Proceedings of the 7th WSEAS International Conference on Mathematics & Computers in Business & Economics, Cavtat, Croatia, June 13-15, 2006 (pp35-40)

A computational solution for generating claim sizes using copula theory

VICTORIA RIVAS, ALFREDO CUESTA
C.E.S. Felipe II
UCM
C/Capitan, 39. 28300, Aranjuez
SPAIN

Abstract: - One of the contributions of copula theory to insurance sector is the consideration of dependent risks when fixing the premium. In this article a software tool developed for Matlab 7.0 is introduced. The goal of the tool is to simulate claim size of different branches or risk factors using Monte Carlo method and including multivariate distribution functions Copulas. The tool is powerful because Matlab environment is very suitable to the high computational load involved. Besides a graphical user interface is provided so that a previous knowledge of either copula theory or Matlab command should be as less as possible.

Key-Words: - Risk management, risk analysis, copula theory, insurance/reinsurance, Monte Carlo, Matlab toolboxes.

1 Introduction

Traditionally the starting point for insurance sector when fixing the premiums has been the hypothesis of independence and the law of large numbers [1]. In certain cases, assuming independence between risk factors has led to not so well or even wrong decisions. For instance, natural disasters or catastrophic losses in which different branches or risk factors are involved and where the dependence between these factors had not been taken into account.

Besides, in the present day it is necessary to increase the risk typology due to the complex environment, globalisation and new risks that were not considered before such as climatic changes or terrorism. Thus insurance/reinsurance sector has been motivated to enlarge the range of products that allow an increase in the hedge capacities. By doing so, companies can optimise their risk transfer policy improving their financial stability. The current quest for a sound methodological basis for integrated risk management also raises the issue of correlation and dependence[2].

For either those products traditionally used or the alternative ones, such us ART [3-6], it is necessary to analyse the effect on the premium when assuming dependency between risks; and therefore to include copula theory into the analysis.

The word ‘copula’ first appeared in the statistics literature in 1959 [7] although similar ideas and results can be traced back to Hoeffding [8]. Copulas allow us to construct models which go beyond the standard ones at the level of dependence. As a consequence the following advantages are also obtained:

-- Fixing the premium takes into account the convergence of diverse dependent risk factors.
-- Adds the Kendall non-linear correlation factor, which keeps invariant before strictly increasing transformations of random variables [9].
-- Continuous distribution marginal functions can have a structure separated from the structure of the copula [7].

In the process of fixing the premium is necessary to have a representative sample of the claim sizes. With them as historic data, the use of computers in the simulation and analysis of possible scenarios is quite suitable due to the large amount of operations needed. Computational solutions are available for different software packages like S-Plus [10] or Matlab [11]. Matlab provides an integrated development and simulation environment as well as core specialised to carry out a huge computational load; such as that needed when copula distribution functions are included. Thus, some recent efforts have been done with this software. Perkins and Lane [12] distribute a set of functions for carrying out Monte Carlo simulations using the statistical toolbox and including copulas. Patton contributes in [13] with a code for testing the Normal copula, the Student's t copula, and the Normal copula with time-varying conditional correlation matrix.

This paper contributes with a set of functions, together with a graphical user interface, named

Proceedings of the 7th WSEAS International Conference on Mathematics & Computers in Business & Economics, Cavtat, Croatia, June 13-15, 2006 (pp35-40)

Insurance/Copula Toolbox (ICT) developed in Matlab 7.0. With the ICT functions the simulation of future claim sizes using Monte Carlo method with copula theory can be carried out easily and without having the statistic toolbox installed.

The rest of the paper is organised as follows. In section 2 the basis about copula theory is briefly explained. In section 3 ICT is introduced and described. Besides, an example of use is shown. Finally the conclusions are given.

2 Dependent risks and copula theory

A copula function is a n-variable distribution with domain $[0,1]^n$ and such that their marginal distributions are uniform in (0,1). A non-uniform invariant marginal distribution function can also be introduced in the copula after an *ad hoc* transformation [14].

The dependence structure of the random variables is determined by relations between uniform distributions which are in the copula. Thus, provided the marginal distributions and a set of uniformly distributed data, a copula can simulate random variables [9].

Family of copulas is quite large. However, in insurance sector, dependence on large claims scale is more frequent than on small one. Such asymmetry is better evaluated with Arquimidiean family [9]. Arquimedian copulas are interesting in practice because they are very easy to construct, but still we obtain a very rich family of dependence structures [15]. Thus four of them are included in ICT: HRT(survival Clayton), Gumbel, Clayton and Frank.

In order to choose the best suitable copula, it is necessary to have at least two data series of amounts of payments. These amounts of payments can be either real or even simulated by Monte Carlo method.

So let X={x_i} and Y={y_i}, with i=1,2,...,n, be two data series of n amounts of payments linked to two different branches of insurance. The steps for modelling the process of choosing a copula function are the following:

1. Transform the data series X and Y in U and V, being U={u_i} and V={v_i} , with i=1,2,...,n . This normalisation must be done using the empirical distribution function given by:

$$F_n(x) = \frac{1}{n+1}\sum_{i=1}^{n} 1_{\{x_i, x\}} = \frac{card\{x_i < x\}}{n+1} \qquad (1)$$

2. Estimate the parameter a of the copula using the maximum likelihood method given by

$$\hat{a} = \arg\max\left(\ell(a)\right)$$
$$\ell(a) = \sum_{i=1}^{n} \ln c\left(u_i, v_i; a\right) \qquad (2)$$

   where c is density function of the copula and $\hat{a}$ is the estimated parameter. Function c for each copula is given in Apendix.

3. Choose a set of marginal functions we want to evaluate.
4. Estimate the parameters of every marginal function using the maximum likelihood method.
5. Choose one marginal distribution for each data series considering HQ criteria, given by

$$HQ = r\ln\left(\frac{n}{2\pi}\right) + \left|\left[\ell\left(\hat{\theta}\right)\right]\right| \qquad (3)$$

   where r is the number of parameters, n is the size of the data series and $\ell(\hat{\theta})$ is the maximum likelihood function for a given marginal. [14]

6. Estimate the parameters of the distribution function obtained from the copula and the two marginal functions chosen previously using the method of maximum likelihood. Function ℓ is the given by:

$$\ell\left(\theta_x, \theta_y, a\right) = \sum_{i=1}^{n}\ln\left(c\left(F_X\left(x_i;\theta_x\right), F_Y\left(y_i;\theta_y\right); a\right)\right) + \sum_{i=1}^{n}\ln f_X\left(y_i;\theta_x\right) + \sum_{i=1}^{n}\ln f_Y\left(y_i;\theta_y\right) \qquad (4)$$

   where c stands for the density function of the copula, F_X and F_Y are the distribution function marginals for two data-vectors x and y, f_X and f_Y are density function of marginals and (θ_x, θ_y, a) are the estimators.

7. Verify the goodness of fit by χ^2.

3 The computational solution: ICT

This section summarizes the functions developed in Matlab and included in ICT. These functions have been grouped and listed in the following tables, giving the name and the input and output arguments for each one.

Table 1. Batteries

Name	*Inputs*	*Outputs*
bat_copula	index	name
bat_marginal	index	name, var, const
bat_Fdistrib	index	name, var, const
bat_Finv	x, iM, params	v

Table 2. Evaluations

Name	*Inputs*	*Outputs*
eval_copulas	x, y	out
eval_marginal	x, y	matRx, matRy
eval_mix	x, y, indm1, indm2, indcop	out

Table 3. Functions objective

Name	*Inputs*	*Outputs*
Lobj	x	y
fobjective	x	y
LMIXobj	x	p

Table 4. Simulating and normalising

Name	*Inputs*	*Outputs*
genUV	vec1, vec2, typeC, a	u, v
normdat	x	u, table

Table 5. Tests

Name	*Inputs*	*Outputs*
Tkendal	a_max, index	Tk
testCHI2	x, vlim, iM, params, alfa	chi2

Table 6. Functions

Copula functions	*Marginal, Distribution and Inverse distribution functions*	
functionL…	marHQ…, FDIS… , FINV…	
1) HRT	1) paralog,	5) bineg
2) Gumbel	2) logis,	6) expon
3) Frank	3) pareto,	7) weibull
4) Clayton	4) lognor,	8) poisson

Table 1 presents four batteries of functions. These are copula and marginal density functions, and marginal distribution and inverse distribution functions. The first three batteries return a string with the name of the function selected in terms of an index. Besides when the function is in bat_marginal or bat_Fdistrib, it also returns the name of the variables and the constants. Bat_Finv returns a vector v when receives a data-vector x, the index of the inverse distribution function and a vector of parameters.

Functions grouped under Table 2 are the motor of the ICT. When running eval_copula it retrieves the name of each copula from the battery bat_copula and evaluates it by maximum likelihood with data-vectors x and y supplied. Finally reports the index of each copula together with the maximum of parameter a, the point at which the maximum is located and the τ of Kendall. Running eval_marginal has the same effect, but now it retrives from the bat_marginal and evaluates by HQ criterion. It also reports with the index of each marginal, the maximum HQ and the point at which is located. For eval_mix a copula together with two marginal must be picked. Giving their indexes and two data-vectors x and y the eval_mix returns the maximum of the estimator of the copula function with such marginal functions and the point where it is reached.

The optimisations carried out by eval_{copulas, marginal, mix} need objective functions, shown in Table 3. Thus, Lobj is the container for L_ functions, fobjective is for margHQ_ functions and LMIXobj is for the copula function with the two marginal, picked and given to eval_mix. Both L_ and margHQ_ are shown in Table 6. The objective functions are called inside the eval_ functions. Therefore the user has to take them into account only if he wants to maximise any of the functions explained above.

Function genUV, in Table 4, is needed to obtain a couple of vectors u and v from two vectors $vec1$ and $vec2$ uniformly distributed with a given copula function and its parameter a. Then u and v will be used to obtain the claims in a Monte Carlo simulation. Function normdat, also in Table 4, is needed to normalise an uniformly distributed vector. It should be used before eval_copula.

Table 5 show the χ^2 test that can be run to measure the goodness of fit of the marginal functions to the data-vectors. Tkendal calculate the τ of Kendall which measures the goodness of the copula function chosen.

Finally Table 6 lists all the copula and marginal functions as well as the distribution functions and the inverse distribution functions included together with their indexes. The so named ‘functionL…’ implement (2), and so ‘margHQ…’ with (3).

3.1 Example

The following script shows how to use the ICT functions (in bold) defined above for simulating two vectors of claims. The lines preceded by ‘%’ are comments.

```
%-1st: load the data vectors 'x' and 'y'
% in which the claims of two different
% branches are stored. Let us assume that
% the archive is named "dataclaims"
    load dataclaims;
%-2nd: Obtain 'u' and 'v'
% normalising 'x' and 'y'
    u=normdat(x); v=normdat(y);
%-3rd: Run evaluation of copula and
% marginal functions
```

```
    ReportCopula=eval_copula(u,v)
    ReportMargin=eval_marginal(u,v)
%>>The reports are displayed in the
%  screen.
%-4th: Let us assume the copula picked
% was HRT and the best marg. functions
% were Lognormal both for 'u' and 'v'.
% Then we create the copula function
% with the marginal functions by doing:
    ReportMix=eval_mix(x,y,4,4,1)
% where 4 is the index of Lognormal and
% 1 is the index of HRT
%>>The report shows the estimators
%  both of the copula and the marginal
%  functions that must be used to
%  simulate the claims.
%-5th: Obtain two uniformly distributed
% vectors 'v1' and 'v2' with for example
% 1000 elements
    v1=rand(1000,1);v2=rand(1000,1);
%-6th: Obtain vectors 'u' and 'v' using
% the copula chosen with the parameter
% obtained in step 4. Let us assume it
% was 'a=1.12'.
    [u,v]=genUV(v1,v2,1,1.12);
%-7th: Obtain the claims 'x' and 'y'
% using the marginal functions, also
% with the parameters calculated in
% step 4. Let us assume the following
% parameters for X {p1=58.4, p2=1.72}
% and for Y {p1=37.2, p2=0.12}.
    x=bat_Finv(u,4,[58.4 1.72]);
    y=bat_Finv(v,4,[37.2 0.12]);
%-8th: In order to have characteristic
% values, 'x' and 'y' are multiplied by
% 10000. This number can be chosen
% properly in terms of the claims.
% So finally doing...
    claims=10e5*[x y]
%>>Returns the vector 'claims' with 2
%  columns and 1000 rows.
```

The risk manager now can use this simulated claims in any insurance/reinsurance product to calculate the premium. This can be carried out in Matlab. Nevertheless the claims can also be copied and pasted in a plain-text archive and then imported by most of the software used in the companies.

3.2 Graphical User Interface (GUI)

The commands can turn into despair for user not familiarised with Matlab. Consequently ICT is provided with a GUI, depicted in Fig. 1, so that the use commands is minimised.

To run the GUI the user just has to execute the command 'DemoChoose'. The following steps are very simple:

1. Press 'X, Y' button
2. Press 'U, V' button if a scatter plot is desired
3. Press 'Eval_copula' button

>> *This will show the copula with maximum â*

4. Press 'Eval_marginal' button

>> *This will show the marginal functions for X and Y with maximum estimators*

5. Press 'Eval_mix' button

>> *This will calculate the estimators of the copula with the marginal functions chosen. They are shown below together with a measure of the goodness-of-fit.*

Besides a variable named 'claims' is created and shown in the command window. A sample of reports is also shown in Fig. 2 and 3.

3.3 Remarks

ICT is a prototype designed for Matlab 6.5 and later. However the GUI will only work in Matlab 7.0 so this last one is recommended.

Optimization toolbox is required since we have used 'fminsearch' function in maximum likelihood and HQ criteria. Thus ICT is sensitive to the accuracy of this function. Optimization process is independent of ICT so it can be changed, for instance using genetic algorithms. However this method usually require a previous knowledge of the objective function and it is not so straightforward.

Another advantage is that no more special toolboxes are requires since ICT includes an own version of distribution and inverse distribution functions, some even not available in the last version of Matlab.

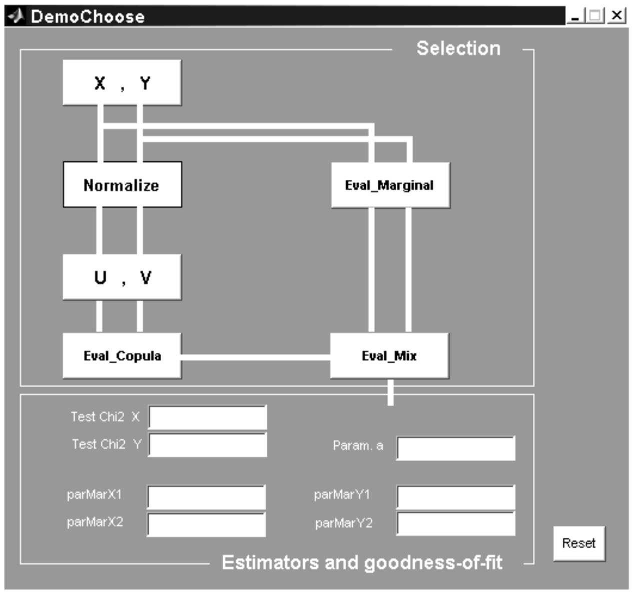

Fig. 1. GUI "DemoChoose"

```
----------------- COPULA ------------------

ans =

[            1.,              2.,
[    1.499052832,    1.346828906,
[    104.4826315,    103.4046939,
[    .2501184521,    .2575151934,


          3.,             4.]
 2.319335651,    539734167.1]
 68.34698182, .6891283138e-5]
-.1464915743, .9263819672e-9]


--------------- MARGINAL ----------------

ans =

[           1.,           2.,           4.]
[  277.2741061,  347.1819408,  427.5800049]
[  3.696462757,  1.989884432,  1.002013077]
[  .6748518750,  .4277693854, -1.003563407]



ans =

[           1.,           2.,           4.]
[  277.2741060,  347.1819406,  427.5800041]
[  3.696502199,  1.989952197,  1.002040999]
[  .6748556339,  .4277665509, -1.003509023]
```

Fig. 2. Reports of a simulation

```
claims=

[ .2713531817e4, .999469966e4]
[ .5352554979e4, .1572854941e4]
[ .2039659543e4, .9342252591e4]
[ .4154545e4, .455925110e4]
[ .2554689186e4, .111220074e4]
[ .4683952468e4, .150572604e4]
[ .3561266034e4, .419207292e4]
[ .5076065945e4, .1728016025e4]
[ .4709988706e4, .984408170e4]
[ .427088420e4, .1506318322e4]
[ .203540003e4, .442870798e4]
[ .4618616706e4, .1235205769e4]
[ .264441533e4, .9243996688e4]
[ .444160053e4, .140733188e4]
[ .2278155818e4, .141282286e4]
[ .215258640e4, .1339559830e4]
[ .2443047816e4, .9726265352e4]
[ .2565641354e4, .1088431878e4]
[ .26567144e4, .986186406e4]
[ .3534350866e4, .1326518124e4]
[ .2311872540e4, .1644430759e4]
[ .241508248e4, .9717229453e4]
[ .3992872204e4, .164459152e4]
[ .493435624e4, .9170762887e4]
[ .220171942e4, .9123127425e4]
```

Fig. 3. Claims obtained

4 Conclusion

Increasing complexity of insurance and reinsurance products has raised recently to the actuarial interest in the modelling of dependent risks. In this field, copula theory is becoming of great interest. On the other hand this complexity and novelty can also be seen as a drawback when introducing in the companies. Computers are a great help and different tailor-made solutions are frequently taken.

In this paper a general solution is proposed. We present a Matlab toolbox named ICT which includes a GUI. The goal of the toolbox is to provide functions for simulating claim sizes of different branches or risk factors using Monte Carlo method and including multivariate distribution functions Archimedean copulas. The main advantage of ICT is that it needs neither specialised statistical functions nor optimisation functions. With this toolbox a huge number of scenarios can be simulated.

References:

[1] Blum, P. Dias, A. Embrechts, P. "The ART of dependence modelling: the latest advances in correlation analysis". *Alternative Risk Strategies*. Publishing House: Morton Lane, Risk Books, London, pp. 339-356, 2002.

[2] Embrechts, P. McNeil, A. Strauman, D. "Correlation and dependence in risk management properties and pitfalls" *in Risk Management: Value at Risk and Beyond.*, Ed. M.A.H. Dempster, Cambridge University Press, Cambridge, pp. 176-223. 2002

[3] Shimpi, P. "A integrating coporate risk management". *FSA, CFA.* Publishing House: Texere New York 2001.

[4] Wang, S. "Aggregation of Correlated Risk Portfolios: Models and Algorithms". *Proceedings of the Casualty Actuarial Society*, Volume LXXXV, 1998.

[5] SWISS RE SIGMA. "Alternative Risk Transfer (ART) for corporations: a passing fashion or risk management for the 21st century?". Swiss Re Sigma. No. 2. 1999.

[6] SWISS RE SIGMA. "The picture of ART". Swiss Re Sigma. No. 1. 2003.

[7] Sklar, A. "Fonctions de repartition à n dimensions et leurs marges". *Publ. Inst. Statist. Univ. Paris 8*, 1959, pp.229-231.

[8] Hoeffding, W. "Masstabinvariante correlations-theorie" *Schriften des Mathematischen Seminars und des Instituts für Angewandte Mathematik der Universität Beril*, No.5 pp. 181-233.

[9] Embrechts, P. Lindskong, F. McNeil A. "Modelling Dependence with Copulas and Applications to Risk Management". *Department of Mathematics ETHZ. Zurich. Switzerland*, 2001. http://www.math.ethz.ch/finance

[10] Carmona, R.A. "Statistical analysis of financial data in S-Plus" Springer, New York. 2004

[11] Matlab website at http://www.mathworks.com/

[12] Perkins, P. Lane, T. "Monte-Carlo simulation in Matlab using Copulas". *Matlab News & Notes*

http://www,mathworks.com/nn_copulas
Nov. 2003.
[13] Patton, A. website: http://fmg.lse.ac.uk/~patton/
[14] Charles Levi, M. "Tempêtes: Etude des dépendance entre les branches Auto et Indendie avec la théorie des copulas". *Guy Carpenter. Univ. Louis Pasteur Strasbrourg. Magistére d'Actuariat*, 2001
[15] Wüthrich, M. V. "Extreme Value Theory and Arquimediean Copulas".*Scandinavian Actuarial Journal.* No.104, 2004, pp. 211-228.

Apendix

The density function $c(u,v)$ is given by:

-- HRT

$$c(u,v)=(1+\frac{1}{a})\left((1-u)^{-1/a}+(1-v)^{-1/a}-1\right)^{-a-2}\left[(1-u)(1-v)\right]^{1-1/a}$$

-- Gumbel

$$c(u,v)=\left(u^{-1}v^{-1}\left[(-\ln u)^{a}+(-\ln v)^{a}\right]^{-2-2/a}\right)\cdot$$
$$\cdot\left((\ln u)(\ln v)\right)^{a-1}\cdot$$
$$\cdot\left(1+(a-1)\left[(-\ln u)^{a}(-\ln v)^{a}\right]^{-1/a}\right)\cdot$$
$$\cdot\exp\left\{-\left[(-\ln(u))^{a}+(-\ln(v))^{a}\right]^{1/a}\right\}$$

-- Frank

$$c(u,v)=-a(e^{-a})\left[\frac{1+e^{-a(u+v)}}{(e^{-au}e^{-av}+e^{-a})^{2}}\right]$$

-- Clayton

$$c(u,v)=(1+\frac{1}{a})(uv)^{-1-\frac{1}{a}}\left(u^{-1/a}+v^{-1/a}-1\right)^{-a-2}$$