

Stochastic Simulations

Autumn Semester 2024

Prof. Fabio Nobile

Assistant: Matteo Raviola

Lab 03 – 26 September 2024

Multivariate random variable generation and Gaussian random processes

Exercise 1

Consider a multivariate Gaussian random variable $\mathbf{X} = (X_1, X_2, \dots, X_n)^T \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$.

1. Generate a sample of $N = 10^6$ independent random vectors $\mathbf{X}_i$, $1 \leq i \leq N$, each $\mathbf{X}_i$ following a $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ distribution with

$$\boldsymbol{\mu} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \text{and} \quad \Sigma = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}.$$

Specifically, use the Cholesky decomposition (`numpy.linalg.cholesky()` in Python¹) to compute the factor A , such that $\Sigma = AA^t$. Generate the standard normal vectors $\mathbf{Y}_i \sim \mathcal{N}(0, I_{2 \times 2})$, $1 \leq i \leq N$, possibly using Numpy's `np.random.randn()` function². Then generate the $\mathbf{X}_i$ as $\mathbf{X}_i = \boldsymbol{\mu} + A\mathbf{Y}_i$. Assess the quality of the samples by plotting a bivariate histogram³.

2. Propose a method for generating Gaussian random variables $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with covariance matrix $\Sigma = \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$ and mean $\boldsymbol{\mu}$ as before. Test your method by generating $N = 10^6$ independent copies of the random vector and plot a bivariate histogram. Compare the outcomes with the previous point and explain the differences.

Exercise 2

Consider a Gaussian process $\{X_t, t \in I\}$ on $I = [0, 1]$ with mean function $\mu_X: I \rightarrow \mathbb{R}$,

$$\mu_X(t) \equiv \mathbb{E}[X_t] = \sin(2\pi t),$$

and covariance function $C_X: I \times I \rightarrow \mathbb{R}$,

$$C_X(t, s) \equiv \mathbb{E}[(X_t - \mu_X(t))(X_s - \mu_X(s))] = e^{-|t-s|/\rho},$$

where $\rho > 0$.

¹<https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.cholesky.html>

²<https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.randn.html>

³<https://docs.scipy.org/doc/numpy/reference/generated/numpy.histogram2d.html>

1. Generate the Gaussian process in a set of n points $t_1, \dots, t_n \in I$. Plot the resulting point values of the random process for various values of n and ρ . Implement both direct generation and circulant embedding method with FFT. Compare the costs varying n which should behave as $\mathcal{O}(n^3)$ and $\mathcal{O}(n \log n)$, respectively.
2. Generate the Gaussian process for $\rho = 1/200$ on a uniform partition of $I = [0, 1]$ with $n = 51$ points, i.e. $t_i = \frac{i-1}{n-1}$ for $i = 1 \dots, n$. Let's denote this collection of point-wise evaluations of $\{X_t, t \in I\}$ by $\mathbf{Z}_n$. Then generate $m = n - 1 = 50$ additional point evaluations of the Gaussian process in new points $t_{n+1}, \dots, t_{n+m}$ by a uniform grid refinement (i.e. $t_{n+j} = \frac{2j-1}{2(n-1)}$ for $j = 1, \dots, m = n - 1$), denoted by $\mathbf{Y}_m$, conditioned upon the previously generated ones $\mathbf{Z}_n$. Specifically, use the results for conditioned multivariate Gaussian random variables discussed in the lecture notes.

Exercise 3

The lecture notes introduce a Brownian bridge as a Wiener process $\{W_t, t \in [0, 1]\}$ conditioned upon $W_1 = b$. Derive a generalized Brownian bridge $\{X_t, t \in [0, 1]\}$ that is given as the Wiener process conditioned on $W_0 = a$ and $W_1 = b$ and generate realizations of this Brownian bridge at $0 = t_0 < t_1 < \dots < t_n < t_{n+1} = 1$. Specifically, carry out the following exercises:

1. Show that

$$\mu_X(t) = a + (b - a)t$$

and

$$C_X(t, s) = \min\{s, t\} - st.$$

2. Propose and implement an iterative algorithm that generates X_{t_i} conditioned upon $X_{t_{i-1}}$ and $X_{t_{n+1}} = b$.

Exercise 4

Consider a fractional Brownian motion (fBM) $\{B^H(t), t \in [0, 1]\}$, which is a centered Gaussian process with $B^H(0) = 0$ and covariance function

$$\text{Cov}(t, s) = \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t - s|^{2H}),$$

where $H \in (0, 1)$ is the so-called *Hurst index*.

1. To sample such a process let us consider, for a fixed $h > 0$, the increment process $\delta B_h(t) = B^H(t + h) - B^H(t)$. Show that $\delta B_h(t)$ is a centered stationary Gaussian process.
2. If one is able to sample exactly the process $\delta B_h(t)$ on a uniform grid $t_j = jh$, then one can construct an exact sample of the fractional Brownian motion on the same grid points as $B^H(t_k) = \sum_{j=0}^{k-1} \delta B_h(t_j)$. Sample a fractional Brownian motion using FFT and circular embedding. Implement your experiment for different values of $H < 1/2$ and $H > 1/2$.

(Optional) Exercise 5

(Lévy-Ciesielski construction). Let $\xi_{-1} \sim \mathcal{N}(0, 1)$ and denote by $Y_0(t)$ the linear function on $[0, 1]$ with $Y_0(0) = 0$, and $Y_0(1) = \xi_{-1}$. For $j \in \mathbb{N}$ and $N = 0, 1, 2, \dots$, let $t_j^N = 2^{-N}j$ and let $Y_N(t)$ be the piecewise linear function such that

$$Y_{N+1}(t_j^N) = Y_N(t_j^N) \tag{1}$$

$$Y_{N+1}(t_{2j+1}^{N+1}) = \frac{1}{2} (Y_N(t_j^N) + Y_N(t_{j+1}^N)) + \xi_{j,N}, \quad \xi_{j,N} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 2^{-N-2}). \tag{2}$$

Here N is to be understood as a “discretization level” of the interval $[0, 1]$ on equal sub-intervals of length 2^{-N} . This process is known as the *Lévy-Ciesielski construction of a Brownian motion*.

1. Simulate the previous process for different values of N .
2. Prove that for any $N \in \mathbb{N}$, $\mathbb{E}[Y_N(t_j^N)] = 0$ and $\text{Cov}(Y_N(t_j^N), Y_N(t_k^N)) = \min\{t_j^N, t_k^N\}$, with $j, k = 0, \dots, 2^N$.
3. For any $s_1, \dots, s_m \in [0, 1]$, prove that $[Y_N(s_1), \dots, Y_N(s_m)]^T$ converges in distribution to a normal $\mathcal{N}(0, C)$, where C has entries $C_{ij} = \min\{t_j^N, t_i^N\}$, with $j, i = 0, \dots, 2^N$.
4. Write $W_n(t) := [Y_N(0), Y_N(t_1^N), \dots, Y_N(t_{2^N-1}^N), Y_N(1)]$. Conclude that $W_N(t) \rightarrow W(t)$, where $W(t)$ is the standard Brownian motion.