MATH-414 – Stochastic simulation

Lecture 2: Random variable Generation

Prof. Fabio Nobile

EPFL MATHICSE

Outline

Random Variable Generation

We have seen in the last lecture examples of *uniform (pseudo)-random number generators*.

In this Lecture we ask the question how to construct a non-uniform (pseudo)-random number generator, i.e. a generator that produces a stream of iid random variables $X_1, X_2, \ldots$, having a non uniform cumulative distribution function F.

The idea will be to start from a uniform RNG and make suitable transformations to obtain the desired non-uniform RNG.

Inverse transform method – discrete rvs

- X: discrete rv taking values $x_1 < x_2 < \ldots < x_n$
- probability mass function (pmf): $p_i = \mathbb{P}(X = x_i)$.
- cumulative probabilities: $F_i = \sum_{j=1}^{i} p_j = \mathbb{P}(X \leq x_i)$

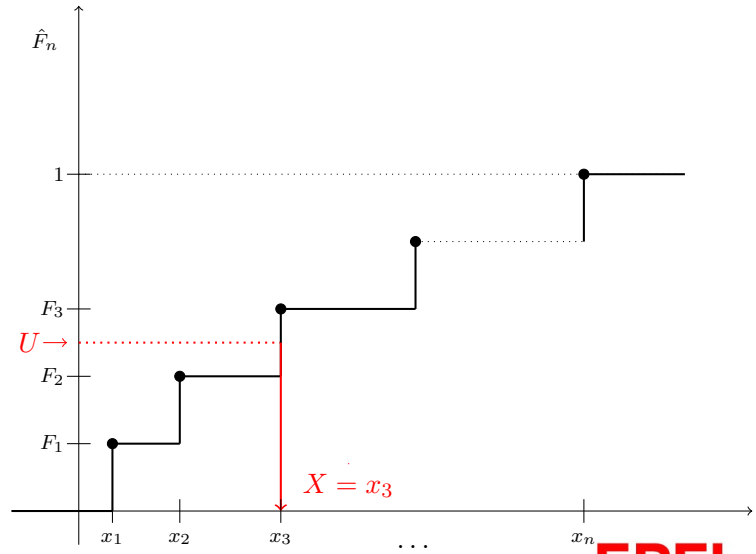
Algorithm: Discrete inverse-transform.

---

1 Generate $U \sim \mathcal{U}([0,1])$
2 Set $X = x_i$ if $F_{i-1} < U \leq F_i$

Indeed

$$\begin{aligned}\mathbb{P}(X = x_i) &= \mathbb{P}(F_{i-1} < U \leq F_i) \\ &= \mathbb{P}(U \subset (F_i - p_i, F_i]) \\ &= p_i\end{aligned}$$



Example Bernoulli rv $X \sim \mathrm{Be}(p)$

$$\begin{cases} X = 1, & \text{if } U > 1 - p \\ X = 0, & \text{otherwise} \end{cases} \qquad U \sim \mathcal{U}([0,1])$$

Inverse transform method – continuous rvs

- X: continuous rv
- cdf $F : [a, b] \to [0, 1]$ continuous and strictly increasing

---

Algorithm: Continuous inverse-transform

---

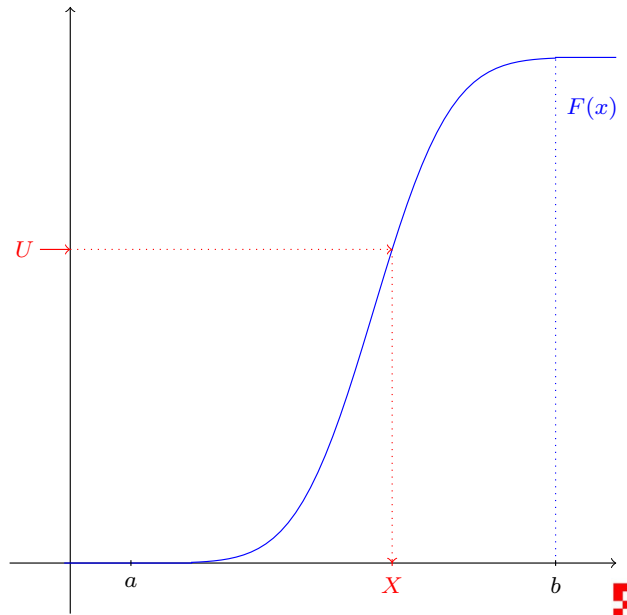
1 Generate $U \sim \mathcal{U}([0, 1])$
2 Set $X = F^{-1}(U)$

Indeed

$$\begin{aligned}\mathbb{P}(X \leq x) &= \mathbb{P}\left(F^{-1}(U) \leq x\right) \\ &= \mathbb{P}(U \leq F(x)) = F(x)\end{aligned}$$

Example Exponential, $X \sim Exp(\lambda)$,
cdf: $F(x) = 1 - e^{-\lambda x}$

$$\begin{aligned}X = F^{-1}(U) &= -\frac{1}{\lambda}\log(1 - U) \\ &\sim -\frac{1}{\lambda}\log(U)\end{aligned}$$



Inverse transform method – general case

The discrete and continuous cases can be combined in a unifying formula by defining a *generalized inverse cdf*

$$F^{-}(u) = \inf\{x : F(x) \geq u\}$$

Then

$$U \sim \mathcal{U}([0,1]) \quad \Longrightarrow \quad X = F^{-}(U) \sim F$$

Composition method

Consider a rv X with a mixture distribution

$$\textit{cdf} \qquad F(x) = \sum_{i=1}^{n} p_i F_i(x)$$

- $F_i,\ i = 1, \ldots, n$ are cdf functions
- $p_i > 0,\ i = 1, \ldots, n$ and $\sum_{i=1}^{n} p_i = 1$

Algorithm: Composition method

---

1. Generate discrete r.v. Y, $\mathbb{P}(Y = i) = p_i$
2. Generate $X \sim F_Y$ e.g. by inversion

Example: sampling from Laplace distribution

$$X \sim \textit{Lapl}(\lambda), \quad \textit{pdf}\ f(x) = \frac{\lambda}{2} e^{-\lambda|x|} = \frac{1}{2} \underbrace{\lambda e^{-\lambda x} \mathbb{1}_{\{x \geq 0\}}}_{\sim \mathrm{Exp}(1)} + \frac{1}{2} \underbrace{\lambda e^{\lambda x} \mathbb{1}_{\{x < 0\}}}_{\sim -\mathrm{Exp}(1)}.$$

$$B \sim \mathrm{Be}(\frac{1}{2}), \quad Y \sim \mathrm{Exp}(\lambda), \quad X = \begin{cases} Y, & \text{if } B = 1 \\ -Y, & \text{if } B = 0 \end{cases}$$

Equivalently, $X = (2B - 1)Y$

Acceptance-Rejection method

Consider a continuous rv X with pdf f and cdf F. Inverse transform method not applicable if

- F is difficult to invert
- f known only up to a multiplicative constant (e.g. in Bayesian statistics, statistical physics, etc.)

How to generate X in this case? AR could be a good option.

Assume that

- We can access a non-normalized version $\tilde{f}$ of f. Let $\kappa = (\int_{\mathbb{R}} \tilde{f}(x)\,dx)^{-1}$ so that $f(x) = k\tilde{f}(x)$
- There exists a pdf g and $C > 0$ such that

$$\tilde{f}(x) \leq Cg(x), \quad \forall x \in \mathbb{R}$$

- sampling from g is "easy"

Acceptance-Rejection method

Algorithm: Acceptance-Rejection (AR) algorithm

1. Generate $Y \sim g$
2. Generate $U \sim \mathcal{U}([0,1])$ independent of Y
3. If $U \leq \dfrac{\tilde{f}(Y)}{Cg(Y)}$ set $X = Y$, otherwise return to step 1

Why does it work?

Observe that $X \sim Y|\{U \leq \dfrac{\tilde{f}(Y)}{Cg(Y)}\}$. Therefore

$$\mathbb{P}(X \leq x) = \mathbb{P}\left(Y \leq x \,\middle|\, U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right) = \frac{\mathbb{P}\left(Y \leq x,\, U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right)}{\mathbb{P}\left(U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right)} = \frac{N}{D}$$

$$N = \int_{-\infty}^{x} \mathbb{P}\left(U \leq \frac{\tilde{f}(Y)}{Cg(Y)} | Y = y\right) g(y)dy = \int_{-\infty}^{x} \frac{\tilde{f}(y)}{Cg(y)} g(y)\, dy = \frac{1}{C}\int_{-\infty}^{x} \tilde{f}(y)\, dy$$

$$D = \mathbb{P}\left(U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right) = \int_{\mathbb{R}} \frac{\tilde{f}(y)}{Cg(y)} g(y)\, dy = \frac{1}{C}\int_{\mathbb{R}} \tilde{f}(y)\, dy = \frac{1}{\kappa C}$$

Hence $\mathbb{P}(X \leq x) = \int_{-\infty}^{x} f(y)\, dy = F(x)$

Acceptance-Rejection method

Remarks

- Probability of acceptance: $\mathbb{P}\left(U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right) = \frac{1}{\kappa C}$
- The method is efficient if $C \approx \kappa^{-1}$

Geometric interpretation

denote $A_{\tilde{f}} = \{(y,u) : 0 \leq u \leq \tilde{f}(y)\}$
and $A_{Cg} = \{(y,u) : 0 \leq u \leq Cg(y)\}$

1. generate $Y \sim g$, $U \sim \mathcal{U}([0,1])$
2. set $\tilde{U} = Cg(Y)U$ Then $(Y, \tilde{U}) \in A_{Cg}$
3. if $(Y, \tilde{U}) \in A_{\tilde{f}}$ set $X = Y$, otherwise return to step 1

Indeed, the (black) points are uniformly distributed in A_{Cg} since their joint pdf $f_\bullet(y,u) = f_{\tilde{U}|Y}(u|y) f_Y(y) = \frac{1}{Cg(y)} g(y) = \frac{1}{|A_{Cg}|}$. Hence, the retained (blue) points $(X, \tilde{U})$ are uniformly distributed on $A_{\tilde{f}}$ and their joint pdf is $f_\bullet(x,u) = \frac{1}{|A_{\tilde{f}}|} = \kappa^{-1}$ and the X-marginal distribution is $\int_0^{\tilde{f}(x)} f_\bullet(x,u)du = \frac{\tilde{f}(x)}{\kappa} = f(x)$ which shows that X has the correct pdf.

Acceptance-Rejection method

Example: Let $Z \sim N(0,1)$: sample from $X \sim Z|\{Z \geq 1\}$

$$\tilde{f}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathbb{1}_{\{x \geq 1\}}$$

Approach 1: generate $Y \sim N(0,1)$; if $Y \geq 1$ set $X = Y$, otherwise repeat.

Corresponds to AR with $g(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$.

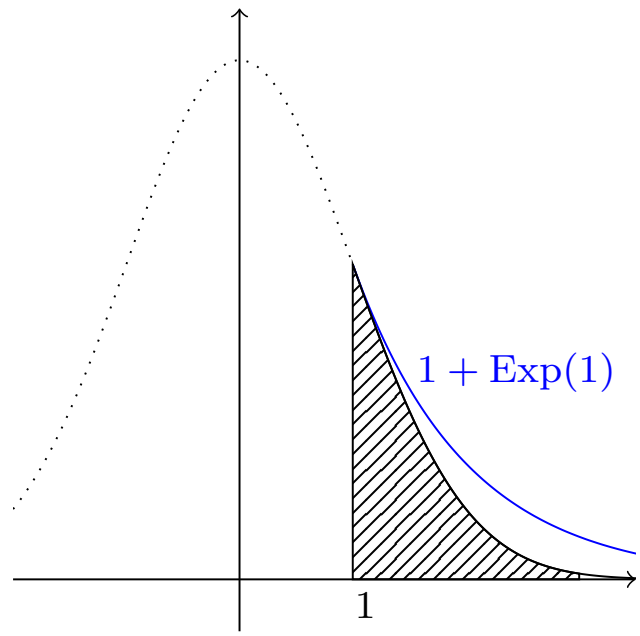
Acceptance rate ≈ 0.16

Approach 2: AR with $g(x) = e^{-(x-1)} \mathbb{1}_{\{x \geq 1\}}$

$C = \sup_{x \geq 1} \frac{\tilde{f}(x)}{g(x)} = \sup_{x \geq 1} \frac{e^{-\frac{x^2}{2}+x-1}}{\sqrt{2\pi}} = \frac{1}{\sqrt{2\pi e}}$

1. Generate $Y = 1 + \mathrm{Exp}(1)$
2. Generate $U \sim \mathcal{U}(0,1)$
3. If $U \leq \frac{\tilde{f}(Y)}{Cg(Y)} = e^{-Y^2/2+Y-1/2}$ set $X = Y$, otherwise return to step 1.

Acceptance rate ≈ 0.66



Acceptance-Rejection with squeezing

Suppose $\tilde{f}(x)$ is costly to evaluate. In the AR algorithm can we reduce the number of evaluations of $\tilde{f}$?

Simple idea: if we have a lower bound $\hat{g}(x) \leq \tilde{f}(x)$, $\forall x$, which is cheap to evaluate, and in the AR step $U \leq \frac{\hat{g}(Y)}{Cg(Y)}$, then we can accept Y without evaluating $\tilde{f}$.

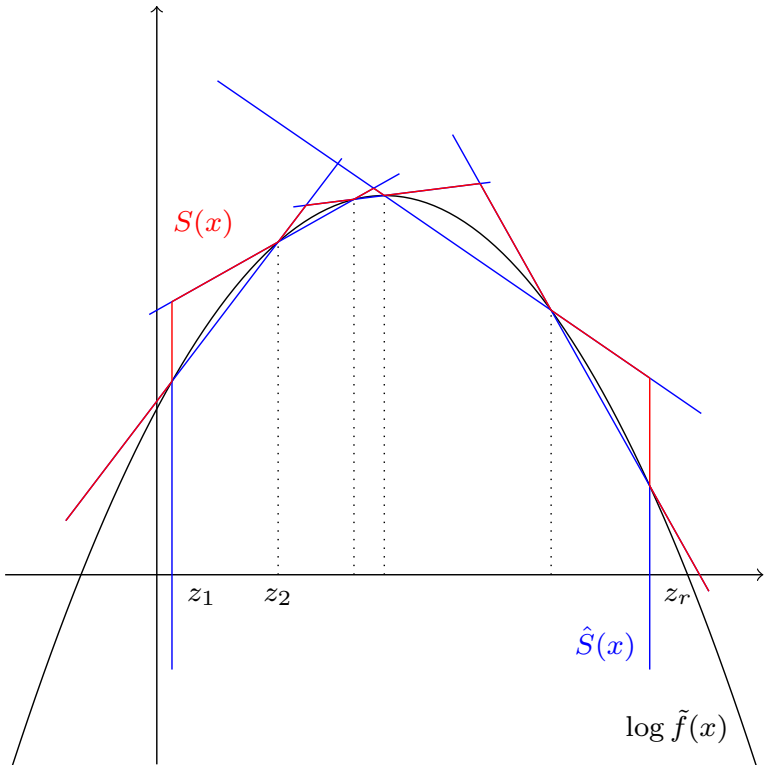
Algorithm: AR algorithm with squeezing.

---

1. Generate $Y \sim g$
2. Generate $U \sim \mathcal{U}([0,1])$
3. If $U \leq \dfrac{\hat{g}(Y)}{Cg(Y)}$ set $X = Y$, otherwise, evaluate $\tilde{f}(Y)$
4. If $U \leq \frac{\tilde{f}(Y)}{Cg(Y)}$ set $X = Y$
5. else reject Y and go back to 1

Adaptive AR for log-concave densities

Let $\tilde{f}$ be log-concave and $Z_r = \{z_1, \ldots, z_r\}$ be an initial set of points. We can bound easily $\log \tilde{f}$ from above and below by piecewise linear functions (see Figure): $e^{\hat{s}(x)} \leq \tilde{f}(x) \leq e^{s(x)}$



- use $g(x) = e^{s(x)} / \int e^{s(x)} dx$ as a proposal distribution in AR
- use $\hat{g}(x) = e^{\hat{s}(x)}$ as a lower bound in the squeezing algorithm.
- Once a new point z_{r+1} has been created, it can be added to Z_r and the functions $s(x)$ and $\hat{s}(x)$ recomputed.

Sampling from Normal distribution – Box-Muller method

Standard distributions often have specialized generators. We mention one for the normal distribution (not the most efficient, though).

It generates couples (X, Y) of independent standard Normal rvs and relies on a transformation in polar coordinates. Given $X, Y \overset{iid}{\sim} N(0,1)$

- $\rho^2 = X^2 + Y^2 \sim \xi_2^2 = Exp(\frac{1}{2})$
- by radial symmetry, the distribution of $(X, Y)|X^2 + Y^2$ is uniform in $[0, 2\pi]$ and independent of ρ.

Algorithm: Box-Muller method.

---

1. Generate $U \sim \mathcal{U}(0,1)$ and set $\rho = \sqrt{-2 \log U}$
2. Generate $V \sim \mathcal{U}(0,1)$ and set $\Theta = 2\pi V$
3. Set $X = \rho \cos\Theta$, $Y = \rho \sin\Theta$.

Multivariate random variable generation

- $\boldsymbol{X} = (X_1, \ldots, X_n)^\top \in \mathbb{R}^n$: vector of rvs.
- joint cdf: $F(\boldsymbol{z}) = F(z_1, \ldots, z_n) = \mathbb{P}(X_1 \leq z_1, \ldots, X_n \leq z_n)$
- joint pdf (if it exists): $f(\boldsymbol{x}) = f(x_1, \ldots, x_n) : \mathbb{R}^n \to \mathbb{R}_+$ such that

$$F(z_1, \ldots, z_n) = \int_{-\infty}^{z_1} \cdots \int_{-\infty}^{z_n} f(x_1, \ldots, x_n)\, dx_1 \ldots dx_n.$$

The inverse transform method is not applicable in the multivariate case (F is not invertible)

The Acceptance-Rejection method generalizes straightforwardly to the multivariate case. However, it might be difficult to find a "good" density g which gives acceptable acceptance rates.

Simple cases

Independent components

$\boldsymbol{X} = (X_1, \ldots, X_n)$ has independent components and cdf $F(\boldsymbol{z}) = F_1(z_1) \ldots F_n(z_n)$.

Then one generates $X_i \sim F_i,\ i = 1, \ldots, n$ independently.

Example: to generate $\boldsymbol{X} \sim \mathcal{U}([0,1]^n)$, we can generate $X_i \sim \mathcal{U}([0,1]),\ i = 1, \ldots, n$ independently.

Generating from conditional distributions

Consider $\boldsymbol{X}$ with dependent components and factorize the pdf as

$$f_{\boldsymbol{X}}(\boldsymbol{z}) = f_{X_1}(z_1) f_{X_2 \mid X_1}(z_2 \mid z_1) \ldots f_{X_n \mid X_{1:n-1}}(z_n \mid z_{1:n-1}).$$

Assuming each conditional density $f_{X_i \mid X_{1:i-1}}(z_i \mid z_{1:i-1}),\ i = 1, \ldots, n$ known and easy to generate:

Algorithm: Conditional distribution generation.

1 Generate $X_1 \sim f_{X_1}(z)$
2 For $i = 2, \ldots, n$,
3 Generate $X_i \sim f_{X_i \mid X_1, \ldots, X_{i-1}}(z \mid X_1, \ldots, X_{i-1})$

Example - generating order statistics

Let $\boldsymbol{X} = (X_1, \ldots, X_n) \sim \mathcal{U}((0,1)^n)$ and denote by $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$ the ordered sample (ordered statistics).

The "sort" operation may be costly for a large sample. Can we generate the order statistics without sort?

Yes, by conditioning. It can be shown (exercise) that

- $X_{(n)} = \max_{i=1,\ldots,n} X_i$ has distribution $F_{X_{(n)}}(z) = z^n$
- $X_{(j)} \mid X_{(j+1)}, \ldots, X_{(n)}$ has (conditional) distribution

$$F_{X_{(j)} \mid X_{(j+1)},\ldots,X_{(n)}}(z \mid x_{j+1}, \ldots, x_n) = \mathbb{P}\left(X_{(j)} \leq z \mid X_{(k)} = x_k, k > j\right)$$

$$= \mathbb{P}\left(X_{(j)} \leq z \mid X_{(j+1)} = x_{j+1}\right) = \left(\frac{z}{x_{j+1}}\right)^j \mathbb{1}_{\{z \leq x_{j+1}\}}$$

Algorithm: Generation of order statistics by conditioning.

1. Generate $U_n \sim \mathcal{U}(0,1)$ and set $X_{(n)} = (U_n)^{\frac{1}{n}}$
2. For $j = n-1, \ldots, 1$,
3. Generate $U_j \sim \mathcal{U}(0,1)$ and set $X_{(j)} = X_{(j+1)}(U_j)^{\frac{1}{j}}$

Multivariate Gaussian random variable generation

Let $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$, mean $\boldsymbol{\mu} \in \mathbb{R}^n$, covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ (spd).

$$\text{joint pdf} \quad f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \quad \boldsymbol{x} \in \mathbb{R}^n$$

Common way to generate $\boldsymbol{X}$: factorize covariance matrix $\Sigma = AA^\top$

Algorithm: Multivariate Gaussian generator

Given: $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\Sigma = AA^\top \in \mathbb{R}^{n \times n}$ (spd)

1. Generate $\boldsymbol{Y} \sim N(\boldsymbol{0}, I_{n \times n})$ (i.e. $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$, $Y_i \overset{\text{iid}}{\sim} N(0,1)$)
2. Compute $\boldsymbol{X} = \boldsymbol{\mu} + A\boldsymbol{Y}$

Indeed

- $\boldsymbol{X}$ is Gaussain (affine transformation of Gaussians)
- $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{\mu}$
- $\mathrm{Cov}[\boldsymbol{X}] = \mathbb{E}\left[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^\top\right] = \mathbb{E}\left[A\boldsymbol{Y}\boldsymbol{Y}^\top A^\top\right] = A\mathbb{E}\left[\boldsymbol{Y}\boldsymbol{Y}^\top\right]A^\top = \Sigma$

Multivariate Gaussian random variable generation

How to factorize Σ

- spectral decomposition of Σ: V matrix of eigenvectors; $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$ matrix of eigenvalues

$$\Sigma = VDV^\top = (VD^{\frac{1}{2}})(VD^{\frac{1}{2}})^\top = AA^\top, \qquad A = VD^{\frac{1}{2}}$$

  Costly for large matrices; works also for singular (or nearly singular) matrices.

- Cholesky factorization: $\Sigma = AA^\top$ with A lower triangular

  Cheaper than the spectral decomposition but works only for strictly positive definite matrices

- A more robust version for nearly singular or singular matrices is the *pivoted* Cholesky factorization

Multivariate Gaussian random variable generation

Sometime one has access to the precision matrix Σ^{-1} instead of the covariance matrix Σ.

In such case, one can factorize the precision matrix $\Sigma^{-1} = AA^\top$

Algorithm: Multivariate Gaussian generator from precision matrix

---

1 Compute the Cholesky factorisation $\Sigma^{-1} = LL^T$
2 Generate $\boldsymbol{Z} \sim N(\mathbf{0}, I)$ // n independent standard normals
3 Solve the linear system $L^\top \boldsymbol{Y} = \boldsymbol{Z}$ // upper triangular
4 Output $\boldsymbol{X} = \boldsymbol{\mu} + \boldsymbol{Y}$

Indeed: $\mathrm{Cov}[\boldsymbol{X}] = \mathbb{E}\left[\boldsymbol{Y}\boldsymbol{Y}^\top\right] = \mathbb{E}\left[L^{-\top}\boldsymbol{Z}\boldsymbol{Z}^\top L^{-1}\right] = L^{-\top}L^{-1} = \Sigma$

Generating conditional Gaussian random variables

Let $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$. Split $\boldsymbol{X} = (Y_1, \dots, Y_{n-k}, Z_1, \dots, Z_k) = (\boldsymbol{Y}, \boldsymbol{Z})$. Then

$$\mathbb{E}[\boldsymbol{X}] = \begin{pmatrix} \boldsymbol{\mu_Y} \\ \boldsymbol{\mu_Z} \end{pmatrix}, \qquad \Sigma = \begin{pmatrix} \Sigma_{\boldsymbol{YY}} & \Sigma_{\boldsymbol{YZ}} \\ \Sigma_{\boldsymbol{YZ}}^\top & \Sigma_{\boldsymbol{ZZ}} \end{pmatrix}$$

The conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{Z} = \boldsymbol{z}$ is again a multivariate Gaussian $N(\boldsymbol{\mu}_{\boldsymbol{Y} \mid \boldsymbol{Z}}, \Sigma_{\boldsymbol{Y} \mid \boldsymbol{Z}})$ with

$$\boldsymbol{\mu}_{\boldsymbol{Y} \mid \boldsymbol{Z}} = \boldsymbol{\mu_Y} + \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} (\boldsymbol{z} - \boldsymbol{\mu_z}) \tag{1}$$

$$\Sigma_{\boldsymbol{Y} \mid \boldsymbol{Z}} = \Sigma_{\boldsymbol{YY}} - \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} \Sigma_{\boldsymbol{ZY}}. \tag{2}$$

To generate $\boldsymbol{Y} \mid \boldsymbol{Z} = \boldsymbol{z}$ we can factorize the conditional covariance $\Sigma_{\boldsymbol{Y} \mid \boldsymbol{Z}}$ and use the previous algorithm.
However, in some cases, factorizing Σ (or Σ^{-1}) is cheap, (e.g. stationary random fields, markov random fields, etc.), whereas factorizing $\Sigma_{\boldsymbol{Y} \mid \boldsymbol{Z}}$ is costly (full $O(n^3)$ cost)

Generating conditinal Gaussian random variables

An alternative way to generate a conditional Gaussian random variable is given by the following algorithm

---

Algorithm: Generation from conditional Gaussian distribution - I

Given: μ, Σ and $\boldsymbol{z} \in \mathbb{R}^k$

1 Generate $\boldsymbol{X} \sim N(\mu, \Sigma)$ `// (unconditional distr.)`

2 Set $\mathcal{Y} = (X_1, \ldots, X_{n-k})$ and $\mathcal{Z} = (X_{n-k+1}, \ldots, X_n)$

3 Output $\boldsymbol{Y} = \mathcal{Y} + \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}(\boldsymbol{z} - \mathcal{Z})$

We can easily verify that $\boldsymbol{Y}$ has the correct distribution. Indeed,

$$\mathbb{E}[\boldsymbol{Y}] = \mathbb{E}[\mathcal{Y}] + \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}(\boldsymbol{z} - \mathbb{E}[\mathcal{Z}]) = \mu_{\boldsymbol{Y}} + \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}(\boldsymbol{z} - \mu_{\boldsymbol{Z}}).$$

Set $\boldsymbol{Y}' = \boldsymbol{Y} - \mathbb{E}[\boldsymbol{Y}] = \mathcal{Y}' - \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}\mathcal{Z}'$. Then

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{Y}) = \mathbb{E}\left[\boldsymbol{Y}'\boldsymbol{Y}'^{\top}\right] &= \mathbb{E}\left[\mathcal{Y}'\mathcal{Y}'^{\top}\right] - \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}\mathbb{E}\left[\mathcal{Z}'\mathcal{Y}'^{\top}\right] \\ &\quad - \mathbb{E}\left[\mathcal{Y}'\mathcal{Z}'^{T}\right]\Sigma_{\boldsymbol{ZZ}}^{-\top}\Sigma_{\boldsymbol{YZ}}^{\top} + \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}\mathbb{E}\left[\mathcal{Z}'\mathcal{Z}'^{T}\right]\Sigma_{\boldsymbol{ZZ}}^{-\top}\Sigma_{\boldsymbol{YZ}}^{\top} \\ &= \Sigma_{\boldsymbol{YY}} - \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}\Sigma_{\boldsymbol{ZY}}\end{aligned}$$

Conditioning on arbitrary linear observations

In the previous setting we assumed to observe some components of $\boldsymbol{X}$. We can generalize the argument by assuming that we observe (possibly with noise) some linear combinations of $\boldsymbol{X}$:

$$\boldsymbol{Z} = H\boldsymbol{X} + \boldsymbol{\eta}, \qquad H \in \mathbb{R}^{k\times n}, \quad \boldsymbol{\eta} \sim N(0,\Gamma) \text{ independent of } \boldsymbol{X}$$

Defining $\tilde{\boldsymbol{X}} = (\boldsymbol{X}, \boldsymbol{Z}) \in \mathbb{R}^{n+k}$ which has

$$\mathbb{E}\left[\tilde{\boldsymbol{X}}\right] = \begin{pmatrix} \mu \\ H\mu \end{pmatrix}, \qquad \mathrm{Cov}(\tilde{\boldsymbol{X}}) = \begin{pmatrix} \Sigma & \Sigma H^\top \\ H\Sigma & S \end{pmatrix}, \quad S = H\Sigma H^\top + \Gamma$$

we obtain $\boldsymbol{X} \mid \boldsymbol{Z} = \boldsymbol{z} \sim N(\mu_{\boldsymbol{X}\mid \boldsymbol{Z}=\boldsymbol{z}}, \Sigma_{\boldsymbol{X}\mid \boldsymbol{Z}})$

$$\mu_{\boldsymbol{X}\mid \boldsymbol{Z}=\boldsymbol{z}} = \mu + \Sigma H^\top S^{-1}(\boldsymbol{z} - H\mu), \quad \Sigma_{\boldsymbol{X}\mid \boldsymbol{Z}} = \Sigma - \Sigma H^\top S^{-1} H\Sigma.$$

Algorithm: Generation from conditional Gaussian distribution - II

Given: μ, Σ and $\boldsymbol{z} \in \mathbb{R}^k$

1. Generate $\boldsymbol{\mathcal{X}} \sim N(\mu, \Sigma)$
2. generate $\boldsymbol{\eta} \sim N(0, \Gamma)$
3. compute perturbed innovation $\tilde{\mathbf{d}} = \boldsymbol{z} - H\boldsymbol{\mathcal{X}} + \boldsymbol{\eta}$
4. Output $\boldsymbol{X} = \boldsymbol{\mathcal{X}} + K\tilde{\boldsymbol{d}}$
