in our tests the **CreditMetrics** package was unable to handle more than one hundred portfolio positions. On the other hand, and in contrast to the **QRM** package[1], risk parameters like the probability of default, the loss ratio in case of a default, the exposure and the assignment to a specific industry sector and country affecting the default dependencies can be defined individually for each counterparty. Together with a C++ implementation of the simulation framework, which takes advantage of modern multi-core systems, the package combines flexibility regarding counterparty characteristics and distributional assumptions with good performance and makes it suitable for practical applications. Moreover, for advanced users, simulation models can be combined with self-defined importance sampling techniques and counterparty pooling approaches in order to stabilize simulation results and to increase performance furthermore.

Please note that we will not address any questions regarding the parametrization of the models. In contrast, in order to guarantee a maximum flexibility regarding the distributional assumptions, we have to leave this task up to the user. However, we will provide several examples and demonstrate how already existing packages and basic R functions can be used to construct a parametrization (i.e. a sample from the multivariate sector distribution). For those who are interested in this topic, we refer to Hamerle and Rösch (2006). Please also note that the package focuses on credit risk only with respect to default events, i.e. migration risk is not considered.

The article is organized as follows. A short overview of credit portfolio models together with common notation is given in Section 2. Afterwards, we present the simulation framework and the derivation of risk contributions. The last section contains a hypothetical example, explaining how the package **GCPM** can be used to quantify credit and model risk. Here, starting from the basic CreditRisk$^+$ model (see Credit Suisse First Boston International 1997), which is characterized by certain distributional assumptions, we show how risk figures might change if these assumptions are modified. Along with this, the available functions of the package are introduced including a simple pooling technique which will be useful for homogeneous portfolios (e.g. retail portfolios).

2. Credit portfolio modeling

2.1. Input data, loss distribution and risk figures

The key function of a classical bank is to hand out loans to enterprises or private persons. For reason of simplicity, let us assume that the loan portfolio consists of M loans given to M different counterparties or obligors. In this situation the bank faces the risk that one or more obligors default which means that they are not able or willing to pay back the out-standing amounts (principal and interests) which, in turn, leads to financial losses. The main purpose of a credit portfolio model is to forecast the portfolio loss distribution for the underlying loan portfolio and a fixed time interval, usually one year. Regardless of the specific modeling approach (two of them are introduced in the subsequent sections), every model requires the following set of information on each counterparty i:

- The exposure at the time of default (EAD_i),
- the probability of default (PD_i) for the given time horizon, usually one year,
- the loss given default rate (LGD_i) or recovery rate ($\text{RR}_i = 1 - \text{LGD}_i$, amount recovered

[1] The **QRM** package also provides the possibility to evaluate so-called Bernoulli mixture models but only with respect to the number of defaults. Therefore, analyzing a portfolio with different default probabilities, exposures and sector affiliations is not possible.

through foreclosure or bankruptcy procedures in the event of default, expressed as a percentage of EAD_i) and

- the assignment of the obligor to predefined industry and/or country sectors in order to rebuild the dependence structure of the portfolio.

With this notation, the overall portfolio loss L reads as

$$L := \sum_{i=1}^{M} L_i = \sum_{i=1}^{M} D_i \cdot \mathrm{LGD}_i \cdot \mathrm{EAD}_i,$$

where $D_i \sim \mathrm{Ber}(\mathrm{PD}_i)$ is the default indicator[2] for obligor i (i.e. $\mathrm{PD}_i = P(D_i = 1)$).

Under the assumption that the parameters LGD and EAD are deterministic and the loss distribution F_L has already been derived, the following key figures are required for the bank's risk reporting and management information (see also Figure 1 for a graphical representation):

- Expected loss $\mathbb{E}(L) = \sum_{i=1}^{M} \mathrm{PD}_i \cdot \mathrm{LGD}_i \cdot \mathrm{EAD}_i$.
- Standard deviation $\mathrm{SD}(L) = \left[\sum_{i,j=1}^{M} \mathrm{EAD}_i \cdot \mathrm{EAD}_j \cdot \mathrm{LGD}_i \cdot \mathrm{LGD}_j \cdot \mathrm{Cov}\left(D_i, D_j\right)\right]^{1/2}$.
- Value at Risk $\mathrm{VaR}_\alpha := \inf\{l | F_L(l) \geq \alpha\}$ for a specified level $\alpha \in (0, 1)$.
- Economic capital $\mathrm{EC}_\alpha := \mathrm{VaR}_\alpha - \mathbb{E}(L)$.
- Expected shortfall or expected tail loss $\mathrm{ES}_\alpha := \mathbb{E}(L | L \geq \mathrm{VaR}_\alpha)$.

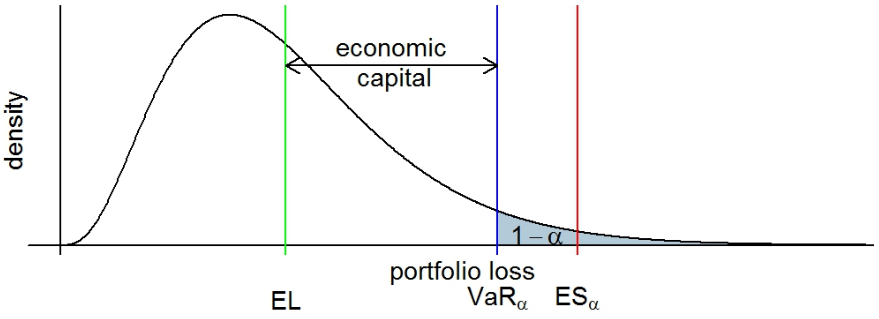


Figure 1: General portfolio loss distribution with risk figures.

In practice, VaR_α and EC_α constitute the relevant risk measures. For example, in the regulatory framework of Basel II (see Basel Committee on Banking Supervision 2006), a loss level of $\alpha = 0.999$ is used to quantify the economic capital.

Whereas the expected loss can be calculated directly from the raw portfolio data, the calculation of the loss distribution in general is a crucial issue. It requires the knowledge of the dependence structure (so-called "default correlations") between the M default indicators $D_1, \ldots, D_M$, where M is typically large. To simplify this problem and reduce the dimension, every counterparty is assigned to one or more out of $K \ll M$ industry and/or country sectors such that dependence between obligors can be traced back to the belonging to same sectors and to the dependence structure between them. The sectors themselves are modeled via a (multivariate) latent variable $\boldsymbol{S}$ which is distributed according to some K-dimensional distribution[3] on $\mathbb{R}^K$.

[2] $\mathrm{Ber}(p)$ denotes the Bernoulli distribution with success probability $p \in (0, 1)$.

[3] How the concrete sector distribution looks like depends on the type of portfolio model (i.e. on the link function) and the calibration, which will be discussed on the following pages.

The **GCPM** package deals with two of the most popular credit portfolio models, namely CreditRisk$^+$ and CreditMetrics, which are briefly summarized in the following subsections. Whereas CreditRisk$^+$ and its generalizations provide an analytic solution under certain restrictive distributional assumptions, CreditMetrics calculates the portfolio loss distribution within a simulation framework which is more flexible but also more time-consuming. For further details on these portfolio models we also refer to Crouhy, Galai, and Mark (2000) or Gordy (2000) who provide an excellent comparative analysis of these models.

2.2. The CreditRisk$^+$ model

The CreditRisk$^+$ model was developed by the Financial Products division of Credit Suisse in 1997, see Credit Suisse First Boston International (1997) for a detailed documentation. It belongs to the class of so-called Poisson mixture models where the intensity of the Poisson distribution (which approximates the Bernoulli distribution of the default indicator D_i) itself is driven by Gamma-distributed random variables. Relying on these specific stochastic assumptions and a discretization of the exposures, it is possible to express the probability mass function of the portfolio loss (or, equivalently, its probability generating function[4]) in a closed analytical form, which is a great advantage of CreditRisk$^+$ and its major difference to its competitors. Hence, even for larger portfolios the risk figures can be obtained within a reasonable run-time.

More formally, the basic idea of the model can be summarized as follows: In a first step, a discretization parameter L_0, called loss unit is introduced. All exposures are approximated by an integer multiple of this unit via $\nu_i = \max\left\{\left\lceil\frac{\mathrm{EAD}_i \cdot \mathrm{LGD}_i}{L_0}\right\rfloor, 1\right\}$, where $\lceil x \rfloor$ denotes the nearest integer value to x. The default probabilities are adjusted such that the discretization does not affect the expected loss. The adjusted PD is given by

$$\widetilde{\mathrm{PD}}_i = \frac{\mathrm{EAD}_i \cdot \mathrm{LGD}_i \cdot \mathrm{PD}_i}{\nu_i \cdot L_0}. \tag{1}$$

As for the calculation of the loss distribution, the loss unit represents the width of the exposure bands on which the marginal probabilities are calculated. For more details, please see Credit Suisse First Boston International (1997, para A 3.2).

Secondly, a further key assumption is to replace the default indicator D_i (naturally Bernoulli distributed) with a Poisson distributed random variable $\tilde{D}_i$ with intensity parameter λ_i. This assumption is necessary in order to compute the portfolio loss distribution analytically. Because, in most cases, λ_i will be very small, the approximation error is not substantial. But if credit quality decreases, the effect of multiple defaults becomes crucial.

Finally, the intensity parameter of each obligor is mapped onto one or more (economic) sectors in order to introduce dependence between the counterparties belonging to the same sector via sector weights. Given a sector realization $\boldsymbol{s} = (s_1, ..., s_K)^T$ of $\boldsymbol{S}$, the conditional default intensity reads as:

$$\lambda_i^S := \widetilde{\mathrm{PD}}_i \left(w_{i,0} + \sum_{k=1}^{K} w_{i,k} s_k \right), \tag{2}$$

with

- the individual adjusted $\widetilde{\mathrm{PD}}_i$,
- individual sector weights $w_{i,k} \in [0,1]$ for obligor i with respect to sector k such that $\sum_{k=1}^{K} w_{i,k} \leq 1$ and the idiosyncratic weight $w_{i,0} = 1 - \sum_{k=1}^{N} w_{i,k}$,

[4] For a discrete random variable X with values in $\mathbb{N}$, the probability generating function (PGF) is defined as $G(z) := \mathbb{E}\left(z^X\right)$.

- sector variables $S_1, \ldots, S_K$ which are assumed to be mutually independent and Gamma distributed with variance[5] σ_k^2 and $\mathbb{E}\left(S_k\right) = 1$ such that $\mathbb{E}(\lambda_i^S) = \mathrm{PD}_i = \lambda_i$.

Under these assumptions, the default correlation between obligor i and j reads as:

$$\mathrm{Cor}\left(\tilde{D}_i, \tilde{D}_j\right) = \frac{\sqrt{\widetilde{\mathrm{PD}}_i \cdot \widetilde{\mathrm{PD}}_j}}{\sqrt{(1-\widetilde{\mathrm{PD}}_i)(1-\widetilde{\mathrm{PD}}_j)}} \sum_{k=1}^{K} w_{i,k} w_{j,k} \sigma_k^2.$$

In order to calculate the probability mass function (PMF) of the portfolio loss, a modified[6] version of the algorithm given in Haaf, Reiss, and Schoenmakers (2003) is used. The algorithm calculates the marginal probabilities that the portfolio loss is equal to $\nu \cdot L_0$ with $\nu \in \mathbb{N}_0$. It stops if a desired level of the cumulative distribution function (CDF) has been reached.

In order to keep the notation simple and comparable to the CreditMetrics model, we will denote the adjusted PD with PD_i as well, instead of $\widetilde{\mathrm{PD}}_i$, in the remainder of this article. Switching back to the original notation does not imply that this approximation is unimportant. Please bear in mind that, if an inappropriately large loss unit L_0 is used, the discretized PDs and hence also the risk figures may be changed noticeably.

2.3. The CreditMetrics model

The CreditMetrics model, described in Gupton, Finger, and Bhatia (1997), is a typical representative of so-called threshold models. The fundamental idea grounds on the firm value model of Merton (1974). For each counterparty i an asset value variable is defined as

$$A_i := \boldsymbol{R}_i^T \boldsymbol{S} + \sqrt{1 - \boldsymbol{R}_i^T \Sigma \boldsymbol{R}_i} \epsilon_i, \tag{3}$$

where $\boldsymbol{R}_i \in \left\{[-1,1]^K \mid \boldsymbol{R}_i^T \boldsymbol{R}_i < 1\right\}$ determines the correlation of i's asset value to the systemic factors $\boldsymbol{S} \sim \mathcal{N}_K(\boldsymbol{0}, \Sigma)$[7]. The idiosyncratic risk is expressed by $\epsilon_i \sim \mathcal{N}(0,1)$ which are independent from each other as well as from $\boldsymbol{S}$. A default occurs if the asset value A_i falls below the default threshold, defined by $\Phi^{-1}(\mathrm{PD}_i)$ where Φ denotes the distribution function of a standard normal variable. Conditioning on a realization $\boldsymbol{s}$ of the systemic factor $\boldsymbol{S}$ the probability of default is given by

$$\mathrm{PD}_i^S = \frac{\Phi^{-1}(\mathrm{PD}_i) - \boldsymbol{R}_i^T \boldsymbol{s}}{\sqrt{1 - \boldsymbol{R}_i^T \Sigma \boldsymbol{R}_i}}. \tag{4}$$

Using formula (3), the default correlation between two counterparties reads as:

$$\mathrm{Cor}\left(D_i, D_j\right) = \Phi_2\left(\Phi^{-1}(\mathrm{PD}_i), \Phi^{-1}(\mathrm{PD}_j), \boldsymbol{R}_i^T \Sigma \boldsymbol{R}_j\right),$$

where $\Phi_2(x_1, x_2, r)$ denotes the distribution function of a bivariate normal distribution with correlation parameter $r \in [-1,1]$ and standard normal margins. The loss distribution is achieved via a Monte Carlo simulation, as described in the next section.

3. Simulation models

Alternatively to the analytical version of the CreditRisk$^+$ model, one can also use a simulation setting. In this case, several distributional assumptions can be modified in order to analyze model sensitivities. By changing the link function (i.e. replacing (2) by (4)), one can also

[5] The variance σ_k^2 can either be estimated from historical default data or using analytical approximations based on the rating specific standard deviation of the PD, see Gundlach (2003).

[6] The loop-structure of the algorithm has been changed to calculate the CDF and the PMF simultaneously.

[7] $\mathcal{N}_K(\boldsymbol{a}, \Sigma)$ denotes the K dimensional normal distribution with mean $\boldsymbol{a}$ and correlation matrix Σ.

switch to a CreditMetrics-like model. Consequently, an analysis of the risk figure sensitivities with respect to the specific link function is also possible. Please take care that the sector drawings (argument `random.numbers` of the `init()` function, see Table 1) meet the correct distributional assumptions of the chosen model, defined via the link function, described in Sections 2.2 and 2.3. E.g. normally distributed sectors are not compatible with the CreditRisk$^+$ setting. For each counterparty, the distribution of the default indicator D_i can be chosen individually between "Bernoulli" (natural choice) or "Poisson" (CreditRisk$^+$- setting) within the portfolio data (see Table 2). Depending on these three elements (sector distribution, link function and default distribution), the basic idea of the simulation framework is to simulate N different portfolio losses. Given these losses, the portfolio loss distribution and risk figures can be estimated via the empirical loss distribution.

3.1. General simulation framework

Given a set of $N \in \mathbb{N}_{>0}$ (multivariate) sector drawings $\boldsymbol{s}^{(1)}, ..., \boldsymbol{s}^{(N)} \in \mathbb{R}^K$ and a portfolio of M counterparties, the general simulation framework of the **GCPM** package is as follows:

Algorithm 1 Basic simulation algorithm

```
For $n = 1, ..., N$ #(simulation loop)
    For $i = 1, ..., M$ #(counterparty loop)
        **Calculate conditional PD:**
        If link.function == "CRP" then
            $\overline{\text{PD}}_i^{(n)} = \text{PD}_i \cdot \left(w_{i,0} + w_i^T \boldsymbol{s}^{(n)}\right)$
        If link.function == "CM" then
            $\overline{\text{PD}}_i^{(n)} = \Phi\left(\frac{\Phi^{-1}(\text{PD}_i) - \boldsymbol{R}_i^T \boldsymbol{s}^{(n)}}{\sqrt{1 - \boldsymbol{R}_i^T \Sigma \boldsymbol{R}_i}}\right)$
        **Draw default:**
        If default$_i$ == "Bernoulli" then
            $D_i \sim \text{Bern}(\overline{\text{PD}}_i^{(n)})$
        If default$_i$ == "Poisson" then
            $D_i \sim \text{Pois}(\overline{\text{PD}}_i^{(n)})$
        **Determine counterparty loss:**
        $L_i^{(n)} = D_i \cdot \text{EAD}_i \cdot \text{LGD}_i$
    **Determine portfolio loss:**
    $L^{(n)} = \sum_{i=1}^{M} L_i^{(n)}$
```

After the simulation, the portfolio losses $L^{(n)}$ are discretized with respect to the loss unit L_0, in order to group losses for the calculation of the probability mass function. The distribution is estimated based on the discretized simulated portfolio losses $\tilde{L} = \left(\tilde{L}^{(1)}, ..., \tilde{L}^{(N)}\right)^T$, i.e.

$$f\left(L = l\right) = \frac{1}{N} \sum_{n=1}^{N} 1_{\{m \,|\, \tilde{L}^{(m)} = l\}}(n), \tag{5}$$

where 1_A denotes the indicator function on set A. For reasons of performance, the simulation algorithm is implemented in C++ and linked to the package via the **Rcpp** package (see Eddelbuettel and François 2011). In order to show the progress status, the **RcppProgress** package (see Forner 2013) is needed as well. In order to avoid errors during the simulation, please ensure that R can allocate enough memory from your operating system, by using the R functions `memory.size()` and `memory.limit()`. In order to increase performance within simulation models, one can also take advantage of multi-core systems. For this purpose, the **parallel** package is required (see Section 5.3.4).

3.2. Adaption of importance sampling techniques

In most cases, the risk figures are based on extreme scenarios with a low probability of occurrence. For instance, if the $\mathrm{ES}_{0.999}$ should be estimated on a basis of 10^3 relevant scenarios (in order to achieve a reliable estimation), one has to perform 10^7 simulations. If portfolios include thousands of counterparties, the simulation will be very time-consuming and it will need lots of memory. With the help of importance sampling techniques, one can "manipulate" the simulation such that extreme scenarios occur more often and tail measures can be calculated on a higher number of simulated losses. Mathematically, importance sampling is just a change of the probability measure from P to P_{IS}. Instead of drawing random numbers from P, one can draw from P_{IS} where the probability of relevant scenarios is higher. The only restriction is that

$$\mathrm{supp}\big(f\big) \subset \mathrm{supp}\left(f_{\mathrm{IS}}\right) \text{ and } f_{\mathrm{IS}}(\boldsymbol{x}) > 0,\ \forall \boldsymbol{x} \in A,$$

where $\mathrm{supp}\big(f_{(\mathrm{IS})}\big)$ denotes the support of the corresponding density functions and A is the set of scenarios the risk measure is calculated on. In order to get an estimator with respect to original measure P, the standard estimator (e.g. for the mean) has to be adjusted by the so-called likelihood ratio

$$\mathrm{LHR}(\boldsymbol{x}_{\mathrm{IS}}) := \frac{f(\boldsymbol{x}_{\mathrm{IS}})}{f_{\mathrm{IS}}(\boldsymbol{x}_{\mathrm{IS}})}, \qquad \text{with} \quad \boldsymbol{x}_{\mathrm{IS}} \sim P_{\mathrm{IS}}.$$

In our case, the standard estimator of the density function (5) changes to

$$f\left(L = l\right) = \frac{1}{\sum_{n=1}^{N} \mathrm{LHR}(L_{\mathrm{IS}}^{(n)})} \sum_{n:\, L_{\mathrm{IS}}^{(n)} = l} \mathrm{LHR}(L_{\mathrm{IS}}^{(n)}). \tag{6}$$

Since a credit portfolio model in general contains a lot of different distributions, also the range of application for an importance sampling algorithm is very wide. For example, one could concentrate on the sector copula. Here, different approaches are possible. For instance, one can simply strengthen the overall level of dependence by increasing the entries of the dispersion matrix of a t-copula or by rising the degrees of freedom (e.g. see Mai and Scherer 2012). Another approach could be to concentrate on those sector drawings where extreme scenarios (e.g. exceeding the 95%-quantile) occur jointly across different sectors (see Arbenz, Cambou, and Hofert 2014). Additionally, one can also use importance sampling on the marginal distributions by shifting the mean or increasing the variance and higher moments or use a more sophisticated approach, see Glasserman and Li (2005).

Please note that, since the sector distribution itself can be defined arbitrarily by the user and the possibilities of importance sampling are manifold, the package does not perform any kind of importance sampling on its own. Instead, the sector drawings (`random.numbers`, see Table 1) can be simulated with a user defined importance sampling approach and passed to a portfolio model together with a vector of likelihood ratios, which will be respected when the loss distribution is calculated. In this way, as in case of the `random.numbers` matrix, the user has maximum flexibility to choose which approach is suitable in his or her situation.

For a more detailed introduction to importance sampling in general we refer to Rubino and Tuffin (2009).

4. Identification of risk drivers

For a portfolio manager, it is important to know which obligors within the portfolio are riskier than others. In order to identify such risk drivers, we briefly introduce different measures which are available in the package for counterparty risk contributions, i.e. contributions to standard deviation σ of the portfolio loss, value at risk, economic capital and expected shortfall. For a detailed derivation of the corresponding formulas in case of the analytical

CreditRisk$^+$ model, please refer to Credit Suisse First Boston International (1997) and Haaf and Tasche (2002).

4.1. Analytical CreditRisk$^+$ model

On counterparty level the following risk contributions (RC) can be calculated:

- **standard deviation:** $\mathrm{RC}_i^\sigma = \frac{\mathrm{PL}_i \cdot \mathrm{PD}_i}{\sigma} \left(\mathrm{PL}_i + \sum_{k=1}^K \sigma_k^2 w_{i,k} \ell_k \right)$, with σ_k denoting the standard deviation of sector k and $\ell_k := \sum_i w_{i,k} \cdot \mathrm{PD}_i \cdot \mathrm{PL}_i$ denoting the expected loss with respect to sector k,

- **VaR$_\alpha$:** $\mathrm{RC}_i^{\mathrm{VaR}_\alpha} = \mathrm{PD}_i \cdot \mathrm{PL}_i \frac{\sum_{k=1}^K w_{i,k} \mathbb{P}(L_k = \mathrm{VaR}_\alpha - \mathrm{PL}_i)}{\mathbb{P}(L = \mathrm{VaR}_\alpha)}$, where L_k denotes the loss in sector k, and

- **ES$_\alpha$:** $\mathrm{RC}_i^{\mathrm{ES}_\alpha} = \mathrm{PD}_i \cdot \mathrm{PL}_i \frac{\sum_{l=\mathrm{VaR}_\alpha - \mathrm{PL}_i}^{\bar{M}} \sum_{k=1}^K w_{i,k} \mathbb{P}(L_k = l)}{\sum_{l=\mathrm{VaR}_\alpha}^{\bar{M}} \mathbb{P}(L = l)}$, where $\bar{M}$ is the maximum portfolio loss a probability is calculated on (depending on `alpha.max`, see Table 1).

Please note that depending on the loss unit L_0 used for exposure discretization and the number of obligors within the portfolio, VaR contributions may be zero for some counterparties because they do not default in the single VaR-event. Therefore, it is reasonable to consider contributions to ES rather than VaR. Because ES is based on the upper tail of the loss distribution rather than a single loss level, the mentioned problem does not occur using ES contributions.

Finally, for all these measures it holds that the individual contributions sum up to the measure calculated on portfolio level. Therefore, one can also analyze contributions, for example on sector level (e.g. business lines or countries) by simply aggregating the corresponding counterparty contributions.

4.2. Simulation models

Within the simulation framework, expected shortfall contributions can be calculated. For this purpose, one has to define a loss threshold `loss.thr`>0, which should be lower to the corresponding VaR but not too low in order to stress memory usage not too much. If the portfolio loss $L^{(n)}$ in scenario n is above `loss.thr`, all counterparty losses $L_i^{(n)}$ are stored. Counterparty risk contributions to ES on level $\alpha \in (0,1)$ are then calculated as:

$$\mathrm{RC}_i^{\mathrm{ES}_\alpha} = \frac{1}{\sum_{n \in N_\alpha} \mathrm{LHR}(L_{\mathrm{IS}}^{(n)})} \sum_{n \in N_\alpha} \mathrm{LHR}(L_{\mathrm{IS}}^{(n)}) \cdot L_{\mathrm{IS}}^{(n)}, \tag{7}$$

where $N_\alpha := \left\{ n = 1, ..., N \,|\, L^{(n)} \geq \mathrm{VaR}_\alpha \right\}$ denotes the set of all ES$_\alpha$- relevant scenarios. Similar to the analytical CreditRisk$^+$model it holds that $\sum_{i=1}^M \mathrm{RC}_i^{\mathrm{ES}_\alpha} = \mathrm{ES}_\alpha$.

For other tail measures (VaR and EC) the risk contributions are calculated with the same approach but with respect to another level $\tau \in (0,1)$ such that $\mathrm{ES}_\tau = \mathrm{VaR}_\alpha$ or $\mathrm{ES}_\tau = \mathrm{EC}_\alpha$, respectively. Therefore, risk contributions to VaR and EC are approximated by risk contributions to ES but on a lower level τ. Using the ES approach instead of a direct calculation with respect to VaR or EC, risk contributions are much more stable because of the higher number of scenarios used for the calculation.

Since the portfolio loss distribution is not continuous, level τ for VaR/EC contributions is chosen such that ES$_\tau$ is as close as possible to VaR$_\alpha$ or EC$_\alpha$, respectively. If deviations are greater or equal to 0.01% an appropriate message comes up.

5. The GCPM package

The main component of the package is the S4 class `GCPM`. Besides this class there are some additional functions, in particular for object creation. The class represents the whole portfolio model framework. It contains all model specifications as well as the portfolio and the loss distribution once it is estimated. In case of a simulation model, losses on counterparty level are also stored depending on a predefined threshold `loss.thr` (see Table 1)

In the next sections we give a detailed overview of the most important features. A complete list of all slots is available in the help pages of the package (see `?GCPM`). The following examples are based on the CreditRisk$^+$ framework. Please note that the same analysis can be also performed within a CreditMetrics framework.

5.1. General structure

The overall structure of the package is very intuitive. At first, one has to initialize a new model using the `init()` function. The process of creation is as follows. Passing the input parameters for a new model to the function creates a new object of class `GCPM` with the specified settings (after some plausibility checks). For example:

```
library("GCPM")
sec.var <- c(0.2, 0.3, 0.4)  #arbitrary sector variances
names(sec.var) <- c("A", "B", "C")  #assign sector names to variances
CRP.classic <- init(model.type = "CRP", loss.unit = 50000, alpha.max = 0.9999,
    sec.var = sec.var)

##    Generalized Credit Portfolio Model
##    Copyright (C) 2015 Kevin Jakob & Dr.  Matthias Fischer
##
##    This program is free software; you can redistribute it and/or
##    modify it under the terms of the GNU General Public License
##    version 2 as published by the Free Software Foundation.
##
##    This program is distributed in the hope that it will be useful,
##    but WITHOUT ANY WARRANTY; without even the implied warranty of
##    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
##    GNU General Public License for more details.
##
##    You should have received a copy of the GNU General Public License
##    along with this program; if not, write to the Free Software
##    Foundation, Inc., 51 Franklin Street, Fifth Floor,
##    Boston, MA  02110-1301, USA.
```

The above code generates a `GCPM` model, named `CRP.classic` with the given attributes. For some slots of the `GCPM` class, default values (e.g. for `alpha.max`) are provided, but they are not necessarily the best choice. Considering this, one should better choose them individually for each portfolio according to exposures, number of counterparties, and hardware restrictions. Depending on the `model.type`, different arguments have to be provided. A summary is given in Table 1 below.

After creating a new portfolio model, one can analyze a credit portfolio using the `analyze()` method. In case of an analytical CreditRisk$^+$ model, the loss distribution will be calculated by using the algorithm described in Haaf *et al.* (2003). For simulation models, the simulation described in Algorithm 1 is used. If loss levels are provided via the parameter `alpha`, tail measures are calculated automatically with respect to those levels. Otherwise, one can calculate those measures afterwards with the corresponding methods as shown in the following examples. The portfolio data frame has to follow the structure described in Table 2.

`model type`	Parameter	Description
`CRP`	`alpha.max`	...is a numeric value between 0 and 1 defining the maximum CDF-level which will be computed.
	`sec.var`	...is a named numeric vector defining the sector variances. The names have to correspond to the sector names given in the portfolio.
`simulative`	`link.function`	... is a character value, specifying the type of the link function (`"CRP"` corresponds to equation (2) and `"CM"` to (4)).
	`N`	... is a numeric value, defining the number of simulations. If `N` is greater than the number of scenarios provided via `random.numbers`, scenarios are reused.
	`seed`	... is a numeric value used to initialize the random number generator. If `seed` is not provided, a value based on the current system time will be used. Therefore, the results are truly random in this case.
	`loss.thr`	... is a numeric value specifying a lower bound for portfolio losses to be stored in order to derive counterparties' risk contributions.
	`random.numbers`	... is a matrix with sector drawings. The columns represent the different sectors, whereas the rows represent the scenarios. The column names must correspond to the sector names used in the portfolio.
	`LHR`	... is a numeric vector of length equal to `nrow(random.numbers)` defining the likelihood ratio of each scenario. If not provided, all scenarios are assumed to be equally likely.
	`max.entries`	... is the number of scenarios stored to calculate risk contributions. The value should be set in consideration of the amount of available memory.

Table 1: Arguments for `init()` in case of a simulation and an analytical model.

Number	**Name**	**Business**	**Country**	**EAD**	**LGD**	**PD**	**Default**	**A**	**B**	**C**
1	Name 1	Energy	US	358475	0.989	0.001	Bernoulli	1	0	0
2	Name 2	IT	DE	1089819	0.608	0.003	Bernoulli	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2: Structure of the portfolio data frame.

5.2. Analyzing credit risk: A first example

Based on a portfolio distributed with the package (in the package's `data` folder) consisting of 3000 counterparties and three industrial sectors, we offer an example to show how the package works. We start from the `CRP.classic` model defined in the previous section.

```
# first example
library("GCPM")
# importing portfolio
data("portfolios")
# analyzing the portfolio (Poisson defaults)
CRP.classic <- analyze(CRP.classic, portfolio.pois)

## Importing portfolio data....
## 3 sectors ...
## 3000 counterparties (0 removed due to EAD=0 (0), lgd=0 (0), pd<=0 (0) pd>=1 (0))
##
## Portfolio statistics....
## Loss unit:  50 K
## Portfolio EAD:1.5 B
## Portfolio potential loss:772.28 M
## Portfolio expected loss:130.69 M(analytical)
## Diversifible risk:  7.67 M  Systematic risk:  41.41 M
## Portfolio standad deviation:42.11 M(analytical)
## Calculate the loss distribution till 0.9999-confidence level is reached.




##
## Calculation completed...
## Reached level of confidence:  0.9999001591125 ( iterations actually done:  7073 )
##
## Calculating risk measures from loss distribution....
## Expected loss from loss distribution:  130.65 M
(deviation from EL calculated from portfolio data:  -0.03%)
## Exceedance Probability of the expected loss:0.454820959078588
## Portfolio mean expected loss exceedance:  167.18 M
## Portfolio loss standard deviation:42.04 M
```

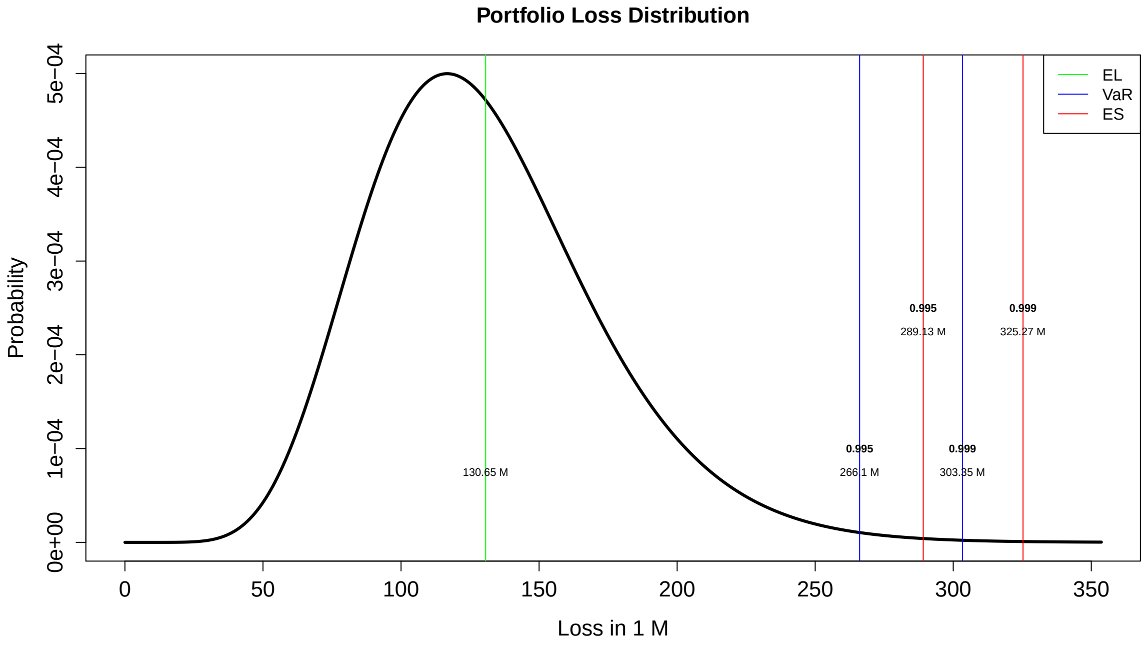
After deriving (or simulating) the loss distribution, risk measures like VaR, EC or ES can be calculated with the help of the corresponding methods.

```
alpha <- c(0.995, 0.999)  #levels for tail measures
VaR(CRP.classic, alpha)

## [1] 266100000 303350000
```

The probability mass function for the loss distribution together with indicators for tail measures can be plotted by using the `plot()` function. The second argument defines the scale of the horizontal axis.

```
plot(CRP.classic, 1e+06, alpha)
```



Finally, one can calculate risk contributions in order to identify particular positions in the portfolio driving EC, VaR or ES.

```
RC.cont <- data.frame(Name = name(CRP.classic), EC.cont(CRP.classic, alpha),
    VaR.cont(CRP.classic, alpha), ES.cont(CRP.classic, alpha))
RC.cont[1:3, ]

##      Name  EC.0.995  EC.0.999 VaR.0.995 VaR.0.999  ES.0.995  ES.0.999
## 1 Name  1 125170.64 170476.28 312088.98 378068.30 352293.31 419188.26
## 2 Name  2  15620.00  18351.49  22802.99  23955.69  23547.25  24518.83
## 3 Name  3  15006.43  20350.73  37055.76  44838.89  41798.39  49689.53
```

5.3. Modifying distributional assumptions

Besides the pure quantification of credit risk, the package assists in analyzing different aspects of model risk related to distributional or functional forms assumptions. Starting from the classic CreditRisk$^+$ model (example of Section 5.2) we will show how the package can be used to build much more flexible models and how to quantify model risk similar to the analyses done by Jakob and Fischer (2014), Fischer and Mertel (2012) or Fischer and Kaufmann (2014).

A key element for this is the `random.numbers` matrix which represents the (multivariate) sector distribution. Since the dimension of this matrix depends on the portfolio, i.e. on the number of sectors used, the matrix has to be defined by the user. Additionally, the sector distribution (expressed by `random.numbers`) also heavily depends on the economic sectors it is associated with. I.e. the sector copula and the marginal distributions may be very different across geographical regions and industries. Since this is a very crucial issue, which has also a significant impact on the risk figures, this matrix must be defined by the user (i.e. no default value is provided). Furthermore, in this way, the user has maximum flexibility to define the sector distribution according to his or her needs. However, a few examples are given on the following pages. For more information about the question of sector parametrization we refer to Hamerle and Rösch (2006) or Dorfleitner, Fischer, and Geidosch (2012).

Checking for simulation error

At first we will check if the results of the simulation model correspond to the analytical one. Therefore, we create a matrix of random numbers, which are independently Gamma distributed with mean equals one and variance given by `sec.var`, which we can pass to the argument `random.numbers` of the `init()` function.

```
# generating random numbers for sector distribution
N <- 1e+05  # number of simulations
set.seed(1)  # for reproducibility
rn.indep.gamma <- matrix(NA, N, 3, dimnames = list(1:N, c("A", "B", "C")))
for (i in 1:3) rn.indep.gamma[, i] <- rgamma(N, shape=1/sec.var[i], scale=sec.var[i])
```

Now we switch to a simulation model but with the same distributional assumptions as in the classic model.

```
CRP.pois <- init(model.type = "simulative", link.function = "CRP", N = N,
    loss.unit = 1000, random.numbers = rn.indep.gamma, seed = 1)

## Warning in init(model.type = "simulative", link.function = "CRP", N = N, :
No LHR provided for simulative model, assuming equally likelihood for all szenarios.
## Warning in init(model.type = "simulative", link.function = "CRP", N = N, :
loss.thr is not finite.  Risk contributions (to EC, VaR and ES) will be not available.
```

Because we did not provide a vector with likelihood ratios, a corresponding warning is displayed. Similarly, we get another warning because the parameter `loss.thr` was not set (default value: infinity). Hence no counterparty specific losses are stored, which means that risk contributions will be not available. Since in this example we only want to calculate risk figures on the overall portfolio level, we can proceed to analyze the given portfolio.

```
CRP.pois <- analyze(CRP.pois, portfolio.pois)

## Importing portfolio data....
## 3 sectors ...
## 3000 counterparties (0 removed due to EAD=0 (0), lgd=0 (0), pd<=0 (0) pd>=1 (0))
##
## Portfolio statistics....
## Loss unit:  1 K
## Portfolio EAD:1.5 B
## Portfolio potential loss:772.28 M
## Portfolio expected loss:130.69 M(analytical)
## Starting simulation (1e+05simulations )
## Simulation finished
##
## Calculating loss distribution...
## Calculating risk measures from loss distribution....
## Expected loss from loss distribution:  130.6 M
(deviation from EL calculated from portfolio data:  -0.06%)
## Exceedance Probability of the expected loss:0.45422
## Portfolio mean expected loss exceedance:  167.06 M
## Portfolio loss standard deviation:41.97 M
```

A comparison of risk figures shows that the simulation error is less than 1% in our example.

```
VaR(CRP.classic, alpha) / VaR(CRP.pois, alpha)  # check if risk figures are close

## [1] 1.005308 1.008863
```

Quantifying the "Poisson effect"

Since the classic CreditRisk$^+$ model assumes that counterparties' defaults are Poisson and not Bernoulli distributed, there is a tendency to overestimated risk figures, especially for portfolios of bad quality[8]. To quantify this effect, we switch the default distribution within the portfolio data frame.

```
# Quantifying the Poisson effect
portfolio.bern <- portfolio.pois  #copy portfolio
portfolio.bern$Default <- "Bernoulli"  #change to Bernoulli distributed defaults
CRP.bern <- CRP.pois  #duplicate model framework
CRP.bern <- analyze(CRP.bern, portfolio.bern)  #analyze Bernoulli portfolio

## Importing portfolio data....
## 3 sectors ...
## 3000 counterparties (0 removed due to EAD=0 (0), lgd=0 (0), pd<=0 (0) pd>=1 (0))
##
## Portfolio statistics....
## Loss unit:  1 K
## Portfolio EAD:1.5 B
## Portfolio potential loss:772.28 M
## Portfolio expected loss:130.69 M(analytical)
## Starting simulation (1e+05simulations )
```

[8] In general, the Poisson distributions serves as a good approximation of the Bernoulli distribution only if the intensity parameter is very low.

```
## Simulation finished
##
## Calculating loss distribution...
## Calculating risk measures from loss distribution....
## Expected loss from loss distribution:  130.51 M
(deviation from EL calculated from portfolio data:  -0.13%)
## Exceedance Probability of the expected loss:0.45437
## Portfolio mean expected loss exceedance:  166.67 M
## Portfolio loss standard deviation:41.5 M
```

In our case, the overestimation due to the Poisson effect is around 2% - 3%.

```
VaR(CRP.pois, alpha) / VaR(CRP.bern, alpha)  #compare risk figures

## [1] 1.017682 1.028880
```

Introducing sector dependencies

One of the most crucial assumptions of the classic CreditRisk$^+$ model is the assumption of independent sectors. Within an analytical framework extensions to correlated sectors are proposed by Fischer and Dietz (2011) and Giese (2003). Here, we use dependent random variables (`random.numbers` matrix) to introduce dependence between sectors. Before we continue with our examples, a brief introduction to the concept of copulas is given, which will be used within the following example.

A copula is a multivariate distribution function on the d-dimensional unit hypercube with uniform one-dimensional margins. By using copulas, an arbitrary multivariate distribution can be decomposed into its one-dimensional margins and the dependence structure. Following Sklar's Theorem (see Sklar 1959) it holds that for any multivariate distribution function F on $\mathbb{R}^d$ with univariate margins F_i a unique function $C : \times_{i=1}^{d} \mathrm{Im}(F_i) \to [0, 1]$ exists, such that $F(\boldsymbol{x}) = C(F_1(x_1), ..., F_d(x_d))$ for all $\boldsymbol{x} \in \mathbb{R}^d$. In reverse, if F_i are arbitrary univariate distribution functions and C is a copula function, then the function F defines a valid multivariate distribution function. Famous representatives of copulas are the Gaussian and the t-copula. For further details on this topic we refer to Joe (1997) and Nelson (2006).

Within our next example, the **copula** package (see Hofert, Yan, Maechler, and Kojadinovic 2014) is used in order to create an exchangeable Gaussian copula for the sector drawings. The margins are again Gamma distributed with parameters equal to the former example.

```
# Introducing sector dependencies
require("copula")
require("methods")
gauss <- normalCopula(param = 0.7, dispstr = "ex", dim = 3)  # define copula
paramMargins <- list()  # define margins
for (i in 1:3) paramMargins[[i]] <- list(shape = 1/sec.var[i], scale = sec.var[i])
# define multivariate sector distribution
mvdf <- mvdc(copula = gauss, margins = rep("gamma", 3), paramMargins = paramMargins)
rn.gauss.gamma <- rMvdc(N, mvdc = mvdf)
colnames(rn.gauss.gamma) <- c("A", "B", "C")
```

With the help of the new matrix `rn.gauss.gamma` we can simulate a model with dependent sectors.

```
CRP.bern.gauss <- init(model.type = "simulative", link.function = "CRP",
    N = N, loss.unit = 1000, random.numbers = rn.gauss.gamma, seed = 1)
CRP.bern.gauss <- analyze(CRP.bern.gauss, portfolio.bern)
```

As one would expect, the comparison of VaR figures shows that the risk clearly rises (by over 30% in our example) in case of dependent sectors.

```
VaR(CRP.bern.gauss, alpha) / VaR(CRP.bern, alpha)  #compare risk figures

## [1] 1.330193 1.363756
```

A great advantage of the package is that one can use any arbitrary portfolio with any possible dependence structure and quantify the markup in his or her special case.

Exchanging both the sector copula and the margins

In our next example, we demonstrate how the sensitivity of risk figures with respect to distributional assumptions (i.e. sector copula and margins) can be quantified. The possibilities are only restricted by the set of distributions (univariate and multivariate) available in R. In order to increase performance, we use multiple cores (i.e. 4 cores) for the Monte Carlo simulation. Therefore, the package **parallel** is required.

```
# using a T-copula with Gamma margins
tcop <- tCopula(param = 0.7, dispstr = "ex", df = 4, dim = 3)
paramMargins <- list()
for (i in 1:3) paramMargins[[i]] <- list(shape = 1/sec.var[i], scale = sec.var[i])
mvdf <- mvdc(copula = tcop, margins = rep("gamma", 3), paramMargins = paramMargins)
rn.t.gamma <- rMvdc(N, mvdc = mvdf)
colnames(rn.t.gamma) <- c("A", "B", "C")
# initialize models and analyze portfolio
CRP.bern.t <- init(model.type = "simulative", link.function = "CRP", N = N,
    loss.unit = 1000, random.numbers = rn.t.gamma, seed = 1)
CRP.bern.t <- analyze(CRP.bern.t, portfolio.bern, Ncores = 4)
```

Again, a comparison of both models shows that specific assumptions of the sector copula may affect the risk figures. In our case, the markup is around 5% if a t-copula with 4 degrees of freedom is used instead of a Gaussian copula.

```
VaR(CRP.bern.t, alpha) / VaR(CRP.bern.gauss, alpha)  #compare risk figures

## [1] 1.032988 1.057561
```

For a more detailed analysis regarding the sector copula within the CreditRisk$^+$ and the CreditMetrics framework we also refer to Fischer and Jakob (2015). When exchanging sector distributions, please take care of the specific model assumptions, e.g. that the mean equals one within the CreditRisk$^+$ framework or the quantification of the default threshold $\Phi^{-1}(\mathrm{PD})$ in a CreditMetrics type model.

In the next step, we switch the marginal sector distributions from a Gamma distribution to a log-normal distribution.

```
# using a T-copula with logN margins
paramMargins <- list()
for (i in 1:3) paramMargins[[i]] <- list(meanlog = -0.5 * log(1 + sec.var[i]),
    sdlog = sqrt(log(1 + sec.var[i])))
mvdf <- mvdc(copula = tcop, margins = rep("lnorm", 3), paramMargins = paramMargins)
rn.t.logN <- rMvdc(N, mvdc = mvdf)
colnames(rn.t.logN) <- c("A", "B", "C")
CRP.bern.t.logN <- init(model.type = "simulative", link.function = "CRP",
    N = N, loss.unit = 1000, random.numbers = rn.t.logN, seed = 1)
CRP.bern.t.logN <- analyze(CRP.bern.t.logN, portfolio.bern, Ncores = 4)
```

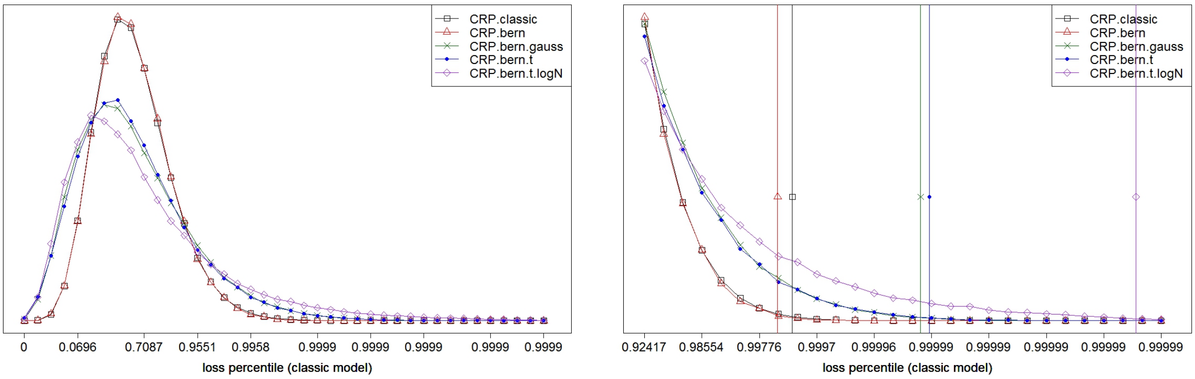


Figure 2: Loss distributions of example models together with indicators for $\mathrm{VaR}_{0.999}$.

Since in contrast to the Gamma distribution, the log-normal distribution is heavy tailed, the values for $\mathrm{VaR}_{0,995}$ and $\mathrm{VaR}_{0,999}$ increase by approximately 7% and 10%, respectively.

```
VaR(CRP.bern.t.logN, alpha) / VaR(CRP.bern.t, alpha)  #compare risk figures

## [1] 1.068731 1.094763
```

Please note that the same analysis can be carried out within a CreditMetrics like default model by using `link.function="CM"`.

The loss distributions of the examples are shown in Figure 2. The x-axis of both charts represent the loss percentile in the classic CreditRisk$^+$ model. The right one exhibits the upper tail of all distributions together with vertical lines indicating the value of $\mathrm{VaR}_{0.999}$ in each model, clearly demonstrating how risk increases if assumptions related to the sector distribution are modified.

5.4. Pooling

Finally, we show how a simple pooling approach can be used in order to speed up calculations. For this purpose, the package's `data` folder contains a prepared portfolio containing three pools (see Table 3). Here, all counterparties within the same sector and a potential loss (PL$=$ EAD $\cdot$ LGD) below 200,000 are grouped into one pool.

Let M_{Pool} denote the number of counterparties within one pool. Then for each pool, the values for EAD, LGD and PD are determined via the following formulas:

- $\mathrm{EAD}_{Pool} = \frac{1}{M_{Pool}} \sum_{i \in Pool} \mathrm{EAD}_i$ (average EAD per counterparty)
- $\mathrm{LGD}_{Pool} = \frac{\sum_{i \in Pool} \mathrm{EAD}_i \mathrm{LGD}_i}{\mathrm{EAD}_{Pool} M_{Pool}}$ (weighted average LGD per counterparty)
- $\mathrm{PD}_{Pool} = \frac{\sum_{i \in Pool} \mathrm{EAD}_i \mathrm{LGD}_i \mathrm{PD}_i}{\mathrm{EAD}_{Pool} \mathrm{LGD}_{Pool}}$ (average number of defaults within the pool)

Since the pooling criteria (i.e. potential loss threshold, sector membership) depend on the underlying portfolio as well as the desired accuracy, we have to leave this task up to the user. Additionally, in order to achieve good approximation results for the risk figures, advanced users may consider more sophisticated pooling techniques, for example based on certain PD and PL ranges, the pool loss standard deviation or the well-known Herfindahl index regarding the counterparty exposures as presented in Gordy (2003). Please note that in case of a CreditMetrics-like link function (i.e. if `link.function="CM"`), which includes the distribution function of a standard normal distribution, default intensities greater or equal to one are not supported.

Number	Name	Business	Country	EAD	LGD	PD	Default	A	B	C
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
100000	Pool A	misc	misc	342298,63	0,246	91,44	Poisson	1	0	0
200000	Pool B	misc	misc	332533,03	0,243	90,29	Poisson	0	1	0
300000	Pool C	misc	misc	334227,25	0,237	87,76	Poisson	0	0	1

Table 3: Structure of the pooled portfolio data frame.

With the help of this technique, we can reduce the number of portfolio positions in our example by over 50%. Using the model with a t-copula and Gamma distributed margins, we check the accuracy of risk figures if the pooled portfolio is used.

```
# Pooling
CRP.bern.t.pool <- analyze(CRP.bern.t, portfolio.pool, Ncores = 4)

## Importing portfolio data....
## 3 sectors ...
## 1483 counterparties (0 removed due to EAD=0 (0), lgd=0 (0), pd<=0 (0) pd>=1 (0))
##
## Portfolio statistics....
## Loss unit:  1 K
## Portfolio EAD:991.99 M
## Portfolio potential loss:648.54 M
## Portfolio expected loss:130.69 M(analytical)
## Starting simulation (1e+05simulations )
## Parallel computing on 4 cores (no progress bar)
## Simulation finished
##
## Calculating loss distribution...
## Calculating risk measures from loss distribution....
## Expected loss from loss distribution:  130.78 M
(deviation from EL calculated from portfolio data:  0.07%)
## Exceedance Probability of the expected loss:0.43296
## Portfolio mean expected loss exceedance:  187.4 M
## Portfolio loss standard deviation:63.37 M
```

Although the simulation of Poisson random variables is more time-consuming than those of Bernoulli ones, the simulation time (using `Ncores=1`) can be reduced by around 50% on our computer (Intel Core i7, 3.6GHz, calculation time: 7s to 3.8s). In combination with the option `Ncores=4` we can reduce the computation time by another 70% such that a simulation that needs 7s (without pooling, single core) can be done in just 1s.

```
VaR(CRP.bern.t.pool, alpha) / VaR(CRP.bern.t, alpha)  #compare risk figures

## [1] 0.9974762 1.0047332
```

Comparing the risk figures of the pooled version with those of the ordinary simulation on two loss levels, we observe that the deviations are not substantial (around 1%) for our hypothetical portfolio.

Please note that the criteria and thresholds for the pooling have to be determined individually for each portfolio and model in order obtain tolerable approximation errors.

6. Summary

Quantifying credit portfolio risk is an essential part of risk controlling of financial institutions. For this purpose, the **GCPM** package offers the opportunity to choose between a CreditRisk$^+$

and a CreditMetrics-type model within a default framework. The examples show that, because of the flexible structure, the package helps to analyze the sensitivity of risk figures if distributional assumptions are modified and therefore to quantify aspects of model risk as well. In order to increase the performance further, simulation models can be combined with user specific importance sampling techniques and pooling approaches. The combination of these possibilities and a fast implementation of the simulation core in C++ together with the capability of parallel computing makes the package a powerful tool which also allows to perform calculations on portfolios with a large number of counterparties.

For more information about the package, especially about the individual methods, please have a look at the help pages provided in the package (e.g. `?init`).

Session Info

```
## R version 3.2.1 (2015-06-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=German_Germany.1252  LC_CTYPE=German_Germany.1252
## [3] LC_MONETARY=German_Germany.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.1252
##
## attached base packages:
## [1] methods   stats     graphics  grDevices utils     datasets  base
##
## other attached packages:
## [1] copula_0.999-13 GCPM_1.2        knitr_1.10.5
##
## loaded via a namespace (and not attached):
##  [1] ADGofTest_0.3      Rcpp_0.11.6        lattice_0.20-31
##  [4] mvtnorm_1.0-2      stabledist_0.7-0   pspline_1.0-16
##  [7] grid_3.2.1         stats4_3.2.1       formatR_1.2
## [10] magrittr_1.5       evaluate_0.7       highr_0.5
## [13] stringi_0.5-2      Matrix_1.2-1       tools_3.2.1
## [16] stringr_1.0.0      RcppProgress_0.2.1 parallel_3.2.1
## [19] gsl_1.9-10
```

Acknowledgment

We would like to thank two anonymous referees for their helpful comments on an early version, which helped to improve this article.

References

Arbenz P, Cambou M, Hofert M (2014). "An Importance Sampling Algorithm for Copula Models in Insurance." *Submitted manuscript*. URL http://arxiv.org/pdf/1403.4291.

Basel Committee on Banking Supervision (2006). "International Convergence of Capital Measurement and Capital Standards: A Revised Framework Comprehensive Version."

Board of Governors of the Federal Reserve System (2011). "Guidance on Model Risk Management." *Technical report*, Federal Reserve System. URL http://www.federalreserve.gov/bankinforeg/srletters/sr1107.htm.

Credit Suisse First Boston International (1997). "CreditRisk$^+$ A Credit Risk Management Framework." *Technical report*. URL http://www.csfb.com/institutional/research/assets/creditrisk.pdf.

Crouhy M, Galai D, Mark R (2000). "A Comparative Analysis of Current Credit Risk Models." *Journal of Banking & Finance*, **24**(1), 59–117.

Dorfleitner G, Fischer M, Geidosch M (2012). "Specification Risk and Calibration Effects of a Multifactor Credit Portfolio Model." *The Journal of Fixed Income*, **22**(1), 7–24.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. URL http://CRAN.R-project.org/package=Rcpp.

Fischer M, Dietz C (2011). "Modeling Sector Correlations with CreditRisk$^+$: The Common Background Vector model." *The Journal of Credit Risk*, **7**(4), 23–43.

Fischer M, Jakob K (2015). "Copula-Specific Credit Portfolio Modeling." In K Glau, M Scherer, R Zagst (eds.), *Innovations in Quantitative Risk Management*, pp. 129–145. Springer-Verlag.

Fischer M, Kaufmann F (2014). "An Analytic Approach to Quantify the Sensitivity of CreditRisk$^+$ with Respect to its Underlying Assumptions." *The Journal of Risk Model Validation*, **8**(2), 23–37.

Fischer M, Mertel A (2012). "Quantifying Model Risk within a CreditRisk$^+$ Framework." *The Journal of Risk Model Validation*, **6**(1), 47–76.

Forner K (2013). ***RcppProgress**: An Interruptible Progress Bar with **OpenMP** Support for C++ in R packages*. R package version 0.1, URL http://CRAN.R-project.org/package=RcppProgress.

Giese G (2003). "Enhancing CreditRisk$^+$." *RISK*, **16**(4), 73–77.

Glasserman P, Li J (2005). "Importance Sampling for Portfolio Credit Risk." *Management Science*, **51**(11), 1643–1656.

Gordy MB (2000). "A Comparative Anatomy of Credit Risk Models." *Journal of Banking & Finance*, **24**(1), 119–149.

Gordy MB (2003). "A Risk-Factor Model Foundation for Ratings-Based Bank Capital Rules." *Journal of Financial Intermediation*, **12**(3), 199–232.

Gundlach VM (2003). "Basics of CreditRisk$^+$." In M Gundlach, F Lehrbass (eds.), *CreditRisk$^+$ in the Banking Industry*, pp. 7–24. Springer-Verlag.

Gupton GM, Finger CC, Bhatia M (1997). "CreditMetrics: Technical Document." *Technical report*. URL http://www.defaultrisk.com/_pdf6j4/creditmetrics_techdoc.pdf.

Haaf H, Reiss O, Schoenmakers J (2003). "Numerically Stable Computation of CreditRisk$^+$." In M Gundlach, F Lehrbass (eds.), *CreditRisk$^+$ in the Banking Industry*, pp. 69–77. Springer-Verlag.

Haaf H, Tasche D (2002). "Credit Portfolio Measurements." *GARP Risk Review*, **7**, 43–47.

Hamerle A, Rösch D (2006). "Parameterizing Credit Risk Models." *Journal of Credit Risk*, **2**(4), 101–122.

Hofert M, Yan J, Maechler M, Kojadinovic I (2014). ***copula**: Multivariate Dependence with Copulas*. R package version 0.999-12, URL http://CRAN.R-project.org/package=copula.

Jakob K, Fischer M (2014). "Quantifying the Impact of Different Copulas in a Generalized CreditRisk+ Framework: An Empirical Study." *Dependence Modeling*, **2**(1), 1–21.

Joe H (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall/CRC.

Mai JF, Scherer M (2012). *Simulating Copulas*. Imperial College Press.

Merton RC (1974). "On the Pricing of Corporate Debt: The Risk Structure of Interest Rates." *The Journal of Finance*, **29**(2), 449–470.

Nelson RB (2006). *An Introduction to Copulas*. Springer-Verlag.

Pfaff B, McNeil A (2014). ***QRM**: Provides R-language Code to Examine Quantitative Risk Management Concepts*. R package version 0.4-10, URL http://CRAN.R-project.org/package=QRM.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Rubino G, Tuffin B (2009). *Rare Event Simulation Using Monte Carlo Methods*. John Wiley & Sons.

Sklar A (1959). "Fonctions de Répartition à n Dimensions et leurs Marges." *Publications de l'Institute de Statistique de l'Université de Paris*, **8**, 229–231.

Wittmann A (2007). ***CreditMetrics**: Functions for Calculating the CreditMetrics Risk Model*. R package version 0.0-2, URL http://CRAN.R-project.org/package=CreditMetrics.

Affiliation:

Kevin Jakob
Universität Augsburg
86135 Augsburg, Germany
E-mail: Kevin.Jakob@Student.Uni-Augsburg.de

Matthias Fischer
Department of Statistics and Econometric
Universität Erlangen-Nürnberg
90402 Nürnberg, Germany
E-mail: Matthias.Fischer@fau.de
URL: http://www.statistik.wiso.uni-erlangen.de/

Austrian Journal of Statistics http://www.ajs.or.at/
published by the Austrian Society of Statistics http://www.osg.or.at/
Volume 45 *Submitted:* 2014-10-31
March 2016 *Accepted:* 2015-08-11