CORRELATION ANALYSIS BASED ON NEURAL NETWORK COPULA FUNCTION

by

Haibin LI[a], Lijun SUN[a], and Qinghu YAO[b*]

[a] Science College, Inner Mongolia University of Technology, Hohhot, China
[b] School of Materials Engineering, Inner Mongolia University of Technology, Hohhot, China

Original scientific paper
https://doi.org/10.2298/TSCI2303081L

The joint-distribution function between variables plays an important role in reliability analysis. A method is proposed for constructing the function using a neural network, which is used to construct a copula model under arbitrarily measured data, including the input and output values of the neural network using an empirical cumulative distribution. Three traditional copula function models are constructed based on the Kendall rank-correlation coefficients. Based on the Euclidean distance method, the neural network copula and three copula function models are compared.

Key words: *applied mechanics, correlation, neural network copula*

Introduction

Within structural reliability analysis, there are generally multiple types of related variables. For examples, the standardised foundation pile payloads [1], and permanent displacement of a structure during earthquake [2], and weld fatigue [3]. The distribution of these parameters is often non-normal, and when computing the structural uncertainty, the joint distribution function of each variable must be known, whereas it requires a large amount of experimental data to obtain the joint distribution function of a given set of variables, which is difficult to realise in practical engineering. For this reason, the correlation between variables is often not considered when computing the structural reliability in an engineering project, which, has, however, caused inaccurate results. Therefore, it is extremely important to analyse the structural uncertainty in a more reasonable way by considering the correlation between variables. The key point is how to create a joint distribution function with a limited amount of sample information.

In recent years, the copula function has provided a new way to establish a joint distribution function between variables [4]. The copula function was first proposed by Sklar [5] in 1959. Sklar [5] proposed the idea that any multivariate joint distribution function can be divided into a corresponding marginal distribution as well as a copula function. This copula function determines the correlation between the variables, including the size of the correlation factor between variables and the types of correlation structures. The copula formula has been successfully applied to finance, hydrology, geotechnical engineering, and mechanics [6-8]. Currently, however, there are a limited number of categories of copula functions, each of

* Corresponding author, e-mail: feilong_yqh@163.com

which has only one or two adjustable parameters, making it difficult to apply to the diverse demands of arbitrary correlations.

Artificial neural networks can learn to map input and output variables, and can approach any functions. Much literature has indicated that a three-layer back propagation neural network can fit any non-linear problems with limited training samples. This satisfies the conditions for constructing any correlation copula functions. To this end, in this study, copula functions for multiple variables with a given sample data set are established.

Theoretical foundation of copula function

A copula function is a type of joint distribution functions composed of a collection of variable marginal distributions. Under a situation with n dimensions, the copula function can be defined as an n-dimensional joint distribution function in the space $[0, 1]^n$, in which the marginal distribution of each variable is uniformly distributed within the interval [0, 1] [4]:

$$F(x_1, x_2, \cdots, x_n) = C\left[F_1(x_1), F_2(x_2), \cdots, F_n(x_n); \theta\right] = C(u_1, u_2, \cdots, u_n; \theta) \tag{1}$$

Based on this definition, $F(x_1, x_2, \cdots, x_n)$ is the joint distribution function of the variables $x_1, x_2, \cdots, x_n$; $u_i = F_i(x_i)(i = 1, 2, \cdots n)$ is the marginal distribution function of the variable x_i; $C(\bullet)$ is a copula function; and θ is the correlation parameter of the copula function.

Deriving the equation in function (1) yields the joint probability density function:

$$\begin{aligned} f(x_1, x_2, \cdots, x_n) &= f(x_1) f(x_2) \cdots f(x_n) c[F_1(x_1), F_2(x_2), \cdots, F_n(x_n); \theta)] = \\ &= c(u_1, u_2, \cdots, u_n; \theta) \prod_{i=1}^{n} f_i(x_i) \end{aligned} \tag{2}$$

Within this function, $f_i(x_i)$ is the marginal probability density function of the variables $x_1, x_2, \cdots, x_n$; and $c(\bullet)$ is the density function of the copula function. Therefore, if one already knows the marginal distribution and copula function of the variables, one can construct the variable joint distribution function and joint probability density function through eqs. (1) and (2).

Estimating the copula function's correlation parameter $C(u_1, u_2, \cdots, u_n; \theta)$ is an important stage of establishing the copula function. The correlation parameter, θ, indicates the degree of correlation between the variables and is generally obtained from the Kendall rank correlation coefficient [4]. This type of estimation methods for parameters related to the copula function is unrelated to the marginal distribution of the variables and is thus referred to as a nonparametric method.

For the 2-D variables x_1 and x_2, the relationship between the Kendall rank correlation coefficient τ and the copula function correlation coefficient, θ, is [4]:

$$\tau = 4 \int_0^1 \int_0^1 C(u_1, u_2; \theta) \mathrm{d}C(u_1, u_2; \theta) - 1 \tag{3}$$

Different categories of copula functions describe the variable correlation in different ways, namely, they have different structures of correlation. Table 1 shows three such distinct types of copula functions. For this reason, quantitatively evaluating which copula function can be considered as the most optimal one is a central problem. There are several methods used to discern the optimal copula function, such as the Euclidean distance method, Akaike information criterion, and Bayesian information criterion. In this study, the Euclidean distance method is used to discern the optimal copula function, in which the copula function with the

lowest Euclidean distance value has the most optimal fitting variable correlation structure. The Euclidean distance method for a distribution model with n dimensions can be expressed:

$$e = \sqrt{\frac{1}{p}\sum_{j=1}^{p}[F_j(x_1, x_2, \cdots, x_n) - C_j(u_1, u_2, \cdots, u_n; \theta)]^2} \tag{4}$$

Within this function, p is the number of test sample points.

Table 1. Copula function types

Copula type	Copula distribution function $C(u_1, u_2; \theta)$	The value range of θ
Plackett	$\frac{S - \sqrt{S^2 - 4u_1u_2\theta(\theta-1)}}{2(\theta-1)}, \quad S = 1 + (\theta-1)(u_1 + u_2)$	$\theta \in (0, \infty) \backslash \{1\}$
Frank	$-\frac{1}{\theta}\ln\left[1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1}\right]$	$(-\infty, \infty) \cap (\theta \neq 0)$
Gaussian	$\phi_2[\phi^{-1}(u_1), \phi^{-1}(u_2); \theta]$	[–1, 1]

Method for establishing a copula function model based on neural networks

A correlation between variables within engineering problems is often extremely complicated. A traditional copula function with a small number of parameters is difficult to apply in accurately constructing a correlation model. The copula function is a joint distribution function that takes the marginal distributions of various correlated variables as its own variables. It maps the relationship between the marginal and joint distributions. If neural networks can map any non-linear functions, they have the potential to become an accurate method for the modelling of copula functions. To this end, this neural network methodology is used in the present study to establish a copula correlation model for any set of measured data.

Determining structure of neural network

The first step in establishing a methodology for modelling the copula functions based on a neural network is to determine the network structure. Suppose we are researching the correlation between random variables in n dimensions, based on the definition of the copula function, the input of the problem is the marginal distribution of the n-dimensional random variables, and its output is the joint distribution of those random variables. Based on this, the neural network will have n input layer units and 1 output layer unit. Based on the principle of structural risk minimisation, one should select as few network parameters as possible. Therefore, a single hidden layer network structure with the least number of units necessary to allow network training convergence is chosen. A diagram of the network structure can be seen below in fig. 1, and the functional relationship between the network input and output is.

$$y = \sum_{j=1}^{m} w_j f\left(\sum_{i=1}^{n} k_{ji} u_i + b_j\right) \tag{5}$$

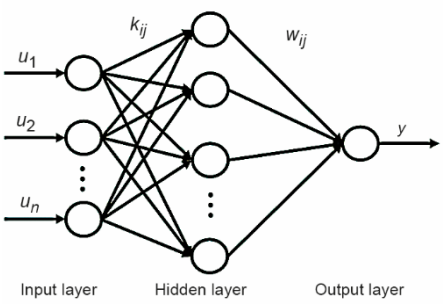

Figure 1. Three-layer back propagation network structure

where u_i is the i^{th} unit input variable of the neural network input layer, y – the unit variable of the network output layer, k_{ji} – the linked weighted value between the i^{th} unit of the input layer and the j^{th} unit of the hidden layer, b_j – the threshold value of the j^{th} unit of the hidden layer, and w_j – the linked weighted value between the j^{th} unit of the hidden layer and the unit of the output layer.

Constructing neural network trained sample set

When conducting a correlation study, one must have a measured data sample set that reflects the correlation between variables. In the case of engineering problems, a sample of the measured data points that contain the correlation information is obtained by collecting various types of experimental data. According to the definition of the copula function, the neural network input should be the marginal cumulative distributions of each random variable, and its output should be the joint cumulative distribution of the random input variables. Understanding how to use available measured sample data to construct a trained sample set is a key step for using a neural network to establish copula function models. This step requires resolving the question of how to change an observed sample set into a trained sample set. This paper calculates the marginal empirical cumulative distribution and joint empirical cumulative distribution of random variables from various actual sample points, with the empirical cumulative distribution function as the foundation for such calculations. This is the process by which the neural network trained sample set is obtained.

Suppose there are random variables $(x_1^i, x_2^i, \cdots, x_n^i)$ in n dimensions. Among them, $i = 1, 2, \cdots, N$ are the numbers of measured sample points. Take each dimension in an n-dimensional space and divide it into N number subspaces based on the position of each sample point in the given dimension. The process by which the aforementioned observed sample values are utilised to determine the marginal empirical cumulative distribution and joint empirical cumulative distribution of each random variable at the sample points is as follows.

According to the definition of an empirical cumulative distribution, the i^{th} random variable's marginal empirical cumulative distribution at the position of the j^{th} sample point x_i^j is:

$$F_i(x_i^j) = \frac{1}{N}\sum_{k=1}^{m} \mathrm{I}(x_i^j - x_i^k) \qquad (i = 1, 2, \cdots, n)$$

The joint empirical cumulative distribution of each random variable at the position of the j^{th} sample point $(x_1^i, x_2^i, \cdots, x_n^i)$ is:

$$F(x_1^j, x_2^j, \cdots, x_n^j) = \frac{1}{N}\sum_{k=1}^{N}\prod_{i=1}^{n} I(x_i^j - x_i^k)$$

where $I(x)$ is the characteristic function, and is represented:

$$I(x) = \begin{cases} x < 0, 1 \\ x \geq 0, 0 \end{cases} \tag{6}$$

From this, the following can be obtained: The corresponding neural network training sample input for the j^{th} sample point $(x_1^j, x_2^j, \cdots, x_n^j)$ is $\{F_1(x_1^j), F_2(x_2^j), \cdots, F_n(x_n^j)\}$, with an output of $F(x_1^j, x_2^j, \cdots, x_n^j)\ \ j = 1, 2, \cdots, N$. In total, there are N trained samples.

Furthermore, according to the boundary conditions of the copula function $C_{NN}(u_1, \cdots, u_{i-1}, 0, u_{i+1}, \cdots, u_n) = 0$, the constructed input is $\{u_1, \cdots, u_{i-1}, 0, u_{i+1}, \cdots, u_n\}$, with the output being the trained sample of $\{0\}$.

Neural network training

For any multi-dimensional question where $n \geq 2$, after obtaining the alternative dataset, a certain proportion of the data is randomly selected from the set to serve as a training sample set and allow the remaining data to serve as a test sample set. At this point, the training sample set can be utilised to train the neural network. Currently, the training algorithm is well developed. Further details can be found in a previous study [9].

After the network has converged, a set of fixed network parameters w_j^*, k_{ji}^*, b_j^* is obtained. At this point, the neural network is already a mathematical model that can fit the relationship between the input and output of the training sample. A more detailed expression of this model can be written:

$$y = \sum_{j=1}^{m} w_j^* f\left[\sum_{i=1}^{n} k_{ji}^* F_i(x_i) + b_j^*\right] \tag{7}$$

Density function of neural network

As can be understood from the definition of the copula function, the neural network model shown in eq. (6) is an approximation of the joint cumulative distribution function of the n-dimensional random variables $(x_1, x_2, \cdots, x_n)$ rom the measured sample data copula function, as shown in eq. (1). This is demonstrated in:

$$y = F(x_1, x_2, \cdots, x_n) = C_{NN}[F_1(x_1), F_2(x_2), \cdots, F_n(x_n)] \tag{8}$$

Based on the same reasoning, the sample point set $[x_i^k, F_1(x_i^k)]\ \ (k = 1, 2, \cdots, N)$ can be used to fit the marginal cumulative distribution of each variable x_i.

According to copula-related theory, the joint probability density function is:

$$f(x_1, x_2, \cdots, x_n) = c_{NN}[F_1(x_1), F_2(x_2), \cdots, F_n(x_n)] \prod_{i=1}^{n} f_i(x_i) \tag{9}$$

where

$$c_{NN}(x_1, x_2, \cdots, x_n) = \frac{\partial C_{NN}(x_1, x_2, \cdots, x_n)}{\partial x_1 \partial x_2 \cdots \partial x_n}, \quad f_i(x_i) = \frac{\partial F_i(x_i)}{\partial x_i}$$

$$c_{NN}(x_1, x_2) = \sum_{j=1}^{m} w_j \prod_{i=1}^{n} {}^* k_{ji} f^{(n)}\left(\sum_{i=1}^{n} k_{ji}^* x_i + b_j^*\right) \tag{10}$$

Assume that:

$$W_j^* = w_j \prod_{i=1}^{n} {}^* k_{ji}$$

$$f(x_1, x_2, \cdots, x_n) = \sum_{j=1}^{m}\left[W_j^* f^{(n)}\left(\sum_{i=1}^{n} k_{ji}^* x_i + b_j^*\right)\right] \prod_{i=1}^{n} f_i(x_i) \tag{11}$$

Equation (9) is the joint probability density function obtained from the neural network methodology.

The total area integral of the above joint probability density function is taken [10], at a value of L, and the joint probability density function obtained after normalisation is:

$$f(x_1, x_2, \cdots, x_n) = \frac{1}{L}\sum_{j=1}^{m}\left[W_j^{*} f^{(n)}\left(\sum_{i=1}^{n} k_{ji}^{*} x_i + b_j^{*} \right)\right]\prod_{i=1}^{n} f_i(x_i) \tag{12}$$

Examples of numerical simulation

Suppose the random variables (X, Y) have the following joint probability density function:

$$f(x,y) = \alpha_1\alpha_2 e^{-(e^{-sx^*}+e^{-sy^*})^{\frac{1}{s}} - s(x^*+y^*)}; \quad x^* = \alpha_1(x-\lambda_1); \quad y^* = \alpha_2(y-\lambda_2) \tag{13}$$

Within this function, $\alpha_1 = \alpha_2 = 1$; $\lambda_1 = \lambda_2 = 0$; $s = 2$. Based on the apparent correlation between variables, in this example calculation, the Plackett copula, Frank copula, and Gaussian copula functions are chosen to construct the 2-D joint distribution functions for the variables X and Y.

To confirm that the method proposed by this paper can be applied to a small sample size, $N = 20$ and $N = 50$ were selected when fitting the model. Because the Kendall rank correlation coefficient considers not only the linear correlation between variables, but also the non-linear correlation between variables, it was used to obtain the value of the correlation parameter, θ, for each copula function, which can be seen in tab. 2 [11].

Table 2. Value of correlation parameter, θ, in example

Copula function		Plackett copula	Frank copula	Gaussian copula
θ value	$N = 20$	12.5018	6.0309	0.7244
	$N = 50$	10.1082	5.3635	0.6832

The use of the copula parameters shown in tab. 2, as well as the marginal distributions of random variables X and Y, yields three types of copula functions. Therefore, we will take $N = 20$ as an example for detailed analysis, and the analysis process of $N = 50$ will not be described. The Euclidean distance calculation results of each of copula function described above are shown in tab. 3.

After this, the neural network methodology proposed herein was utilised to carry out an analysis. A neural network Copula function with two input layer units and one output layer unit was constructed. This paper utilised a neural network structure with a single hidden layer. The unit number in the hidden layer was set to $m = 8$ based on the principle of parameter minimisation. Because, *logsig* function can be written:

$$\text{logsig}(t) = \frac{1}{1+\exp(t)} \tag{14}$$

where t input layer variable of the neural network. So, the *logsig* function was selected as the transfer function for both the hidden and output layers in this article.

Applying the Levenberg-Marquardt training algorithm [12], the number of training epochs was set to 1500 epochs. When the training proceeded to the 1133th epoch, the default precision had already met the required level and the training stopped. The mean squared error arriving at a level of $2.464 \cdot 10^{-4}$. Figure 2 shows a scatterplot of the empirical cumulative distribution of the trained sample points $(x_1{}^j, x_2{}^j)$ and the empirical cumulative distribution of the copula neural network. It can be seen from fig. 2 that the empirical cumulative distribution of the simulated data encapsulates the empirical cumulative distribution of the original data relatively well. Therefore, the data obtained from the joint empirical cumulative distribution function simulation produced by the neural network copula function fits the original data in terms of both the shape and trend.

The test sample set was substituted into a well-established neural network copula model. Its residual error after testing is as shown in fig. 3. The Euclidean distance is used simultaneously for identification, which can be seen in tab. 4.

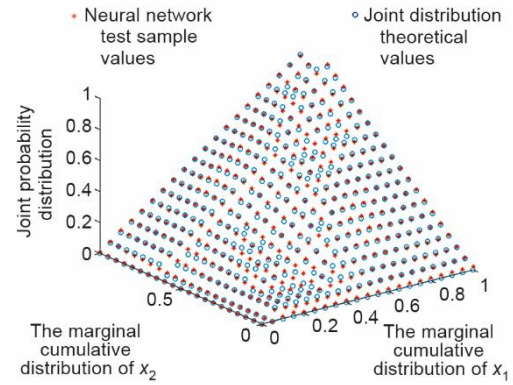

Figure 2. Copula neural network and empirical cumulative distribution of original samples

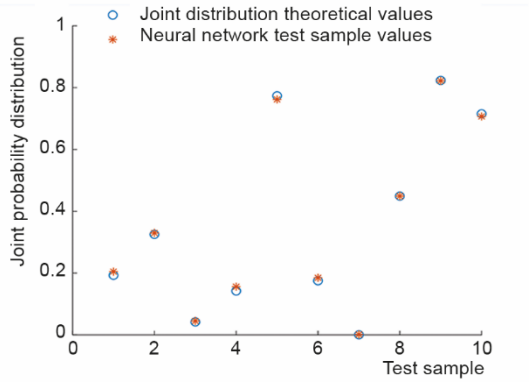

Figure 3. Results of test sample in example

Figure 3 clearly shows that the training results of the neural network copula basically fit the actual results. The errors of the test sample after copula training of the neural network were mostly quite small, although there were some sample points that had comparatively large errors. There were primarily two factors causing large errors. The first reason why the errors were large was because the training results had a relatively small value. As the second reason, because the training of the marginal points of the neural network was comparatively poor, and because the selection of the test sample was random, a portion of the points selected were marginal points that increased the error.

Table 3. The Euclidean distance discrimination results of example

Sample size N	20	50
Plackett copula	0.0278	0.0191
Frank copula	0.0230	0.0207
Gaussian copula	0.0247	0.0154
Copula neural network	0.0222	0.0177

As shown in tab. 3, When $N = 20$ that the results of the Euclidean distance differentiation for the neural network copula function were much more optimal than those of the other three copula functions. When $N = 50$ that the results of the Euclidean distance differentiation for the neural network copula function were not much different from the other three copula functions. This indicates that the neural network copula model can effectively represent a 2-D distribution model with a correlation. When utilising this methodology to establish a 2-D joint distribution model, there is no need to estimate the type of distribution to which each variable is subordinated. In this way, the neural network copula model provides a new methodology for establishing a 2-D joint distribution model.

Conclusion

When conducting a structural reliability analysis, one must already be generally aware of the correlation information between variables, such as the joint density function or joint distribution function of the variables. The neural network copula method applied in this study produces a joint distribution function of the correlated variables. Through a comparison of this method with three traditional copula functions in terms of the Euclidean distance, it is clear that the proposed method has an obvious advantage in terms of precision. Compared with traditional algorithms, this methodology does not require selecting an optimal copula model from the numerous existing copula functions, and thus greatly reduces the required workload. The present technology can be extended to other neural network [13-16].

Acknowledgment

This research is funded by National Natural Science Foundation of China (Grant No. 11962021) and Natural Science Foundation of Inner Mongolia (Grant No. 2021MS05020).

References

[1] Phoon, K. K., Kulhawy, F. H., Characterisation of Model Uncertainties for Laterally Loaded Rigid Drilled shafts, *Geotechnique*, *55* (2005), 1, pp. 45-54
[2] Goda, K., Statistical Modeling of Joint Probability Distribution Using Copula: Application to Peak and Permanent Displacement Seismic Demands, *Structural Safety*, *32* (2010), 2, pp. 112-123
[3] Leira, B. J., Probabilistic Assessment of Weld Fatigue Damage for a Non-Linear Combination of Correlated Stress Components, *Probabilistic Engineering Mechanics*, *26* (2011), 3, pp. 492-500
[4] Nelsen, R. B., *An Introduction to Copulas*, Springer, New York, USA, 2006
[5] Sklar, A., Functions de Repartition an Dimensions Etleurs Max'ges, *Publications de l'Institut de Statistique de l'Universit6 de Paris*, *8* (1959), pp. 229-231
[6] Kjersti, A., *et al.*, Pair-Copula Constructions of Multiple Dependence, *Insurance Mathematics & Economics*, *44* (2009), 2, pp. 182-198
[7] Subimal, G., Modelling Bivariate Rainfall Distribution and Generating Bivariate Correlated Rainfall Data in Neighbouring Meteorological Subdivisions Using Copula, *Hydrological Processes*, *24* (2010), 24, pp. 3558-3567
[8] Tang, X. S., *et al.*, Bivariate Distribution Models Using Copulas for Reliability Analysis, *Proceedings of the Institution of Mechanical Engineers*, *227* (2013), 5, pp. 499-512
[9] Liu, J. W., *et al.*, Research and Development on Deep Learning (in Chinese), *Application Research of Computers 31* (2014), 7, pp. 1921-1930
[10] Li, H. B., *et al.*, Structural Reliability Calculation Method Based on the Dual Neural Network and Direct Integration Method, *Neural Computing & Applications*, *29* (2018), 7, pp. 425-433
[11] Fan, R., *et al.*, Survey of Research on Statistical Correlation Analysis (in Chinese), *Mathematical Modeling and Its Applications*, *3* (2014), 1, pp. 1-12
[12] Daliakopoulos, I. N., *et al.*, Groundwater Level Forecasting Using Artificial Neural Networks, *Journal of Hydrology*, *309* (2005), 1, pp. 229-240

[13] Yu, W., *et al.*, Tensorizing GAN with High-Order Pooling for Alzheimer's Disease Assessment, *IEEE Transactions on Neural Networks and Learning Systems, 33* (2021), 9, pp. 4945-4959
[14] You, S., *et al.*, Fine Perceptive Gans for Brain MR Image Super-Resolution in Wavelet Domain, *IEEE Transactions on Neural Networks and Learning Systems*, On-line first, https://doi.org/10.1109/TNNLS.2022.3153088, 2022
[15] Hu, S., *et al.*, Bidirectional Mapping Generative Adversarial Networks for Brain MR to PET Synthesis, *IEEE Transactions on Medical Imaging, 41* (3021), 1, pp. 145-157
[16] Yu, W., *et al*., Morphological Feature Visualization of Alzheimer's Disease *via* Multidirectional Perception GAN, *IEEE Transactions on Neural Networks and Learning Systems*, On-line first, https://doi.org/10.1109/TNNLS.2021.3118369, 2021

Paper submitted: December 15, 2021
Paper revised: June 10, 2022
Paper accepted: June 13, 2022