Some methods for approximating the distribution of a randomly weighted sum of random variables

Denis Enăchescu* and Raluca Vernic

Abstract

In this paper we present and compare two methods (a neural network and a kernel method) for approximating the distribution of the random variable financial situation of an insurance company. This variable is seen as a randomly weighted sum of independent random variables (representing the net payouts during several years), while the weights are the random discount factors for each year. Such a sum may appear e.g. when evaluating the ruin probability.

1 Introduction

In traditional risk theory, standard techniques for determining the cumulative distribution function (cdf) of aggregate claims are based on independence assumptions (e.g. Panjer and De Pril recursions). Assuming independence is very convenient from the mathematical point of view, but recently, the introduction of stochastic financial aspects in actuarial models led to the study of sums of dependent random variables. More precisely, randomly weighted sums of independent random variables may appear in financial and insurance contexts e.g. when evaluating the ruin probability.

Let $(X_n)_{n\geq 1}$ be a sequence of independent and identically distributed (i.i.d.) real-valued random variables with generic random variable X and common distribution $F_X = 1 - \overline{F}_X$, and let $(\theta_n)_{n\geq 1}$ be another sequence of positive random variables, independent of the sequence $(X_n)_{n\geq 1}$. Recently, in a general probabilistic setting, Goovaerts et al. (2004) investigated the tail probabilities of the randomly weighted sum

$$S_n = \sum_{k=1}^{n} \theta_k X_k, \qquad n \geq 1, \tag{1}$$

and their maxima. We will now present an actuarial model that reduces to (1).

We consider a discrete time risk model in which negative returns are taken into account. We denote by X_n the random variable net payout during period n (also called *the insurance risk*), by Y_n the random variable discount factor from time n to time $n-1$ (also called

*Faculty of Mathematics, University of Bucharest; e-mail: denaches@fmi.unibuc.ro

the financial risk), by x the initial surplus and by U_n the discounted value of the surplus process accumulated till time $n, n = 1, 2, \ldots$. Then U_n satisfies the recurrence equation

$$U_0 = x \geq 0, \quad U_n = x - \sum_{k=1}^{n} X_k \prod_{i=1}^{k} Y_i = x - W_n, \qquad n = 1, 2, \ldots,$$

where $W_n = \sum_{k=1}^{n} X_k \prod_{i=1}^{k} Y_i$ denotes the total discounted amount of losses at time n, with $W_0 = 0$. We notice that W_n is reduced to model (1) with $\theta_k = \prod_{i=1}^{k} Y_i$, θ_k being a product of positive random variables. The finite and infinite time ruin probabilities are defined as, respectively,

$$\psi(x; n) = \Pr\left(\min_{0 \leq k \leq n} U_k < 0 \;\middle|\; U_0 = x \right) = \Pr\left(\max_{0 \leq k \leq n} W_k > x \;\middle|\; U_0 = x \right) \tag{2}$$

and

$$\psi(x) = \lim_{n \to \infty} \psi(x; n) = \Pr\left(\max_{0 \leq k < \infty} W_k > x \;\middle|\; U_0 = x \right). \tag{3}$$

Goovaerts et al. (2004) derived asymptotic expressions for the tail probabilities of (1) and its maxima, and, based on them, considered two particular distributions of X for which the finite time ruin probability can be asymptotically evaluated. In the first particular case, Goovaerts et al. (2004) assumed that the random variables $(X_n)_{n \geq 1}$ are i.i.d. with common Pareto(α, β) distribution for some $\alpha > 0$ and $\beta > 0$, with density function

$$f_X(x) = \frac{\alpha \beta^{\alpha}}{x^{\alpha+1}}, \qquad x > \beta,$$

and that for each $n = 1, 2, \ldots$, the vector $(Y_1, ..., Y_n)$ follows an n-dimensional lognormal distribution with parameters $-\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n$. Therefore, under some regularity conditions, we have from Goovaerts et al. (2004) that for $n = 1, 2, \ldots$

$$\Pr\left(\sum_{k=1}^{n} \theta_k X_k > x \right) \sim \overline{F}_X(x) \sum_{k=1}^{n} \exp\left\{ -\alpha \mu_{(k)} + \frac{1}{2} \alpha^2 \sigma_{(k)} \right\}, \tag{4}$$

where $\mu_{(k)} = \sum_{i=1}^{k} \mu_i$, $\sigma_{(k)} = \sum_{1 \leq i,j \leq k} \sigma_{ij}$ and "$\sim$" means *asymptotically* (for two positive infinitesimals $a(x)$ and $b(x)$, we write $a(x) \sim b(x)$ if $\limsup_{x \to \infty} \frac{a(x)}{b(x)} = 1$).

Several other recent papers investigated the sum (1) using different approximative methods. For example, based on the concept of comonotonicity, Dhaene et al. (2002a, b) presents some approximations, upper and lower bounds for this sum. Like in Goovaerts et al. (2004), another approach is the asymptotic estimation of the tails of (1). For previous results of this kind, under different assumptions (the most common being the independence of $(Y_n)_{n \geq 1}$), see e.g. Nyrhinen (1999), Tang & Tsitsiashvili (2003, 2004).

In this paper we design and compare two methods for approximating the distribution of S_n: a neural network method (section 2) and a kernel method (section 3). Comparatively with the kernel method, we show that the neural network obtains a prediction error at least

1,000 times lower on the tail of the distribution as estimated by the asymptotic expression deduced by Goovaerts et al. (2004).

Neural networks are a particular case of Adaptive Nonlinear Models (ANMs). In the actuarial literature, an overview of ANMs including their explanation, description of key features, modeling considerations and comparison with linear models was realized by Shapiro (2000) and Shapiro & Gorman (2000). Also, neural networks were already used in different insurance contexts. For example, Brockett et al. (1994) introduced a neural network model to predict insurer's insolvency, while Brockett et al. (1998) used a feed-forward neural network model in a study of automobile claims fraud. For details on neural networks, the following books can be consulted: Fine (1999), Hassoun (1995), Hertz et al. (1991), Ripley (1996).

Though not enough used, kernel methods are considered by Klugman et al. (1998, section 2.11.1) as an important topic in the actuarial literature. The reason is their usefulness when the number of observations is large. In this case, the empirical distribution function is likely to be fairly smooth and smoothing it a bit more with a kernel method may eliminate the necessity of using a parametric family. For example of actuarial application to graduation see Gavin et al. (1993, 1994). More recently, Bolancé et al. (2003) applied a semiparametric transformation method to estimate actuarial loss functions with a lognormal type of shape and a Pareto type of tail. Also, Nielsen et al. (2004) used a class of local linear kernel density estimators to model old age mortality.

In the following we will consider a fixed value for n (the number of time periods under study), hence, for simplicity, we drop the index n from S_n. We denote by $(s_1, ..., s_{vol})$ the observed (generated) values of S, where vol is the sample size (in the case of generated values the random generator is initialized with the value given in $state$). Hence, the empirical cdf of S is defined as

$$\hat{F}_{emp}(x) = \frac{1}{vol} \sum_{i=1}^{vol} I(x - s_i),$$

where $I(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$. We also denote by $F_S(x) = \Pr(S \leq x)$, i.e. the cdf of S.

2 The Multilayer Perceptrons model

Assuming that we dispose of the set $(s_1, ..., s_{vol})$ of observed/generated values, in this section we consider the following model for the cdf of S,

$$\hat{F}_{MLP}(x) = \sum_{i \in Ind} c_i K(P_i, x), \tag{5}$$

where $Ind \subseteq \{1, ..., volr\}$, $volr \ll vol$, $c_i > 0$ for all $i \in Ind$, $\sum_{i \in Ind} c_i = 1$. Also,

$$K(P_i, x) = \frac{1}{1 + e^{-\gamma(x - P_i)}},$$

with $\gamma \geq 0$, $P_i = minS + \dfrac{maxS - minS}{volr - 1}(i-1)$, $i = 1, ..., volr$, $minS = \min(s_1, ..., s_{vol})$ and $maxS = \max(s_1, ..., s_{vol})$. Initially $Ind = \{1, ..., volr\}$ and $c_i = \dfrac{1}{volr}$ for all i.

We propose as estimating model *the neural network model*, which, as mentioned in the introduction, is a particular nonlinear statistical model. The proposed model is a two-stage regression model represented by a network diagram as in Figure 1.

The units in Figure 1 are called *perceptrons*. The units in the middle of the network are called *hidden* because their outputs are not directly observed. Notice that the activation function of the hidden perceptrons is the logistic function K, and that the activation function of the output perceptron is the identity function. Hence, a neural network with several layers of perceptrons, also called *multilayer perceptrons network* (MLP), can be thought of as a nonlinear generalization of the linear regression model. By introducing the nonlinear transformation it greatly enlarges the class of linear models.

Note that model (5) has exactly the same form as the projection pursuit regression, PPR (see Hastie et al. (2002)). The difference is that the PPR model uses more general nonparametric functions while the neural network uses a far simpler function based on the logistic function.

Finally, we note that the name "neural networks" derives from the fact that it was first developed as model for the human brain. Each perceptron represents a neuron and the connections (links in Figure 1) represent synapses.

The MLP model has unknown parameters that we shall call *weights*, and denote their complete set by Θ. We seek values for these weights that make the model fit the training data well. Hence, in our case, Θ consists of the values $(c_i)_{i \in Ind}$; these values and indirectly the set Ind will be estimated. As already mentioned, initially the set $Ind = \{1, ..., volr\}$; after each performance of the algorithm, only the strictly positive c_i's are kept, so finally Ind will contain only the indexes of the remaining strictly positive c_i's. The input weight γ and the biases $(P_i)_{i \in Ind}$ of the hidden-layer perceptrons are given (i.e. not estimated).

As a measure of fit (or error function), we use the sum-of-squared errors (SSE) defined by

$$R(\Theta) = \sum_{i=1}^{volr} (T_i - y(P_i))^2,$$

where $T_i = \hat{F}_{emp}(P_i)$ and $y(P_i)$ is the output of the neural network for the input P_i, $i = 1, ..., volr$.

The generic approach for minimizing $R(\Theta)$ is by gradient descent (see Fine (1999)), called *back-propagation* in this setting. Because of the compositional form of the model, the gradient can be easily derived using the chain rule for differentiation (see Fine (1999)). This can be computed by a forward and backward sweep over the network, keeping track only of quantities local to each perceptron.

There are several training algorithms that implement the back-propagation approach. In the present study we used the Levenberg-Marquardt training algorithm (`trainlm`) reported to have the fastest convergence (see Demuth & Beale (2000)). This advantage is especially noticeable if very accurate training is required. In many cases `trainlm` is able to obtain lower SSE than any of the other algorithms tested.

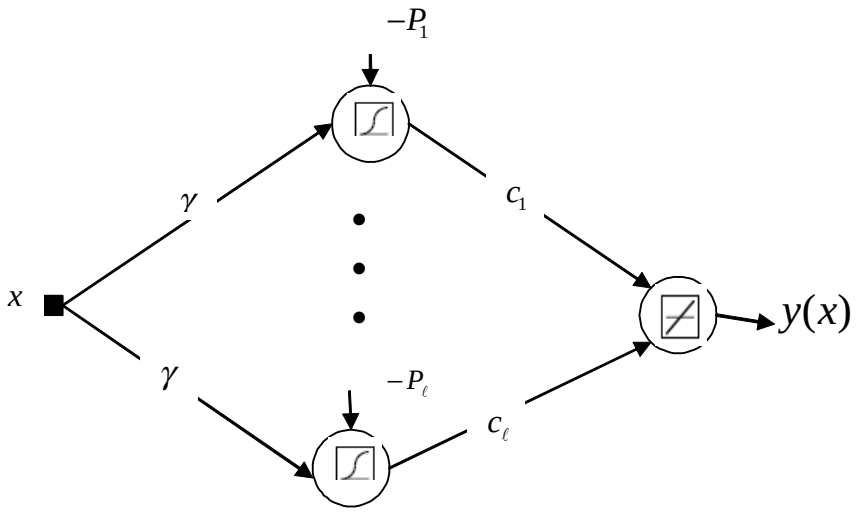


Figure 1: The architecture of the MLP network

The training set $(P_i, T_i)_{i=1,...,volr}$ is divided into training, validation and test subsets (one fourth of the data for the validation set, one fourth for the test set and the rest for the training set). Here $volr$ is the sample size of the training set chosen to be smaller than the sample size vol to prevent an "out of memory" message.

After each training, the positiveness of the c_i 's is checked; the non-positive layer-weights are eliminated together with the corresponding perceptrons, while the remaining perceptrons are retrained. The process stops when all the c_i's are positive.

We will also denote by $iter$ the number of iterations needed until all c_i's are positive, by mem_red the factor used to split the Jacobian matrix (necessary in the Levenberg-Marquardt algorithm) in submatrices in order to avoid an "out-of-memory" message, and by t_mlp the time in seconds needed to compute $1-\hat{F}_{MLP}$ on a laptop with a 1.4GHz Intel Mobile Processor and 1Gb RAM which run Matlab R14 under Windows XP Professional System.

For simulation, we have taken the dimension $n = 10$, the mean vector $\boldsymbol{\mu}_{10}^T = (0.1, 0.1, ..., 0.1)$ and the covariance matrix

$$\boldsymbol{\Sigma}_{10} = \begin{pmatrix} 0.05 & 0.01 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.01 & 0.1 & 0.01 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.01 & 0.01 & 0.1 & 0.01 & 0.02 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.02 & 0.01 & 0.05 & 0.05 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.02 & 0.05 & 0.1 & 0.01 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0.01 & 0.1 & 0.02 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0.02 & 0.05 & 0.01 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0.01 & 0.02 & 0.01 & 0.01 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0.01 & 0.1 & 0.05 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.01 & 0.05 & 0.05 \end{pmatrix}.$$

For α we considered the value 1.5 as is reasonable, for example, in fire insurance (see Beirlant *et al.* (1996)).

We have generated $vol = 1,000,000$ i.i.d. values of S, initializing the random generator with $state = 123456789$. Some of the descriptive statistics of the generated sample are:

$$\begin{aligned} minS &= 1.754; \quad maxS = 76,090.647; \quad \bar{S} = 48.55; \quad \bar{\sigma} = 150.363; \\ skw &= 191.4356 \text{ (the skewness factor);} \\ \text{IQR} &= 33.65 \text{ (the interquartile range, i.e. [the third quartile - the first quartile])} \end{aligned}$$

Also, the values of the parameters involved in the MLP model are

$$volr = 4,000 \text{ (i.e. } 0.4\% \text{ from } vol); \ \gamma = 1; \ mem\_red = 16;$$
$$\textit{final number of elements of } Ind = 411 \text{ (i.e. } 10.275\% \text{ from } volr).$$

The values on the performance function on the training, validation and test subsets are $SSE.train = 1.1124e-006$, $SSE.val = 0.0002476$, $SSE.test = 0.0022865$, respectively.

Some numerical results are given in Table 1. Also, we finally obtained

$$iter = 12; \ \sum_i c_i = 1.00244; \ t\_mlp = 840.88\text{s} \ (\approx 14 \text{ min.}) .$$

Table 1. Simulated versus asymptotic and MLP values of the tail probability for Pareto claims with lognormal discount factors

x	$1-\hat{F}_{emp}$	$1-F_{asym}$	$1-\hat{F}_{MLP}$	$1-\frac{1-F_{asym}}{1-\hat{F}_{emp}}$	$1-\frac{1-\hat{F}_{emp}}{1-\hat{F}_{MLP}}$	$1-\frac{1-F_{asym}}{1-\hat{F}_{MLP}}$
100	0.082271	0.02631	0.073855	0.68020	0.00909	0.64376
200	0.020225	0.00930	0.019722	0.54017	0.00051	0.52844
300	0.008982	0.00506	0.009152	0.43665	-0.00017	0.44714
400	0.005164	0.00329	0.005249	0.36290	-8.5874e-005	0.37326
500	0.003353	0.00235	0.003271	0.29914	8.2044e-005	0.28161
600	0.002396	0.00179	0.002378	0.25292	1.7520e-005	0.24743
700	0.001826	0.00142	0.001843	0.22234	-1.7056e-005	0.22953
800	0.001465	0.00116	0.001474	0.20819	-9.8265e-006	0.21346
900	0.001180	0.00097	0.001169	0.17797	1.1354e-005	0.16999
1000	0.000991	0.00083	0.000996	0.16246	-4.5812e-006	0.16631
1500	0.000508	0.00045	0.000510	0.11417	-2.2300e-006	0.11804
2000	0.000315	0.00029	0.000314	0.07937	1.1599e-006	0.07596
2500	0.000236	0.00021	0.000235	0.11017	6.7655e-007	0.10761
3000	0.000185	0.00016	0.000184	0.13514	1.0156e-006	0.13036
3500	0.000150	0.00013	0.000150	0.13333	2.5790e-007	0.13184
4000	0.000123	0.00010	0.000123	0.18699	4.5989e-007	0.18394

Figure 2 represents, comparatively, the empirical and the MLP estimated cdf of S (black '*' represent $\hat{F}_{emp}$, gray '+' represent $\hat{F}_{MLP}$); A) and B) represent the same figure zoomed in with different coefficients.

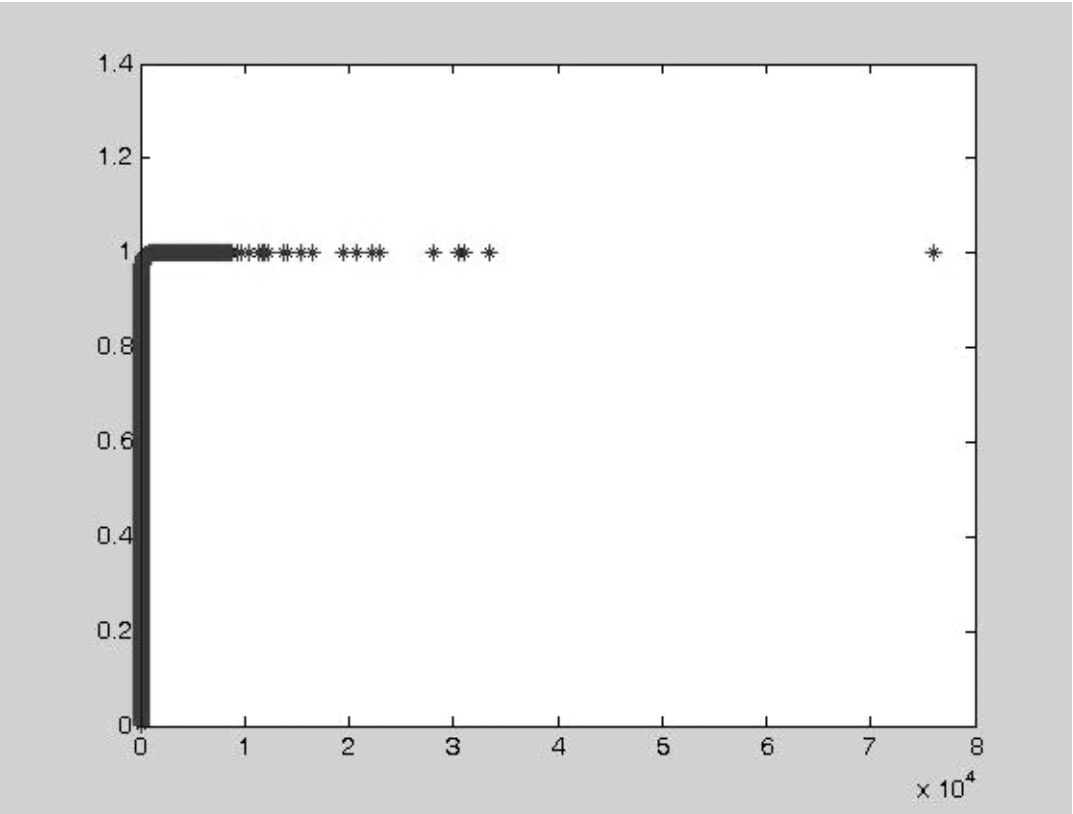
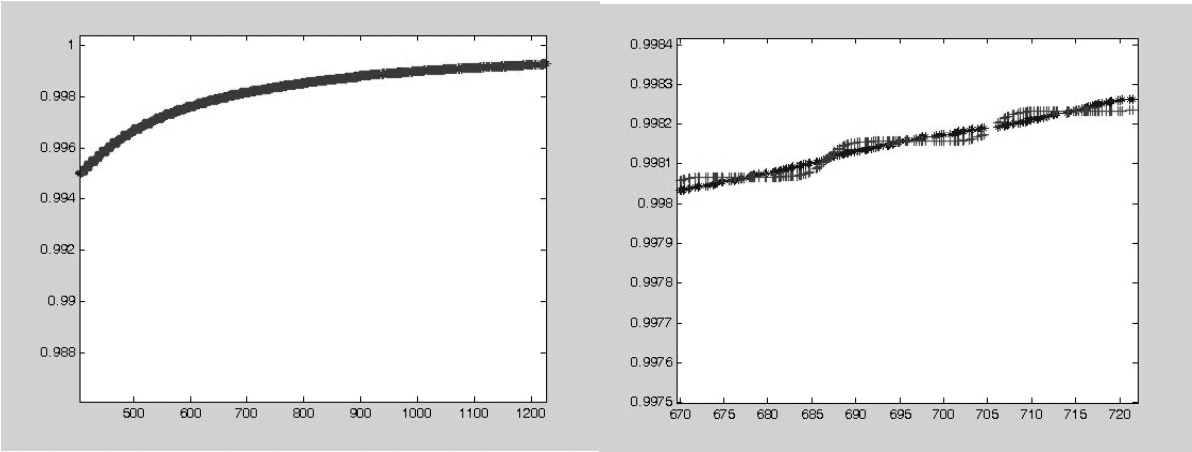


Figure 2: Empirical against MLP-estimated cdf of S



A) **B)**

3 The Kernel model

Details on this model can be found in Martinez & Martinez (2002). Such a model assumes that the probability density function (pdf) of S can be approximated as

$$\hat{f}_S(x) = \frac{1}{vol \times h} \sum_{i=1}^{vol} K\left(\frac{x - s_i}{h}\right),$$

where h is the window width given by the Freedman-Diaconis formula

$$h = 0.786 \times \text{IQR} \times vol^{-1/5}.$$

Here IQR is the interquartile range introduced in section 2.

The function K, also called *kernel,* can be any legitimate probability density function.

For our simulation, we have chosen both the Epanechnikov kernel

$$K(t) = \frac{3}{4}\left(1 - t^2\right),\ -1 \leq t \leq 1,$$

and Gauss kernel

$$K(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right),\ -\infty < t < \infty.$$

Hence, we estimated the cdf of S by

$$\hat{F}_{Epan}(x) = \int_{-\infty}^{x} \hat{f}_{Epan}(s)\, ds$$

and then by

$$\hat{F}_{Gauss}(x) = \int_{-\infty}^{x} \hat{f}_{Gauss}(s)\, ds,$$

respectively.

For the same set of simulated data, taking $h = 1.668$, the results are given in Table 2.

Table 2. Simulated versus Epanechnikov and Gauss values of the tail probability for Pareto claims with lognormal discount factors

x	$1-\hat{F}_{emp}$	$1-\hat{F}_{Gauss}$	$1-\hat{F}_{Epan}$	$1-\frac{1-\hat{F}_{emp}}{1-\hat{F}_{Gauss}}$	$1-\frac{1-\hat{F}_{emp}}{1-\hat{F}_{Epan}}$	$1-\frac{1-\hat{F}_{emp}}{1-\hat{F}_{MLP}}$
100	0.082271	0.082316	0.082469	0.00054	0.00240	0.00909
200	0.020225	0.020223	0.020242	-0.00010	0.00083	0.00051
300	0.008982	0.008985	0.008989	0.00039	0.00076	-0.00017
400	0.005164	0.005164	0.005164	4.5653e-005	-5.9582e-005	-8.5874e-005
500	0.003353	0.003349	0.003348	-0.00118	-0.00163	8.2044e-005
600	0.002396	0.002397	0.002398	0.00041	0.00072	1.7520e-005
700	0.001826	0.001827	0.001827	0.00062	0.00062	-1.7056e-005
800	0.001465	0.001465	0.001465	1.3832e-005	-0.00016	-9.8265e-006
900	0.001180	0.001183	0.001182	0.00234	0.00170	1.1354e-005
1000	0.000991	0.000992	0.000992	0.00053	0.00136	-4.5812e-006
1500	0.000508	0.000508	0.000507	-0.00079	-0.00142	-2.2300e-006
2000	0.000315	0.000315	0.000315	4.5036e-006	0.00028	1.1599e-006
2500	0.000236	0.000236	0.000236	-1.6281e-006	-0.00018	6.7655e-007
3000	0.000185	0.000185	0.000185	-0.00026	-0.00199	1.0156e-006
3500	0.000150	0.000150	0.000150	-0.00070	-0.00197	2.5790e-007
4000	0.000123	0.000123	0.000123	0.00013	0.00135	4.5989e-007

4 Conclusions

In the previous sections we presented several models of approximative representations for the cdf of S. Based on the numeric comparisons, we can conclude that in what concerns the tail probability under investigation, the MLP approximation is the best since:

- From Table 1 we see that the MLP model approximates the tail probabilities of S better than the asymptotic formula.

- From Table 2 we see that the quality of the Epanechnikov and Gauss models is similar. Also, both are better that the asymptotic approximation, while the MLP model is more accurate than both kernel methods.

- Using the MLP allows the "learning" of S while the data accumulate.

- Related to the MLP method, it is interesting to notice that finally $Ind \neq \emptyset$, while $\sum_i c_i \simeq 1$. Hence, it seems that the cdf of S can be represented as a mixture of logistic functions.

- It is also remarkable that the MLP obtains similar results with $\hat{F}_{emp}$ based on only 4,000 observed/generated data (i.e. 0.4% from the sample size), and even this number is reduced to 400 (i.e. 0.04% from the sample size). In other words, MLP realizes an impressing compression of information by obtaining the same information as from 1,000,000 data, but using only 400 of them.

- This method can be used to estimate the cdf of (1) in a very general setting, while e.g. the asymptotic formula of Goovaerts et al. (2004) is valid only when F_X is regularly varying, hence heavy-tailed (for more details on "regularly varying" see e.g. Bingham *et al.* (1987)).

References

[1] Beirlant, J.; Teugels J.L.; Vynckier, P. (1996). *Practical Analysis of Extreme Values*. Leuven: Leuven University Press.

[2] Bingham, N.H.; Goldie C.M.; Teugels, J.L. (1987). *Regular Variation*. Cambridge: Cambridge University Press.

[3] Bolancé, C.; Guillen, M.; Nielsen, J.P. (2003). Kernel density estimation of actuarial loss functions. Insurance: Mathematics and Economics 32, 19-36.

[4] Brockett, P.; Cooper, W.; Golden, L; Pitaktong, U. (1994). A neural network method for obtaining an early warning of insurer insolvency. The Journal of Risk and Insurance 61, no.3, 402-424.

[5] Brockett, P.; Xia, X.; Derrig, R. (1998). Using Kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. The Journal of Risk and Insurance 65, no.2, 245-274.

[6] Demuth, H.; Beale, M. (2000). *Neural Network Toolbox. User's Guide*, The Mathworks Inc., Natick, USA.

[7] Dhaene, J.; Denuit, M.; Goovaerts, M.J.; Kaas, R.; Vyncke, D. (2002a). The concept of comonotonicity in actuarial science and finance: theory. Insurance: Mathematics and Economics 31, 3-33.

[8] Dhaene, J.; Denuit, M.; Goovaerts, M.J.; Kaas, R.; Vyncke, D. (2002b). The concept of comonotonicity in actuarial science and finance: applications. Insurance: Mathematics and Economics 31, 133-161.

[9] Fine, L.T. (1999). *Feedforward Neural Network Methodology*, Springer NY, Heidelberg.

[10] Gavin, J.; Haberman, S.; Verrall, R. (1993). Moving weighted average graduation using kernel methods. Insurance: Mathematics and Economics 12, 113-126.

[11] Gavin, J.; Haberman, S.; Verrall, R. (1994). On the choice of bandwidth for kernel graduation. Journal of the Institute of Actuaries, 121, 119-134.

[12] Goovaerts, M. J.; Kaas, R.; Laeven, R. J. A.; Tang, Q.; Vernic, R. The tail probability of discounted sums of Pareto-like losses in insurance. Scandinavian Actuarial Journal (2004), to appear.

[13] Hassoun, M.H. (1995). *Fundamentals of Artificial Neural Networks*. The MIT Press, Cambridge, Massachusetts.

[14] Hastie, T.; Tibshirani, R.; Friedman, J. (2002). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, NY, Heidelberg.

[15] Hertz, J.; Krogh, A.; Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation.* Addison-Wesley, Reading, Massachusetts.

[16] Klugman, S.A.; Panjer, H.H.; Willmot, G.E. (1998). *Loss Models: from Data to Decisions.* Wiley, New York.

[17] Martinez, L.W.; Martinez, R.A. (2002) *Computational Statistics Handbook with Matlab.* Chapman & Hall/CRC.

[18] Nielsen, J.P.; Tanggaard, C.; Jones, M.C. (2004). Local linear density estimation for filtered survival data, with bias correction. Centre for Analytical Finance working paper no. 185, Aarhus School of Business, Denmark.

[19] Nyrhinen, H. (1999). On the ruin probabilities in a general economic environment. *Stochastic Processes and their Applications* 83, no. 2, 319-330.

[20] Ripley, B. (1996). *Pattern Recognition and Neural Networks.* Cambridge: Cambridge University Press.

[21] Shapiro, A. (2000). A Hitchhiker's guide to the techniques of adaptive nonlinear models. Insurance: Mathematics and Economics 26, 119-132.

[22] Shapiro, A.; Gorman, R.P. (2000). Implementing adaptive nonlinear models. Insurance: Mathematics and Economics 26, 289-307.

[23] Tang, Q.; Tsitsiashvili, G. (2003). "Precise estimates for the ruin probability in finite horizon in a discrete-time model with heavy-tailed insurance and financial risks," *Stochastic Processes and their Applications* 108, no. 2, 299-325.

[24] Tang, Q.; Tsitsiashvili, G. (2004). "Finite and infinite time ruin probabilities in the presence of stochastic return on investments," *Advances in Applied Probability* 36, no. 4, 1278-1299.