

Lecture 14. Simulation

Math 586

Intro

Kriging answers the question: what is the mean (expected) value of the random field, given the data, and what is the typical deviation from this value? The results are smooth curves that are suitable for producing maps but might be misleading as to the nature of the random field (“On average, we got ’im”).

Simulation attempts to reproduce the entire random field, and *conditional simulation* does it while preserving the data. This feature is desirable to many applications. It is often that one simulates, for example, a permeability field and then solves the differential equations of flow and transport to determine the movement of subsurface fluids, contamination etc. Many simulated samples of such a field might be generated as a part of a Monte-Carlo simulation of the flow.

To compare kriging and simulation results, consider the problem of estimating the length of a cable at the bottom of the ocean:

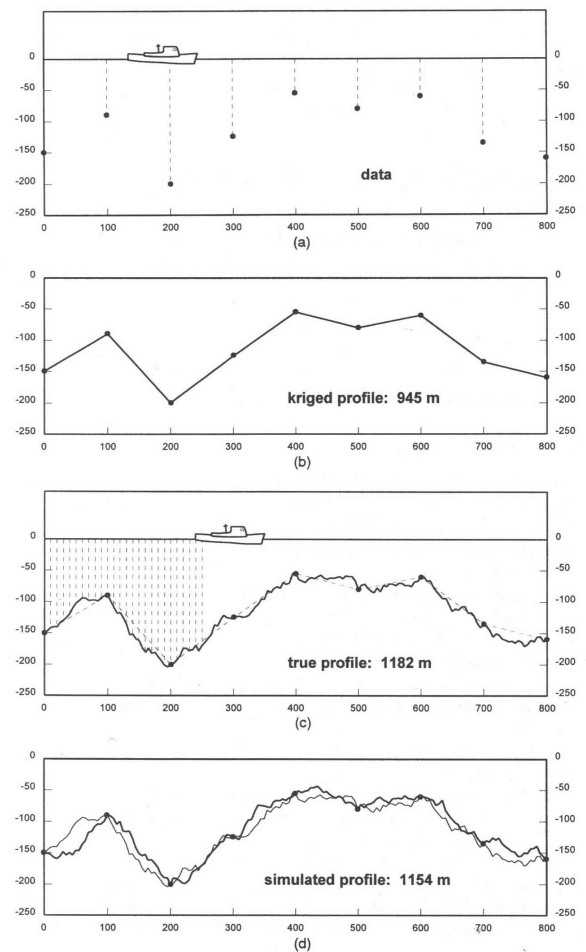


FIGURE 7.1. Submarine cable: (a) seafloor survey with 100-m spacing; (b) kriging estimate of the seafloor; (c) continuous survey (true profile); (d) conditional simulation based on the 100-m data, and true profile. From Alfaro (1979).

Of course, conditional simulation does not exactly reproduce the entire random field, but it gives a plausible impression of what that might look like. For Gaussian random field, it is faithful to the kriging results: at the point \mathbf{x}_0 , its distribution is normal with the mean equal to kriging mean $\hat{V}(\mathbf{x}_0)$ and variance equal to kriging variance. Moreover, the dependency pattern between

different locations follows the given variogram model.

Fortunately, one can use kriging to turn any simulation into a conditional one.

The algorithm for conditional simulation:

1. Start with the data $V(\mathbf{x}_i), i = 1, \dots, n$, and a covariance function $C(\mathbf{x}_1, \mathbf{x}_2)$. The goal is to “fill in” the rest of random field $V(\cdot)$ subject to the covariance behavior given by C (or variogram γ).
2. Generate a non-conditional simulated RF $S(\cdot)$ (below, we will discuss various methods for doing that).
3. For any point \mathbf{x} , consider the kriging estimator $V^*(\mathbf{x})$ and the one for $S^*(\mathbf{x})$ based on values of simulated $S(\mathbf{x}_i)$ at the data locations only.
- 4.

Set the

$$\begin{aligned} &\text{conditional simulation} \\ &= \\ &\text{kriging estimator} \\ &+ \\ &\text{simulation of kriging error} \end{aligned}$$

that is

$$T(\mathbf{x}) = V^*(\mathbf{x}) + [S(\mathbf{x}) - S^*(\mathbf{x})]$$

Compare with the true

$$V(\mathbf{x}) = V^*(\mathbf{x}) + [V(\mathbf{x}) - V^*(\mathbf{x})]$$

(see the picture)

Note that at data locations $T(\mathbf{x}_i) = V(\mathbf{x}_i)$. Also, kriging errors $[S(\mathbf{x}) - S^*(\mathbf{x})]$ have the same distribution as $[V(\mathbf{x}) - V^*(\mathbf{x})]$.

Thus, T has the same distribution as V .

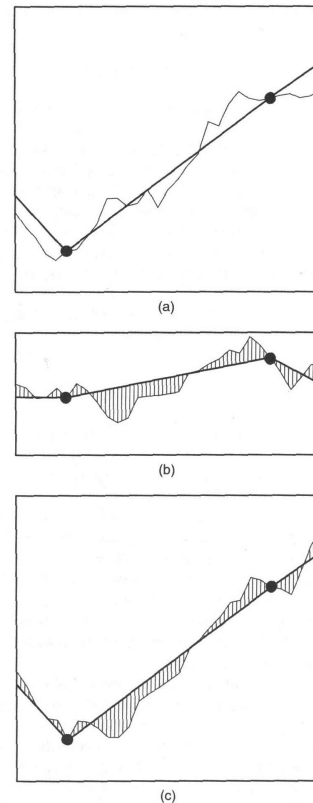


FIGURE 7.4. Conditioning a simulation: (a) real curve (unknown), sample points and kriging; (b) nonconditional simulation (known), sample points, and simulation of the kriging error; (c) kriging errors are picked from the simulation and added to the kriged curve.

Herewith, we will concentrate on the non-conditional simulation techniques.

Simulating univariate r.v.'s

At the core of any simulation algorithm, there is a random number generator, producing a value U taking $\text{Uniform}[0, 1]$ distribution. Given that, how do we simulate random fields?

It's relatively simple to simulate one-dimensional r.v.'s. Suppose we need a random variable X with CDF $F(x)$. Then X is obtained from uniform U as

$$X = F^{-1}(U)$$

Also, without trouble, we can simulate *independent* random variables (i.e. pure nugget random field) one by one. The trouble is to account for dependency structure (e.g. variogram) of the field.

For the remainder of the discussion, we will concentrate on Gaussian RF's (Normal distributions).

Direct matrix decomposition method

Given the $n \times n$ covariance matrix \mathbf{C} , simulate multivariate normal (mean 0) random vector \mathbf{X} with covariance $\mathbb{E}[\mathbf{X}\mathbf{X}'] = \mathbf{C}$.

Solution: Decompose $\mathbf{C} = \mathbf{S}\mathbf{S}'$. Then, generate a standard normal vector \mathbf{Z} with covariance equal to unit matrix \mathbf{I} . (Why is this easy?) Then, let $\mathbf{X} = \mathbf{S}\mathbf{Z}$. Note that

$$\mathbb{E}[\mathbf{X}\mathbf{X}'] = \mathbb{E}[\mathbf{S}\mathbf{Z}\mathbf{Z}'\mathbf{S}'] = \mathbf{S} \mathbb{E}[\mathbf{Z}\mathbf{Z}']\mathbf{S}' = \mathbf{S}\mathbf{S}' = \mathbf{C}$$

The actual method for finding such matrix \mathbf{S} is not unique. One can use e.g. the LU-decomposition, or Cholesky, or matrix square root.

Example:

Generate two r.v.'s with the variance 1 and correlation coefficient ρ .

We can decompose

$$\mathbf{C} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{bmatrix} \begin{bmatrix} 1 & \rho \\ 0 & \sqrt{1-\rho^2} \end{bmatrix} = \mathbf{S}\mathbf{S}'$$

Thus, $X_1 = Z_1$ and $X_2 = \rho Z_1 + \sqrt{1-\rho^2} Z_2$

Mean 0 is not really a restriction: you can always add any arbitrary mean at the end of your routine!

Matrix decomposition methods are all-purpose and very reliable, but it takes a lot of memory and time to decompose large matrices. Therefore, it puts a severe limit on the number of locations for which you can simulate.

Sequential Gaussian Simulation

Generates values of $V_i = V(\mathbf{x}_i)$ one by one using kriging. Generate i -th value based on kriging mean and variance obtained at the location \mathbf{x}_i from the data V_1, \dots, V_{i-1} .

It works no faster than the direct method, unless we use some local version of kriging. However, local methods would lead to imperfect results. One might occasionally run into a negative variance!

Spectral methods

These methods are considered the fastest and the most reliable. However, they can only work for the regularly-spaced grids. One can envision some combination of spectral and SGS method to fill in the gaps for the off-grid points.

Spectral methods are based on the idea: instead of directly computing the values of V , decompose V into a sum of some basis functions (for example, Fourier basis made up of sines and cosines) and only generate coefficients A_k , B_k , that is

$$V(x) = \sum_{k=0}^{\infty} A_k \cos(2\pi kx/T) + \sum_{k=1}^{\infty} B_k \sin(2\pi kx/T)$$

We will obtain a periodic function, so we will only keep the values of V for $0 \leq x < T$.

The practical implementation of this idea is based on Fast Fourier Transform (FFT). First, we will discretize the values of x , therefore we will obtain a simulation of V on a grid.

We will simulate the values $V_n = V(n\Delta x)$, $n = 1, \dots, N$. There is a correspondence between values V_n and (random) coefficients Z_k through discrete Fourier transform:

$$V_n = \frac{1}{N} \sum_{k=0}^{N-1} Z_k e^{2\pi i kn/N} \quad (1)$$

$$Z_k = \sum_{n=0}^{N-1} V_n e^{-2\pi i kn/N} \quad (2)$$

Z_k measures how much does k -th frequency contribute.

The symbol i represents the imaginary unit. If you feel uneasy about the equations, recall the Euler's formula

$$e^{it} = \cos(t) + i \sin(t)$$

However, if Z_k 's are real and symmetric ($Z_0 = Z_{N-1}, Z_1 = Z_{N-2}$ etc.; we will assume that N is even), then V_n 's are also real and symmetric. In fact, we could re-write the formulas above in terms of cosines then.

How do we generate Z_k ? Based on *power spectrum* F_k

$$F_k = \sum_{m=0}^{N-1} C_m e^{-2\pi i k m / N} \quad (3)$$

Compare this to (2).

Discrete spectral simulation algorithm in 1D:

(See Chilés and Delfiner, Geostatistics, 1999)

$$\begin{array}{ccc} \text{non-random} & C_m & \Rightarrow & F_k \\ & & & \Downarrow \\ \text{random} & V_n & \Leftarrow & Z_k \end{array}$$

1. Compute C_m , $m = 0, \dots, N/2$ and complete by symmetry.
2. Compute the discrete spectrum F_k
3. Generate independent random X_k , $k = 0, \dots, N/2$ with mean 0 and variances $\sigma_k^2 = NF_k/2$, except X_0 and $X_{N/2}$, with variance NF_k . Also, generate Y_k , $k = 1, \dots, N/2 - 1$ with variances $NF_k/2$ (with $Y_0 = 0$). Complete X 's by symmetry and Y 's by antisymmetry (that is, $Y_{N+1-k} = -Y_k$).

Then, set $Z_k = X_k + iY_k$, $k = 0, \dots, N - 1$

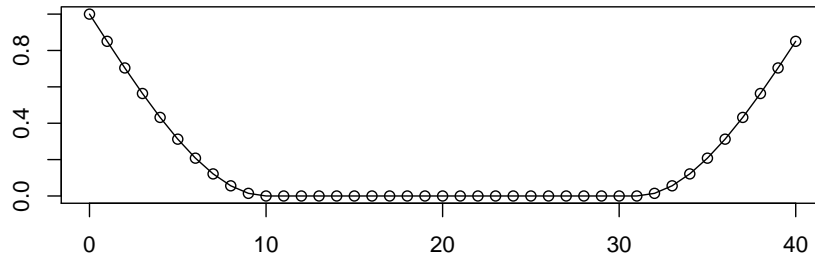
4. Obtain V_n using inverse Fourier transform (2). Since the correlation C_m was symmetrized, keep only $V_0, \dots, V_{N/2}$.

As an illustration, consider simulating from spherical covariance

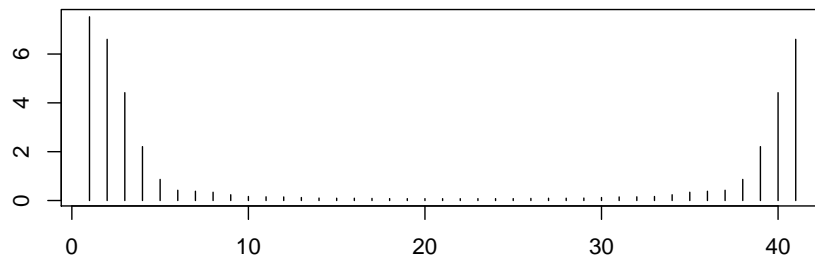
$$C(h) = 1 - 1.5(h/\ell) + 0.5(h/\ell)^3, \quad h \leq \ell$$

Let $\ell = 10$, $N = 40$ and $\Delta x = 1$.

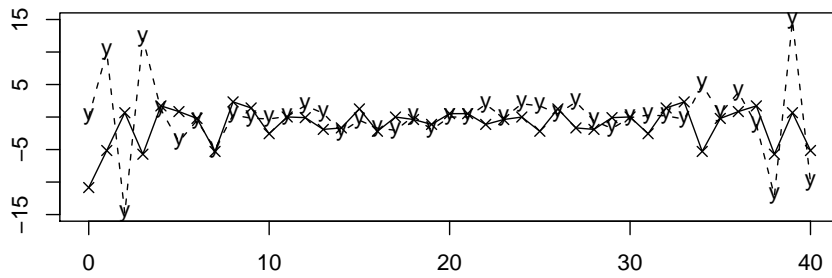
Discretized and symmetrized covariance



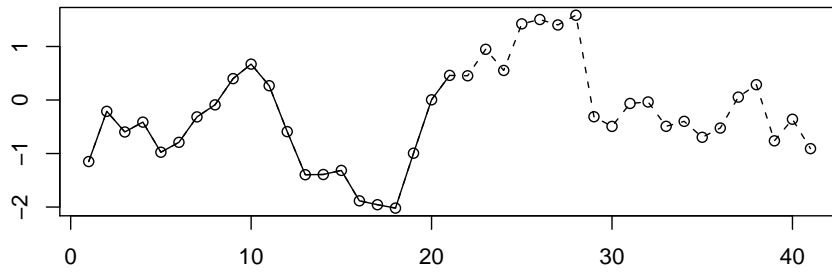
F_k



X and Y



Simulated V

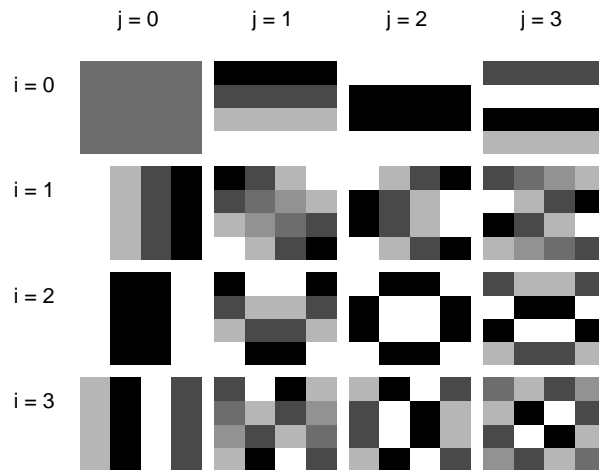


This approach can be extended to 2-D or 3-D as desired.

Example: discrete Fourier basis in 2D consists of functions

$$\Psi_{ij}(x, y) = \cos(2\pi ix/T) \cos(2\pi jy/T)$$

(grayscale level on the plot corresponds to values between -1 and 1 .)



These 16 basis functions may represent any function on 4×4 grid.

Spectral methods are also widely used in signal processing and geophysics. Such popular data compression formats as JPEG and MP3 are based on spectral methods.

Turning Bands

The standard reference is

Mantoglou, A. and Wilson, J.L. (1982) The turning bands method for simulation of random fields using line generation by a spectral method. *Water. Resour. Res.* 18, 1379-1394.

The method allows to turn one-dimensional simulations along given bands into 2- and 3-D random fields.

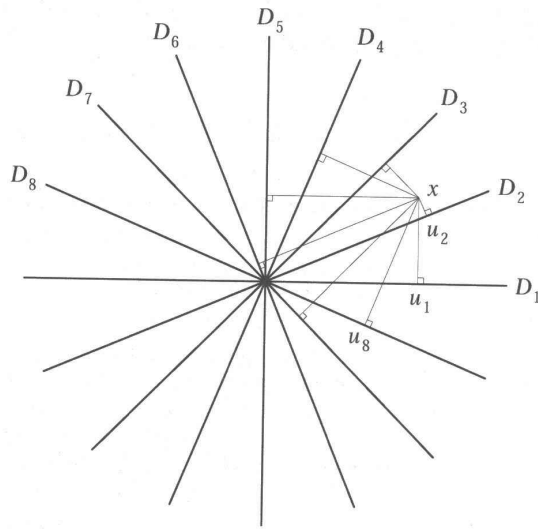


FIGURE 7.6. The principle of turning bands in 2D.