Simulating Spatial Correlation for Catastrophic Events

Georg Hofmann
Validus Research*, Waterloo, Ontario, Canada

Abstract— *Catastrophe models calculate the stochastic distributions of loss originating from events like hurricanes and earthquakes. These models are typically based on a stochastic event catalog. For each event spatial correlation needs to be simulated. The standard approach is based on the evaluation of a copula. However the complexity of the corresponding algorithm is $O(n^3)$ and it becomes difficult to execute for n in the thousands. So as the number n of locations in the footprint of the event grows, this approach quickly becomes infeasible. We propose a slight modification of the well-know Kriging technique, in order to solve this problem. With our solution the creation of simulation data for catastrophe models becomes manageable with the use of Big Data techniques.*

Keywords: Simulation, Spatial Correlation, Copula, Catastrophe Model, Kriging

1. Introduction

Catastrophe models are very common tools in the insurance industry. They are used, among other things, to predict distributions for financial loss originating from events like hurricanes and earthquakes. These models are usually based on a stochastic event catalog. For each simulated event in this catalog it is important to quantify uncertainty in the hazard and the vulnerability of structures affected. For this so-called **secondary** uncertainty it is important to understand spatial correlation. Claims data for historic earthquake losses show that there is a relationship between the distance of two locations and their loss correlation.

In this paper we start from the assumption that such a spatial relationship is given and provide a methodology to simulate secondary uncertainty with the prescribed correlation. The traditional approach to this simulation would be to compute a copula directly. However, this approach does not scale well computationally as the number of locations increases. Instead we propose using a copula on a coarser subset of locations and to interpolate for the finer set using a technique similar to the well-known method of Kriging. As this is an approximation of the intended correlation, its accuracy needs to be investigated. We prove results that provide upper and lower bounds for the accuracy of the simulated correlation.

* At the time of the preliminary research, Kai Cui was a member of the Validus Research team. Thanks go to him for adding the Krigging methodology to the mix of interpolation methods initially investigated.

With the proposed approach simulating single events becomes computationally feasible. A code example is provided in the appendix. However, catastrophe models often simulate many events in order to capture the full stochastic nature of a peril. Simulating as many as 1 million events is not uncommon. Implementing our approach in Big Data framework with increased computational resources allows this to be accomplished.

2. A normalized Kriging approach

In this section we define the necessary details for the proposed normalized Kriging approach. The problem at hand can be summarized in the following way: For a set of locations (referred to in the introduction as the coarser set) random variables are defined. They follow a covariance structure. The objective is the following: For a new location define a random variable as a linear combination of the given ones, such that the covariance structure is closely approximated.

The following special case is instructive: If the locations in the coarser set have normally distributed random variables, then so does a linear combination of them. If in addition to that the covariance structure $c(\cdot,\cdot)$ satisfies $c(x^*,x^*) = 1$ for all locations x^*, then all random variables will follow the standard normal distribution. In this case the covariance structure can equivalently be expressed by a correlation structure.

This case is particularly relevant for simulations. The locations in the coarser set can be simulated using a Gaussian copula. The simulation can be extended to any location outside of the coarser set while maintaining a standard normal distribution and approximating the desired correlation. The code presented in the appendix implements this particular simulation.

We use the variable x with subscripts to refer to locations. For any pair of locations (x^*, x^{**}) we denote by $c(x^*, x^{**})$ the target covariance of the two locations. Suppose the n locations $x_1,\ x_2, \ldots,\ x_n$ in the coarser set have the random variables $f(x_1),\ f(x_2), \ldots,\ f(x_n)$. Further, suppose they satisfy

$$\mathrm{Cov}\left(f(x_i), f(x_j)\right) = c(x_i, x_j)$$

for i and $j = 1,\ 2, \ldots,\ n$.

For every location x^* the objective is to define the random variable $\overline{f}(x^*)$ as a linear combination of the

$f(x_1),\ f(x_2),\ldots,\ f(x_n)$ in a way such that

$$\mathrm{Var}(\overline{f}(x^*)) = c(x^*,x^*) \qquad \text{and}$$
$$\mathrm{Cov}(\overline{f}(x^*),\overline{f}(x_i)) \approx c(x^*,x_i)$$

for all i.

Set

$$C = \begin{pmatrix} c(x_1,x_1) & \cdots & c(x_1,x_n) & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ c(x_n,x_1) & \cdots & c(x_n,x_n) & \end{pmatrix}$$

We use the notation

$$f(x) = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \quad \text{and} \quad c(x^*,x) = \begin{pmatrix} c(x^*,x_1) \\ \vdots \\ c(x^*,x_n) \end{pmatrix}.$$

The well-known Kriging estimator $\hat{f}$ is defined in the following way:

$$\hat{f}(x^*) = \big(f(x)\big)^T C^{-1} c(x^*,x)$$

It immediately follows that $\hat{f}(x_i) = f(x_i)$ for each $i = 1,\ldots,n$. It also can be calculated that

$$\mathrm{Cov}(\hat{f}(x^*),\hat{f}(x_i)) = c(x^*,x_i).$$

In other words, Kriging models the prescribed covariance. On the other hand we will prove in this article that

$$\mathrm{Var}(\hat{f}(x^*)) \le c(x^*,x^*)$$

and in general, equality does not hold.

In other words, Kriging does not preserve variance in the same way that it preserves covariance. For the application that we have in mind, the preservation of variance is more important than that of covariance. That is why it is stated as an objective. This leads us to using a normalization: We define the normalized estimator:

$$\overline{f}(x^*) = \sqrt{\frac{c(x^*,x^*)}{\mathrm{Var}(\hat{f}(x^*))}}\,\hat{f}(x^*)$$

This estimator preserves variance:

$$\mathrm{Var}(\overline{f}(x^*)) = c(x^*,x^*).$$

But as a consequence, covariance will be overestimated:

$$\begin{aligned}\mathrm{Cov}(\overline{f}(x^*),\overline{f}(x_i)) &= \frac{c(x^*,x^*)}{\mathrm{Var}(\hat{f}(x^*))}\,\mathrm{Cov}(\hat{f}(x^*),\hat{f}(x_i)) \\ &\ge c(x^*,x_i)\end{aligned}$$

In this article we will derive an upper bound for this overestimation.

3. Upper bound for variance

As promised in the previous section, we prove a proposition about the lower bound of the estimator variance. The proof is necessary for the foundation of the proposed bound. But it is not indispensable for the further understanding of the applications in later sections.

Proposition 3.1:

$$\mathrm{Var}(\hat{f}(x^*)) \le c(x^*,x^*)$$

Proof: The matrix

$$C' = \begin{pmatrix} c(x_1,x_1) & \cdots & c(x_1,x_n) & c(x_1,x^*) \\ \vdots & \ddots & \vdots & \vdots \\ c(x_n,x_1) & \cdots & c(x_n,x_n) & c(x_n,x^*) \\ c(x^*,x_1) & \cdots & c(x^*,x_n) & c(x^*,x^*) \end{pmatrix}$$

is positive semi-definite, in particular its determinant is non-negative.

We will denote the minors of the matrix C by $M_{i,j}$ and the minors of the matrix C' by $M'_{i,j}$. We compute this determinant by expanding along the last column:

$$\begin{aligned}\det(C') = &\sum_{i=1}^{n} c(x_i,x^*)\det(M'_{i,n+1})(-1)^{i+n+1} \\ &+ c(x^*,x^*)\det(\underbrace{M'_{n+1,n+1}}_{=C})\underbrace{(-1)^{2n+2}}_{=1}. \end{aligned} \tag{1}$$

Now compute $\det(M'_{i,m+1})$ by expanding along the last row:

$$\det(M'_{i,m+1}) = \sum_{j=1}^{n} c(x^*,x_j)\det(M_{i,j})(-1)^{j+n} \tag{2}$$

By reinserting (2) into (1), we obtain:

$$\begin{aligned}&\det(C') \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} c(x_i,x^*)\det(M_{i,j})(-1)^{i+j+2n+1}c(x^*,x_j) \\ &\quad + c(x^*,x^*)\det(C)\end{aligned} \tag{3}$$

Now note that

$$\det(C)C^{-1} = \big(\det(M_{i,j})(-1)^{i+j}\big)_{ij}.$$

Using this in (3), we obtain

$$\begin{aligned}&\det(C') \\ &= -\det(C)c(x^*,x)C^{-1}c(x,x^*) + \det(C)c(x^*,x^*) \\ &= \det(C)(c(x^*,x^*) - \mathrm{Var}(\hat{f}(x^*)))\end{aligned}$$

Rearranging this leads to

$$\begin{aligned}\mathrm{Var}(\hat{f}(x^*)) &= c(x^*,x^*) - \frac{\det C'}{\det C} \\ &\le c(x^*,x^*)\end{aligned}$$

This completes the proof. ■

4. Lower bound for variance

Similar to the upper bound in the previous section, we provide a lower bound in this section.

Proposition 4.1: For every $i = 1, 2, \ldots, n$ the following inequality holds:

$$\mathrm{Var}(\hat{f}(x^*)) \geq \frac{\big(c(x^*, x_i)\big)^2}{c(x_i, x_i)}.$$

In other words,

$$\mathrm{Var}(\hat{f}(x^*)) \geq \max_{i=1,2,\ldots,n} \frac{\big(c(x^*, x_i)\big)^2}{c(x_i, x_i)}.$$

Proof: Define

$$\langle \cdot, \cdot \rangle \;\; \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \;\; \langle a, b \rangle = a^T C^{-1} b.$$

This is a scalar product. So the Cauchy-Schwartz inequality holds:

$$\langle a, b \rangle^2 \leq \langle a, a \rangle \langle b, b \rangle$$

By setting $a = c(x, x^*)$ and $b = c(x, x_i)$, we obtain

$$\Big(\big(c(x, x^*)\big)^T \underbrace{C^{-1} c(x, x_i)}_{=e_i}\Big)^2 \leq \big(c(x_i, x_i)\big)^T \underbrace{c(x, x^*) C^{-1} c(x, x^*)}_{\mathrm{Var}(\hat{f}(x^*))}$$

where e_i denotes the ith unit vector. This can be simplified to

$$\big(c(x_i, x^*)\big)^2 \leq c(x_i, x_i)\, \mathrm{Var}(\hat{f}(x^*))$$

That completes the proof. ■

5. Correlation bounds

We bring the upper and lower bounds derived in the previous two sections together in a form that can be directly applied in examples.

Consider the ratio of the achieved to the intended correlations

$$R_i(x^*) = \frac{\rho\big(\overline{f}(x^*), \overline{f}(x_i)\big)}{\rho\big(f(x^*), f(x_i)\big)} = \frac{\mathrm{Cov}\,\big(\overline{f}(x^*), \overline{f}(x_i)\big)}{\mathrm{Cov}\,\big(f(x^*), f(x_i)\big)} = \frac{c(x^*, x^*)}{\mathrm{Var}\,\big(\hat{f}(x^*)\big)}.$$

Since we want to measure, how closely $\rho\big(f(x^*), f(x_i)\big)$ is approximated by $\rho\big(\overline{f}(x^*), \overline{f}(x_i)\big)$, we want to understand, how close $R_i(x^*)$ is to 1. We can apply the bounds found for $\mathrm{Var}(\hat{f}(x^*))$ to the ratio above to obtain:

$$\boxed{\min_{j=1,2,\ldots,n} \frac{c(x_j, x_j) c(x^*, x^*)}{\big(c(x^*, x_j)\big)^2} \geq R_i(x^*) \geq 1}$$

This implies that the technique proposed in this article generally overstates correlation. This can be understood intuitively by considering the extreme case where the coarser set consists of only one location. In that case pairs of locations from the finer set will receive full correlation, regardless of the intended correlation.

The inequality above, however, provides an upper bound to the correlation overstatement. In the next section we will show in an example that the deviation of the correlation from the desired value can be reduced by increasing the resolution of the coarser location set.

6. Controlling the approximation error in the simulation example

We return to the example of simulating standard normal variates for each location. In particular the variances of the variates is 1, so we have $c(x^*, x^*) = 1$ for ever location x^*. As an example of a target correlation assume that $c(\cdot, \cdot)$ be given by

$$c(x^*, x^{**}) = \exp(-0.002\, \mathrm{dist}(x^*, x^{**})),$$

where dist stands for the distance of two locations on the earth measured in kilometers.

We require that the ratio $R = R_i(x^*)$ is between 0.95 and 1.05, which amounts to allowing a 5% error in the simulation of correlation. In other words, we need to make sure that $R < 1.05$. This can be accomplished by assuring that

$$\min_{j=1,2,\ldots,n} \frac{1}{\big(c(x^*, x_j)\big)^2} \leq 1.05$$

This is equivalent to

$$\min_{j=1,2,\ldots,n} \exp\big(0.004\, \mathrm{dist}(x^*, x_j)\big) \leq 1.05$$

$$\min_{j=1,2,\ldots,n} \mathrm{dist}(x^*, x_j) \leq \frac{\ln(1.05)}{0.004} \approxeq 12.2$$

In other words, as long as the location x^* is less than 12.2 kilometers from the nearest location in the coarser location set, the desired accuracy is achieved.

7. Conclusion

Advanced catastrophe models simulate the correlation observed among the losses originating from a catastrophic event. The traditional copula approach is computationally infeasible due to the large number of locations to be considered in the event footprint. We provide an alternative approach to approximating the correlation and show how the accuracy of the approximation can be controlled. This allows catastrophe models with millions of simulated events to be implemented.

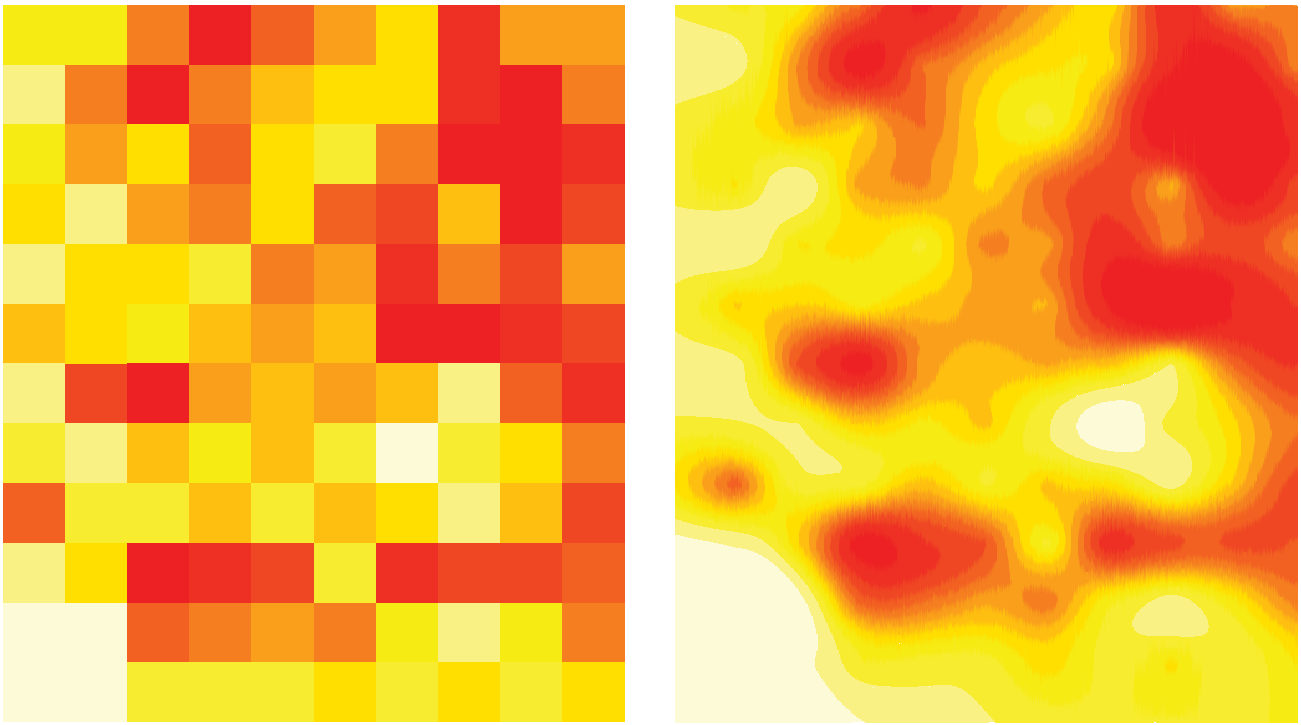

Fig. 1: Output from the R code. Visualization of the copulas with a coarse grid (left) and a finer grid (right).

Appendix

The code below runs in R version 3.1.0 once the MASS package is loaded. Its output can be viewed in Figure 1.

```
# This code is part of the article
    "Simulating Spatial Correlation
# for Catastrophic Events"

### Preset constants.
m1.nrow <- 12
m1.ncol <- 10
m1.num <- m1.nrow * m1.ncol
m2.nrow <- 720
m2.ncol <- 500
m2.num <- m2.nrow * m2.ncol

### Functions
target.cor <- function(x1, y1, x2, y2){
  dist <- sqrt((x1 - x2) ^ 2 + (y1 - y2) ^ 2)
  return(exp(- 0.01 * dist))
}

### Main Code
i1 <- rep(1:m1.nrow, times = m1.ncol)
j1 <- rep(1:m1.ncol, each = m1.nrow)

x1 <- i1 * m2.nrow / m1.nrow
y1 <- j1 * m2.ncol / m1.ncol

x2 <- rep(1:m2.nrow, times = m2.ncol)
y2 <- rep(1:m2.ncol, each = m2.nrow)

covar.mat <- outer(X = 1:m1.num, Y =
    1:m1.num, FUN = function(i, j)
  target.cor(x1 = x1[i], x2 = x1[j], y1 =
      y1[i], y2 = y1[j]))

covar.mat2 <- outer(X = 1:m1.num, Y =
    1:m2.num, FUN = function(i, j)
  target.cor(x1 = x1[i], x2 = x2[j], y1 =
      y1[i], y2 = y2[j]))

require(MASS)
set.seed(606)
sim.1 <- mvrnorm(mu = rep(0, m1.num), Sigma
    = covar.mat)
covar.mat.inv <- solve(covar.mat)
tmp.stdev <- sqrt(colSums((covar.mat.inv %*%
    covar.mat2) * covar.mat2))
sim.2 <- as.vector((t(sim.1) %*%
    covar.mat.inv) %*% covar.mat2) /
    tmp.stdev

layout(matrix(c(1,2), ncol = 2))
par(mar=c(0,1,0,1))
image(pnorm(matrix(sim.1, ncol = m1.nrow,
    byrow = TRUE)),
     xaxt="n", yaxt="n", bty = "n")
image(pnorm(matrix(sim.2, ncol = m2.nrow,
    byrow = TRUE)),
     xaxt="n", yaxt="n", bty = "n")
```

ISBN: 1-60132-427-8, CSREA Press ©