MATH-414 – Stochastic simulation

Lecture 3: Generation of stochastic processes

Prof. Fabio Nobile

Outline

Generation of Gaussian processes
- Wiener process
- Brownian bridge
- Stationary Gaussian processes – circulant embedding

Generation of discrete time Markov processes
- Discrete state space
- Continuous state space

Stochastic process / random field

Definition. *Let $I \subset \mathbb{R}^d$. A stochastic process $\{X_t,\ t \in I\}$ is a collection of random variables indexed by $t \in I$.*

Usually we use the terminology

- *stochastic process* when $d = 1$ and t denotes the time variable
- *random field* when $d \geq 1$ and t denotes a space variable.

Gaussian process

Definition. *A Gaussian process (or Gaussian random field)* $\{X_t,\ t \in I\}$ *is a stochastic process (random field) for which all finite dimensional distributions are Gaussian, i.e. for all* $n \in \mathbb{N}$ *and* $t_1, \ldots, t_n \in I$, *the random vector* $\boldsymbol{X} = (X_{t_1}, \ldots, X_{t_n})$ *has a multivariate Gaussian distribution.*

A Gaussian process is uniquely determined by

- **Mean function**: $\mu_X : I \to \mathbb{R}$,

$$\mu_X(t) = \mathbb{E}\left[X_t\right],\ t \in I$$

- **Covariance function**: $C_X : I \times I \to \mathbb{R}$,

$$C_X(t,s) = \mathbb{E}\left[(X_t - \mu_X(t))(X_s - \mu_X(s))\right],\ t, s \in I.$$

Notation: $X \sim N(\mu_X, C_X)$

Gaussian process generation

Let $t_1, t_2, \ldots, t_n \in I$ be fixed and $\boldsymbol{X} = (X_{t_1}, X_{t_2}, \ldots, X_{t_n})$. Then $\boldsymbol{X}$ has a multivariate Gaussian distribution (by definition)

$$\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma), \quad \text{with } \boldsymbol{\mu} = (\mu_X(t_1), \ldots, \mu_X(t_n)), \quad \Sigma_{ij} = C_X(t_i, t_j)$$

Question: is Σ positive definite?

Definition. *A function $C : I \times I \to \mathbb{R}$ is positive (semi-)definite if, for all n and $t_1, \ldots, t_n \in I$, the matrix $\Sigma \in \mathbb{R}^{n \times n}$, $\Sigma_{ij} = C(t_i, t_j)$ is positive (semi-)definite.*

The covariance function C_X of a stochastic process (random field) has to be a symmetric and positive (semi-)definite function.

- To generate $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$ we can proceed as in the last lecture by factorizing $\Sigma = AA^\top$ (Cholesky or spectral)
- Similarly, if we have generated already $\boldsymbol{X} = (X_{t_1}, \ldots, X_{t_n})$ and we want to generate the process in other points $\boldsymbol{Y} = (X_{t_{n+1}}, \ldots, X_{t_{n+k}})$, *conditional* on the values already generated, we can use the algorithm for conditional Gaussian generation from last lecture.

Wiener process

Definition. *The Wiener process is a Gaussian stochastic process* $\{W_t, t \geq 0\}$ *with*

- $W_0 = 0$,
- *Independent increments: for all* $0 < t_1 < t_2 \leq t_3 < t_4$, $(W_{t_2} - W_{t_1})$ *and* $(W_{t_4} - W_{t_3})$ *are independent random variables*
- *Gaussian stationary increments: for all* $0 \leq t_1 \leq t_2$, $W_{t_2} - W_{t_1} \sim N(0, t_2 - t_1)$

mean: $\mu_W(t) = \mathbb{E}[W_t] = \mathbb{E}[W_t - W_0] = 0$
covariance function: $C_W(t,s) = \min\{s, t\}$

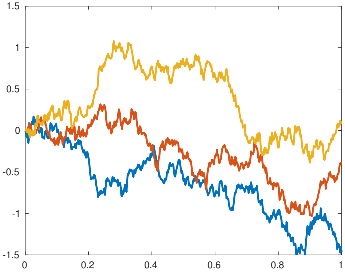
Indeed: $$C_W(t,s) = \mathbb{E}[W_t W_s] = \begin{cases} \mathbb{E}[(W_t - W_s)W_s] + \mathbb{E}\left[W_s^2\right] = s, & t \geq s \\ \mathbb{E}[(W_t(W_s - W_t)] + \mathbb{E}\left[W_t^2\right] = t, & t < s \end{cases}$$

Wiener process generation

Algorithm to generate $\{W_t,\ t \geq 0\}$ on a set of points $0 = t_0 < t_1 < \ldots < t_n$

Algorithm: Wiener process generation

Set $t_0 = 0$ and $W_{t_0} = 0$
for $k = 1, \ldots, n$ **do**
  Generate $\Delta W_k \sim N(0, t_k - t_{k-1})$
  Set $W_{t_k} = W_{t_{k-1}} + \Delta W_k$
end



Brownian bridge

Definition. *A Brownian bridge process* $\{X_t, t \in [0,1]\}$ *is a Wiener process* $\{W_t, t \in [0,1]\}$ *conditioned upon* $W_1 = b$.

mean: take $Y = W_t$, $Z = W_1$. Then $\Sigma_{YZ} = t$, $\Sigma_{ZZ} = 1$ and

$$\mu_X(t) = \mathbb{E}[X_t] = \mathbb{E}[Y \mid Z = b] = \mu_Y + \Sigma_{YZ}\Sigma_{ZZ}^{-1}(b - \mu_Z) = tb$$

covariance function: take $Y = (W_s, W_t)$, $Z = W_1$ so that

$$\Sigma_{YY} = \begin{pmatrix} s & \min\{s,t\} \\ \min\{s,t\} & t \end{pmatrix}, \quad \Sigma_{YZ} = \begin{pmatrix} s \\ t \end{pmatrix}, \quad \Sigma_{ZZ} = 1.$$

Therefore

$$\begin{aligned} \Sigma_{Y \mid Z} &= \Sigma_{YY} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZY} \\ &= \begin{pmatrix} s & \min\{s,t\} \\ \min\{s,t\} & t \end{pmatrix} - \begin{pmatrix} s \\ t \end{pmatrix} (s, t) \end{aligned}$$

and

$$\mathrm{Cov}_X(s,t) = (\Sigma_{Y|Z})_{12} = \min\{s,t\} - st$$

Generating a Brownian bridge

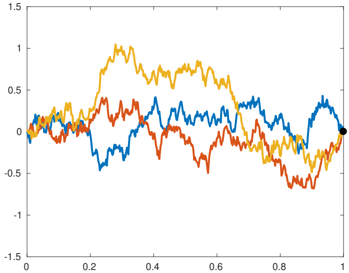
To generate a Brownian bridge in a set of points $0 < t_1 < \ldots < t_n < t_{n+1} = 1$ we can use the following algorithm

Algorithm: Brownian bridge generation.

Given: $0 < t_1 < \cdots < t_n < t_{n+1} = 1$ and b

1 Generate W_{t_i}, $i = 1, \ldots, n+1$ from standard Wiener process

2 Output $X_{t_i} = W_{t_i} + t_i(b - W_{t_{n+1}})$, $i = 1, \ldots, n$.



Stationary Gaussian processes

Definition. *A Gaussian process $\{X_t, t \in \mathbb{R}\}$ is*

- *weakly stationary* if $C_X(s,t) = \tilde{C}(t-s)$
- *strongly stationary* if also $\mu_X(t) = \mu$, independent of t.

Generation on a uniform grid $\boldsymbol{X} = (X_{t_0}, \dots, X_{t_n})$, with $t_j = t_0 + jh$, $j = 0, \dots, n$. Covariance matrix

$$\Sigma_{ij} = C_X(t_i, t_j) = \tilde{C}((j-i)h) \quad \Longrightarrow \quad \text{Toeplitz matrix}$$

$$\Sigma = \begin{pmatrix} \sigma_0 & \sigma_1 & \sigma_2 & \dots & & \sigma_n \\ \sigma_1 & \sigma_0 & \sigma_1 & \dots & & \sigma_{n-1} \\ \sigma_2 & \sigma_1 & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \sigma_1 \\ \sigma_n & \sigma_{n-1} & \dots & & \sigma_1 & \sigma_0 \end{pmatrix} \qquad \sigma_i = \tilde{C}(ih)$$

Circulant embedding

Consider the following circulant embedding of Σ:

$$\tilde{\Sigma} = \left(\begin{array}{cccccc|cccc}
\sigma_0 & \sigma_1 & \sigma_2 & \dots & \sigma_{n-1} & \sigma_n & \sigma_{n-1} & \sigma_{n-2} & \dots & \sigma_1 \\
\sigma_1 & \sigma_0 & \sigma_1 & \dots & & \sigma_{n-1} & \sigma_n & \sigma_{n-1} & \dots & \sigma_2 \\
\sigma_2 & \sigma_1 & \ddots & \ddots & & \vdots & \vdots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & & \vdots & \vdots & & \ddots & \sigma_{n-1} \\
\vdots & \vdots & & & \ddots & \sigma_1 & \sigma_2 & & & \sigma_n \\
\sigma_n & \sigma_{n-1} & \dots & & \sigma_1 & \sigma_0 & \sigma_1 & \dots & \dots & \sigma_{n-1} \\
\hline
\sigma_{n-1} & \sigma_n & \sigma_{n-1} & \dots & \dots & \sigma_1 & \sigma_0 & \sigma_1 & \dots & \sigma_{n-2} \\
\sigma_{n-2} & \sigma_{n-1} & \ddots & \ddots & & \sigma_2 & \sigma_1 & \sigma_0 & \dots & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
\sigma_1 & \dots & \dots & \sigma_{n-1} & \sigma_n & \sigma_{n-1} & \sigma_{n-2} & & \sigma_1 & \sigma_0
\end{array}\right) \in \mathbb{R}^{2n\times 2n}$$

In short, $\tilde{\Sigma} = \mathrm{circ}(\boldsymbol{\alpha})$, with $\boldsymbol{\alpha} = (\sigma_0, \sigma_1, \dots, \sigma_n, \sigma_{n-1}, \dots, \sigma_1)$.

$\tilde{\Sigma}$ can be easily diagonalized as $\tilde{\Sigma} F^* = F^* \Lambda \quad \Longrightarrow \quad \tilde{\Sigma} = \frac{1}{2n} F^* \Lambda F$

- F: Fourier matrix, $F_{k\ell} = e^{-2\pi i(\ell-1)(k-1)/2n}$, $\quad F^*F = FF^* = 2nI_{2n}$
- $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$, and $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_{2n})^T = F\boldsymbol{\alpha} = FFT(\boldsymbol{\alpha})$

$$\text{Hence} \quad \tilde{\Sigma} = AA^*, \qquad A = \frac{1}{\sqrt{2n}} F^* \Lambda^{1/2}$$

EPFL

Circulant embedding

Indeed, notice that

$$\tilde{\Sigma}_{jk} = \tilde{\Sigma}_{kj} = \alpha_{\{(2n+j-k+1) \mod 2n\}}$$

with $\alpha_0 := \alpha_{2n}$. Then

$$\begin{aligned}
\sum_{k=1}^{2n} \tilde{\Sigma}_{jk} F^*_{k\ell} &= \sum_{k=1}^{2n} \alpha_{\{(2n+j-k+1) \mod 2n\}} e^{2\pi i(\ell-1)(k-1)/2n} \\
&= \sum_{k=1}^{2n} \alpha_{\{(2n+j-k+1) \mod 2n\}} e^{2\pi i(\ell-1)(k-j-2n)/2n} e^{2\pi i(\ell-1)(j-1)/2n} \\
&\overset{s\,=\,2n+j-k+1}{=} \underbrace{\left(\sum_{s=1}^{2n} \alpha_s e^{-2\pi i(\ell-1)(s-1)/2n} \right)}_{\lambda_\ell} F^*_{j\ell}
\end{aligned}$$

Circulant embedding

Idea: let us try to generate our (zero mean) discretized random field as $\tilde{\boldsymbol{X}} = A\boldsymbol{Y}$ with $\boldsymbol{Y} = (Y_1, \ldots, Y_{2n})$ a vector of complex standard normal r.vs, i.e.

$$\boldsymbol{Y} = \boldsymbol{Y}_R + i\boldsymbol{Y}_I, \qquad \boldsymbol{Y}_R, \boldsymbol{Y}_I \overset{iid}{\sim} N(0, I_{2n})$$

Properties of $\tilde{\boldsymbol{X}}$

- $\mathbb{E}\left[\boldsymbol{Y}\boldsymbol{Y}^*\right] = 2I_{2n}$
- $\mathbb{E}\left[\boldsymbol{Y}\boldsymbol{Y}^T\right] = 0 = \mathbb{E}\left[\bar{\boldsymbol{Y}}\bar{\boldsymbol{Y}}^T\right]$,
- $\mathbb{E}\left[\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^*\right] = \mathbb{E}\left[A\boldsymbol{Y}\boldsymbol{Y}^*A^*\right] = 2\tilde{\Sigma}, \quad \mathbb{E}\left[\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\top\right] = 0 = \mathbb{E}\left[\bar{\tilde{\boldsymbol{X}}}\bar{\tilde{\boldsymbol{X}}}^T\right]$,
- $\mathbb{E}\left[\mathrm{Re}(\tilde{\boldsymbol{X}})\,\mathrm{Re}(\tilde{\boldsymbol{X}})^\top\right] = \mathbb{E}\left[\frac{\tilde{\boldsymbol{X}}+\bar{\tilde{\boldsymbol{X}}}}{2}\left(\frac{\tilde{\boldsymbol{X}}+\bar{\tilde{\boldsymbol{X}}}}{2}\right)^\top\right] = \tilde{\Sigma} = \mathbb{E}\left[\mathrm{Im}(\tilde{\boldsymbol{X}})\,\mathrm{Im}(\tilde{\boldsymbol{X}})^\top\right]$,
- $\mathbb{E}\left[\mathrm{Re}(\tilde{\boldsymbol{X}})\,\mathrm{Im}(\tilde{\boldsymbol{X}})^\top\right] = \mathbb{E}\left[\frac{\tilde{\boldsymbol{X}}+\bar{\tilde{\boldsymbol{X}}}}{2}\left(\frac{\tilde{\boldsymbol{X}}-\bar{\tilde{\boldsymbol{X}}}}{2i}\right)^\top\right] = 0.$

Conclusion:

- $\mathrm{Re}(\tilde{\boldsymbol{X}}), \mathrm{Im}(\tilde{\boldsymbol{X}}) \overset{iid}{\sim} N(0, \tilde{\Sigma})$
- $\mathrm{Re}(\tilde{\boldsymbol{X}}_{1:n+1}), \mathrm{Im}(\tilde{\boldsymbol{X}}_{1:n+1}) \overset{iid}{\sim} N(0, \Sigma)$

Circulant embedding

Notice that $\tilde{\boldsymbol{X}}$ can be computed efficiently by iFFT as

$$\tilde{\boldsymbol{X}} = A\boldsymbol{Y} = F^*(\frac{1}{\sqrt{2n}}\Lambda^{1/2}\boldsymbol{Y}) = iFFT(\sqrt{2n}\Lambda^{1/2}\boldsymbol{Y})$$

Algorithm: Circulant embedding.

Given: $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\Sigma = \begin{pmatrix} \sigma_0 & \sigma_1 & \dots & \sigma_n \\ \sigma_1 & \sigma_0 & \dots & \sigma_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_n & \dots & \dots & \sigma_0 \end{pmatrix} \in \mathbb{R}^{n+1 \times n+1}$

1 Generate the vector $\boldsymbol{\alpha} = (\sigma_0, \sigma_1, \dots, \sigma_n, \sigma_{n-1}, \dots, \sigma_1) \in \mathbb{R}^{2n}$

2 Compute $\boldsymbol{\lambda} = \mathrm{FFT}(\boldsymbol{\alpha})$ and $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$

3 Generate $\boldsymbol{Y} = \boldsymbol{Y}_R + i\boldsymbol{Y}_I$ with $\boldsymbol{Y}_R, \boldsymbol{Y}_I \overset{\text{iid}}{\sim} N(0, I_{2n})$

4 Compute $\tilde{\boldsymbol{X}} = \mathrm{iFFT}(\sqrt{2n}\Lambda^{1/2}\boldsymbol{Y})$

5 Output $\boldsymbol{X}^{(1)} = \boldsymbol{\mu} + \mathrm{Re}(\tilde{\boldsymbol{X}}_{1:n+1})$ and $\boldsymbol{X}^{(2)} = \boldsymbol{\mu} + \mathrm{Im}(\tilde{\boldsymbol{X}}_{1:n+1})$

Circulant embedding

Problem: the matrix $\tilde{\Sigma}$ might not be semi positive definite.

Possible remedy: enlarge the circulant embedding

$$\boldsymbol{\alpha} = (\sigma_0, \sigma_1, \dots, \sigma_n, \sigma^*_{n+1}, \dots, \sigma^*_m, \sigma^*_{m-1}, \dots, \sigma^*_{n+1}, \sigma_n, \dots, \sigma_1)$$

with $m > n$ large enough. Typical choice: $\sigma^*_j = \sigma_j = \tilde{C}(jh)$.

Discrete time / discrete space Markov chain

- State space $\mathcal{X} = \{y_1, y_2, \dots\}$ (finite or countable)
- Stochastic process on $\mathcal{X}$: $\{X_n \in \mathcal{X},\ n \in \mathbb{N}_0\}$

Definition. *A stochastic process $\{X_n \in \mathcal{X},\ n \in \mathbb{N}_0\}$ is a Markov chain if it satisfies the Markov property*

$$\mathbb{P}\left(X_{n+1} = y_{n+1} \mid X_n = y_n, X_{n-1} = y_{n-1}, \dots, X_0 = y_0\right) \\ = \mathbb{P}\left(X_{n+1} = y_{n+1} \mid X_n = y_n\right)$$

with $y_0, \dots, y_{n+1} \in \mathcal{X}$.

Transition matrix: $P_{ij}(n) = \mathbb{P}\left(X_n = y_j \mid X_{n-1} = y_i\right)$

- The Markov chain $\{X_n,\ n \in \mathbb{N}_0\}$ is entirely defined by
  - transition matrices $P(n)$, $n = 1, 2, \dots$
  - distribution of initial state $X_0 \sim \lambda$

  In short, $X_n \sim \mathrm{Markov}(\lambda, P(n))$
- $P(n)$ is a stochastic matrix: $\sum_j P_{ij}(n) = 1, \quad \forall i = 1, 2, \dots,$
- The Markov chain is time-homogeneous if $P(n)$ does not depend on n.

Generation of discrete time / discrete space Markov chains

Algorithm: Generation of discrete time / discrete space Markov chain

Given: λ and $P(n)$, $n \in \mathbb{N}_0$

Generate $X_0 \sim \lambda$

For $n = 1, 2, \ldots,$

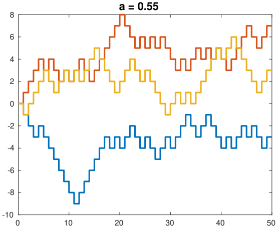
    Generate $X_n \sim P_{X_{n-1},:}(n)$ `// pmf:` X_{n-1}`-th row of` $P(n)$

Example: Random walk on integers. Start at $X_0 = 0$:

$$\mathbb{P}\left(X_{n+1} = j \mid X_n = j-1\right) = \mathbb{P}\left(X_{n+1} = j \mid X_n = j+1\right) = a \in (0,1),$$

$$\mathbb{P}\left(X_{n+1} = j \mid X_n = j\right) = 1 - 2a,$$

$$\mathbb{P}\left(X_{n+1} = j \mid X_n = i\right) = 0, \quad i \neq j, j-1, j+1.$$



Discrete time / continuous space Markov chains

- State space $\mathcal{X} \subset \mathbb{R}^d$ (continuous set)
- Stochastic process on $\mathcal{X}$: $\{X_n \in \mathcal{X},\ n \in \mathbb{N}_0\}$

Definition. *A Markov transition kernel on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, with $\mathcal{B}(\mathcal{X})$ the Borel σ-algebra, is a function $P : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0,1]$ such that*

- *for all $y \in \mathcal{X}$, $P(y, \cdot)$ is a probability measure on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$;*
- *for all $A \in \mathcal{B}(\mathcal{X})$, $P(\cdot, A)$ is a measurable function on $\mathcal{X}$.*

Often, the transition kernel is defined from a density function

$$P(x, A) = \int_A p(x,y)\, dy, \quad \text{with } p \geq 0,\ \int_{\mathcal{X}} p(x,y) dy = 1,\ \forall x$$

Definition. *$\{X_n,\ n \in \mathbb{N}_0\}$ is a homogeneous Markov chain on $\mathcal{X}$ with kernel $P : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0,1]$ and initial distribution $X_0 \sim \lambda$, denoted $\{X_n\} \sim \mathrm{Markov}(\lambda, P)$, if for any $n \in \mathbb{N}$, $A \in \mathcal{B}(\mathcal{X})$,*

$$\begin{aligned}\mathbb{P}\left(X_{n+1} \in A \mid X_n = y_n, \ldots, X_0 = y_0\right) \\ = \mathbb{P}\left(X_{n+1} \in A \mid X_n = y_n\right) = P(y_n, A)\end{aligned}$$

Generation of discrete time / cont. space Markov chains

Algorithm: Generation of discrete time / continuous space Markov process.

Given: λ and P

Generate $X_0 \sim \lambda$

For $n = 1, 2, \dots,$

  Generate $X_n \sim P(X_{n-1}, \cdot)$

Example: Continuous random walk in 2D. Start at $X_0 = 0$:

$$\boldsymbol{X}_{n+1} = \boldsymbol{X}_n + \boldsymbol{\xi}_n, \qquad \boldsymbol{\xi}_n \overset{\text{iid}}{\sim} N(0, \sigma^2 I_2).$$

Let $p(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)$ be the pdf of a Gaussian vector $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$. Then

$$\text{transition kernel:} \qquad P(\boldsymbol{y}, A) = \int_A p(\boldsymbol{x}; \boldsymbol{y}, \sigma^2 I_2) d\boldsymbol{x}$$