MATH-414 – Stochastic simulation

Lecture 3: Generation of stochastic processes

Prof. Fabio Nobile

Outline

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. *[Gaussian process] 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 $t_1, \dots, t_n \in I$, the random vector $\boldsymbol{X} = (X_{t_1}, \dots, X_{t_n})$ 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: $C_W(t,s) = \min\{s,t\}$

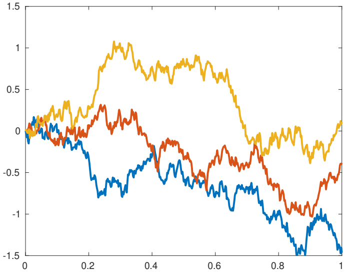
Indeed: $$C_W(t,s) = \mathbb{E}[W_t W_s] = \begin{cases} \mathbb{E}[(W_t - W_s)W_s] + \mathbb{E}\left[W_s^2\right] = s, & t \geq s \\ \mathbb{E}[(W_t(W_s - W_t)] + \mathbb{E}\left[W_t^2\right] = t, & t < s \end{cases}$$

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

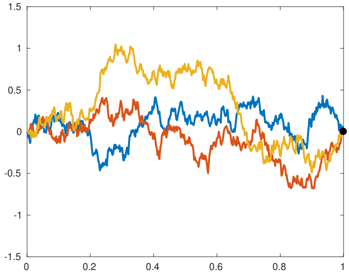
---

Algorithm: Brownian bridge generation.

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

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

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

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

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

Circulant embedding

Consider the following circulant embedding of Σ:

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

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

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

- 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 $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{2n})^T = F\boldsymbol{\alpha} = FFT(\boldsymbol{\alpha})$

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

Circulant embedding

Indeed, notice that $\tilde{\Sigma}_{jk} = \alpha_{\{(2n+k-j+1) \mod 2n\}}$ (with $\alpha_0 := \alpha_{2n}$) Then

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

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 $\tilde{\boldsymbol{X}}$

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

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

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

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 $n = 1, 2, \dots,$

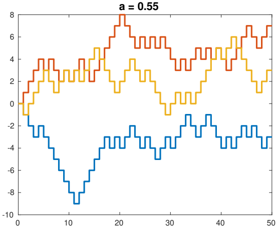
  Generate $X_n \sim P_{X_{n-1},:}(n)$ `// pmf:` X_{n-1}`-th row of` $P(n)$

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 $n \in \mathbb{N}$, $A \in \mathcal{B}(\mathcal{X})$,*

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

EPFL

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 $n = 1, 2, \ldots,$
    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

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

Continuous time / discrete state Markov chains

- State space $\mathcal{X} = \{y_1, y_2, \dots\}$ (finite or countable isolated points)
- Stochastic process on $\mathcal{X}$: $\{X_t \in \mathcal{X},\ t \geq 0\}$

A process $\{X_t, t \geq 0\}$ is right continuous if for any realization ω,

$$\lim_{h \to 0^+} X_{t+h}(\omega) = X_t(\omega).$$

A right continuous discrete state Markov process is piecewise constant

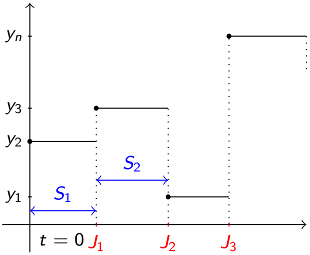
- Jump times: $J_0 = 0$,

$$J_n = \inf\{t \geq J_{n-1} : X_t \neq X_{J_{n-1}}\}, \quad n > 0$$

- Holding times:

$$S_n = \begin{cases} J_n - J_{n-1}, & \text{if } J_{n-1} < \infty, \quad n = 1, 2, \dots \\ \infty, & \text{otherwise.} \end{cases}$$

- Jump process: $\{Y_n = X_{J_n},\ n \in \mathbb{N}_0\}$



Poisson process

Definition. *A Poisson process $\{N_t \in \mathbb{N}_0, t \geq 0\}$ with initial state $N_0 = 0$ and parameter $0 < \lambda < \infty$, is a non decreasing, right-continuous, integer valued process which satisfies:*

1. *Independent increments: for all $0 < t_1 < t_2 \leq t_3 < t_4$,*

$$N_{t_2} - N_{t_1} \quad \text{is independent of} \quad N_{t_4} - N_{t_3}$$

2. *Poisson stationary increments: for all $0 < s < t$, $N_t - N_s \sim \mathrm{Pois}(\lambda(t-s))$ i.e.*

$$\mathbb{P}\left(N_t - N_s = j\right) = \frac{(\lambda(t-s))^j}{j!} e^{-\lambda(t-s)}.$$

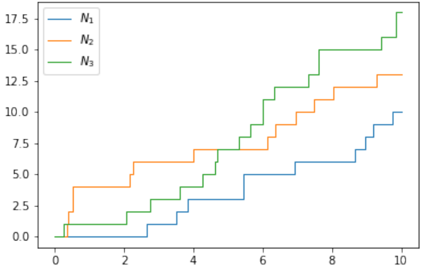
The Poisson process is a (continuous time / discrete state) Markov process. Hence $\{\tilde{N}_t = N_{s+t} - N_s,\ t \geq 0\}$ is also a Poisson process of parameter λ, intependent of $\{N_t,\ t \leq s\}$.

Equivalent characterizations of the Poisson process

a. The holding times $S_1, S_2, \ldots$ are independent exponential random variables $\mathrm{Exp}(\lambda)$ and the jump chain is $Y_n = N_{J_n} = n$.

Indeed

- $\mathbb{P}(S_1 > t) = \mathbb{P}(N_t = 0) = e^{-\lambda t} \quad \Rightarrow \quad S_1 \sim \mathrm{Exp}(\lambda)$.
- $\mathbb{P}(S_{n+1} > t) = \mathbb{P}(N_{J_n+t} - N_{J_n} = 0) = e^{-\lambda t}, \quad \Rightarrow \quad S_{n+1} \sim \mathrm{Exp}(\lambda)$
  Moreover, S_{n+1} is independent of $S_1, \ldots, S_n$ by independence of increments of N_t.



Equivalent characterizations of the Poisson process

b. For any $t > 0$ and $h \to 0^+$, uniformly in t it holds

$$\begin{aligned} \mathbb{P}\left(N_{t+h} - N_t = 0\right) &= 1 - \lambda h + o(h), \\ \mathbb{P}\left(N_{t+h} - N_t = 1\right) &= \lambda h + o(h), \\ \implies \mathbb{P}\left(N_{t+h} - N_t > 1\right) &= o(h). \end{aligned}$$

c. Conditional on $N_t = n$, the n jump times are uniformly distributed in $(0, t)$, i.e. $J_1, \ldots, J_n$ have the same distribution of the order statistics $U_{(1)}, \ldots U_{(n)}$ with $U_i \overset{\text{iid}}{\sim} \mathcal{U}(0, t)$.

Generation of a Poisson process

Generation based on property a.

Algorithm: Poisson process – version I.

Set $N_0 = 0,\ J_0 = 0,\ Y_0 = 0$
For $n = 1, 2, \ldots,$
    Generate $S_n \sim \mathrm{Exp}(\lambda)$ and set $J_n = J_{n-1} + S_n$
    Set $N_t = N_{J_{n-1}},\ t \in [J_{n-1}, J_n)$ and $N_{J_n} = N_{J_{n-1}} + 1$.

Generation on an interval $[0, T]$ based on procerty c.

Algorithm: Poisson process – version II.

Generate $N_T \sim \mathrm{Pois}(\lambda T)$
Generate $U_1, \ldots, U_{N_T} \overset{\text{iid}}{\sim} \mathcal{U}(0, T)$
Order the sample $U_{(1)} < \cdots < U_{(N_T)}$
Set $J_0 = 0,\ J_n = U_{(n)}$, and $N_t = n,\ t \in [J_n, J_{n+1}),\ n = 1, \ldots, N_T$

Non homogeneous Poisson process

Definition. $\{N_t,\ t \geq 0, N_0 = 0\}$ *is a non-homogeneous Poisson process with rate* $\lambda : [0, \infty) \to \mathbb{R}_+$ *if it is a right-continuous process with independent increments, such that*

$$\mathbb{P}\left(N_{t+h} - N_t = 0\right) = 1 - \lambda(t)h + o(h),$$
$$\mathbb{P}\left(N_{t+h} - N_t = 1\right) = \lambda(t)h + o(h).$$

Lemma

Distribution of the holding times:

$$F_{n+1}(t) = \mathbb{P}\left(S_{n+1} \leq t\right) = 1 - \exp\left\{-\int_{J_n}^{J_n+t} \lambda(s)\, ds\right\}.$$

Proof of the Lemma

$$
\begin{aligned}
F'_{n+1}(t) &= \lim_{h\to 0} \frac{F_{n+1}(t+h) - F_{n+1}(t)}{h} \\
&= \lim_{h\to 0} \frac{\mathbb{P}\left(t < S_{n+1} \le t+h\right)}{h} = \lim_{h\to 0} \frac{\mathbb{P}\left(S_{n+1} \le t+h \mid S_{n+1} > t\right)}{h}(1 - F_{n+1}(t)) \\
&= \lim_{h\to 0} \frac{\mathbb{P}\left(N_{J_n+t+h} > n \mid N_{J_n+t} = n\right)}{h}(1 - F_{n+1}(t)) \\
&= \lim_{h\to 0} \frac{1 - \mathbb{P}\left(N_{J_n+t+h} = n \mid N_{J_n+t} = n\right)}{h}(1 - F_{n+1}(t)) \\
&= \lambda(J_n + t)(1 - F_{n+1}(t))
\end{aligned}
$$

Solving the ODE on F_{n+1} gives the desired result.

Generation of non homogeneous Poisson process

Algorithm: Non-homogeneous Poisson process.

Set $N_0 = 0,\ J_0 = 0,\ Y_0 = 0$
For $n = 1, 2, \ldots$
    Generate $S_n \sim F_n(t) = 1 - \exp\left\{-\int_{J_{n-1}}^{J_{n-1}+t} \lambda(s)\, ds\right\}$
    Set $J_n = J_{n-1} + S_n$,
    Set $N_t = N_{J_{n-1}},\ t \in [J_{n-1}, J_n)$,
    Set $N_{J_n} = N_{J_{n-1}} + 1$

Alternative construction:

- Define $\Lambda(t) = \int_0^t \lambda(s)\, ds$
- Let $\tilde{N}_t$ be a homogeneous Poisson process with rate 1
- Then,

$$N_t = \tilde{N}_t \circ \Lambda = \tilde{N}_{\Lambda(t)}$$

is a non-homogeneous Poisson process with rate $\lambda(t)$.

General continuous time / discrete space Markov chain

Let $\{X_t, t \geq 0\}$ be a continuous time Markov chain on the discrete state space $\mathcal{X} = \{y_1, y_2, \ldots\}$.
Then, $\{X_t\}$ is fully characterized by

- distribution of initial state $X_0 \sim \mu$ (with μ a pmf on $\mathcal{X}$)
- the transition probabilities (jump rates)

$$q_{ij}(t) = \lim_{h \to 0^+} \frac{\mathbb{P}\left(X_{t+h} = j \mid X_t = i\right)}{h}$$
$$q_i(t) = \lim_{h \to 0^+} \frac{1 - \mathbb{P}\left(X_{t+h} = i \mid X_t = i\right)}{h}$$

The process is homogeneous if q_{ij} and q_i do not depent on t.

Generator of the Markov process: $Q_{ij} = \begin{cases} q_{ij} & i \neq j \\ -q_i & i = j \end{cases}$

- Q is stable if $q_i < \infty$, $\forall i$
- Q is conservative if $q_i = \sum_{j \neq i} q_{ij}$, $\forall i$

General continuous time / discrete space Markov chain

Definition. *A homogeneous continuous time Markov chain $\{X_t \in \mathcal{X},\ t \geq 0\}$ with initial state $X_0 \sim \mu$ and stable and conservative generator matrix Q, is a right-continuous, piecewise constant process denoted $\mathrm{Markov}\,(\mu, Q)$ s.t.*

- *the jump process $\{Y_n = X_{J_n},\ n \in \mathbb{N}_0\}$ is a discrete time Markov chain with transition probability*

$$\begin{aligned} &\pi_{ij} = \frac{q_{ij}}{q_i}, \quad i \neq j, \quad \pi_{ii} = 0, && \text{if } q_i \neq 0 \\ &\pi_{ij} = 0, \quad i \neq j, \quad \pi_{ii} = 1, && \text{if } q_i = 0. \end{aligned}$$

- *conditional on $Y_0, Y_1, \ldots, Y_{n-1}$, the holding times $S_1, \ldots, S_n$ are independent random variables, $S_i \sim \mathrm{Exp}(q_{Y_{i-1}}), \quad i = 1, \ldots, n$.*

Generation of cont. time / discrete space Markov chain

Algorithm: $\mathrm{Markov}\,(\mu, Q)$.

Generate $X_0 \sim \mu$ and set $J_0 = 0,\ Y_0 = X_0$

For $n = 1, 2, \ldots$

  Generate $S_n \sim \mathrm{Exp}(-Q_{Y_n Y_n})$ and set $J_n = J_{n-1} + S_n$,

  Generate $Y_{n+1} \sim \pi_{Y_n,\cdot}$

  Set $X_t = Y_n,\ t \in [J_{n-1}, J_n)$, and $X_{J_{n+1}} = Y_{n+1}$

Example – Poisson process

Generator (Q-matrix) of a homogeneous Poisson process of rate $\lambda > 0$:

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & \dots \\ 0 & -\lambda & \lambda & \dots \\ 0 & \ddots & \ddots & \ddots \end{bmatrix}$$

Indeed

$$\begin{aligned} q_i &= -Q_{ii} = \lim_{h \to 0^+} \frac{1 - \mathbb{P}\left(N_{t+h} = i \mid N_t = i\right)}{h} = \lambda \\ q_{i,i+1} &= Q_{i,i+1} = \lim_{h \to 0^+} \frac{\mathbb{P}\left(N_{t+h} = i+1 \mid N_t = i\right)}{h} = \lambda \\ q_{i,j} &= Q_{i,j} = 0, \quad j \neq i, i+1. \end{aligned}$$

Meaning of the generator Q

Let

- $p_i(t) = \mathbb{P}(X_t = y_i)$
- $\mathbf{p}(t) = (p_1(t), p_2(t), \dots)$ (row vector)

$$
\begin{aligned}
\frac{dp_i}{dt}(t) &= \lim_{h \to 0^+} \frac{p_i(t+h) - p_i(t)}{h} = \lim_{h \to 0^+} \frac{1}{h}\left(\mathbb{P}(X_{t+h} = y_i) - p_i(t)\right) \\
&= \lim_{h \to 0^+} \frac{1}{h}\Bigg(\sum_{j \neq i} \underbrace{\mathbb{P}(X_{t+h} = y_j \mid X_t = y_i)}_{= q_{ij}(t)h + o(h)} p_j(t) \\
&\qquad + \underbrace{\mathbb{P}(X_{t+h} = y_i \mid X_t = y_i)}_{= 1 - q_i(t)h + o(h)} p_i(t) - p_i(t) \Bigg) \\
&= \sum_{j \neq i} q_{ij}(t) p_j(t) - q_i(t) p_i(t) = \sum_j p_i(t) Q_{ij}(t)
\end{aligned}
$$

$$
\Longrightarrow \qquad \frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t) Q(t)
$$

EPFL