Université de Bourgogne 2021-2022

Étude d'un mélange de deux lois de probabilité

Mise en contexte. Dans cette étude on souhaite décrire la loi de probabilité de la variable aléatoire Z définie de la façon suivante,

$$Z = \lambda_1 e^{\beta_1 X_1} + \lambda_2 e^{\beta_2 X_2},$$

où, (X_1, X_2) est un couple de variables aléatoires réelles, λ_1, λ_2, β_1 et β_2 sont des réels positifs.
En particulier on évaluera la probabilité $p = \mathbb{P}(Z > t)$ à l'aide de diverses méthodes.

1. Méthode d'échantillonnage moyen de Monte-Carlo

On suppose ici que, $X = (X_1, X_2)$ est un vecteur gaussien centré tel que,

$$\mathbb{V}(X_1) = \mathbb{V}(X_2) = 1 \text{ et, } \mathrm{Cov}(X_1,X_2) = \rho.$$

avec $|\rho| \leq 1$.

(a) But : Simuler une variable aléatoire selon la loi de Z.

Notons Σ la matrice de covariance du vecteur gaussien X,

$$\Sigma_X = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

Pour simuler un vecteur aléatoire gaussien, on peut se réferer à la proposition suivante,

Proposition 23. ([1] p.13) Soient Y un vecteur gaussien à valeurs dans $\mathbb{R}^n$, m sa moyenne et Σ_Y sa matrice de covariance, A une matrice $p \times n$ et z un vecteur de $\mathbb{R}^p$.
On pose $X = AY + z$. Alors X est un vecteur gaussien et,

$$\mathbb{E}[X] = z + Am, \quad \Sigma_X = A\Sigma_Y A^*$$

Σ_X désignant la matrice de covariance de X.
A^* désignant la transposée de A.

On sait simuler des variables gaussiennes indépendantes centrées-réduites via la proposition 1 du cours ([2] p.13) en utilisant deux variables aléatoires indépendantes de loi uniforme sur $[0, 1]$, ou directement via la fonction python dédiée :

```
numpy.random.normal(mu, sigma, n)
#mu, sigma : parametres de la loi
#n : nombre de variables selon cette loi a simuler
```

Ainsi, on peut poser $Y = (Y_1, Y_2)$ où Y_1 et Y_2 sont deux v.a gaussiennes réelles, indépendantes, équidistribuées de loi $\mathcal{N}(0,1)$. Y est un vecteur gaussien et on a alors $\Sigma_Y = Id_2$.

On définit X par :

$$X = m + AY = AY$$

Puisque $m = 0_{\mathbb{R}^2}$.

Pour que X soit un vecteur gaussien de matrice de covariance Σ_X, d'après la proposition 23, on cherche A telle que,

$$AA^* = \Sigma_X = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

L'unique matrice, dite racine carrée de Σ_X, triangulaire inférieure dont les coefficients diagonaux sont positifs est donnée par [1],

$$A = \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix}$$

Alors, pour simuler une variable aléatoire selon la loi de Z, on peut procéder de la façon suivante,

```
import numpy as np
import numpy.random as npr
rho = 1/2 #a faire varier
Y = npr.normal(0,1,2)
A = [ [1, 0], [rho, np.sqrt(1-rho**2)] ]
X = A@Y
l1 = 1; l2 = 1; #a faire varier
b1 = 1; b2 = 1; #a faire varier
Z = l1*np.exp(b1*X[0]) + l2*np.exp(b2*X[1])
```

Sous python, pour simuler le vecteur gaussien X, on peut aussi directement utiliser la fonction :

```
numpy.random.multivariate_normal(mu, Sigma)
#mu : vecteur des moyennes
#Sigma : matrice de covariance
```

(b) Application de la méthode d'échantillonnage moyen de Monte-Carlo.

On souhaite approcher la valeur de p,

$$p = \mathbb{P}(Z > t) = \iint \mathbb{1}_{\{Z>t\}} f_Z(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

La méthode d'échantillonnage moyen consiste à représenter l'intégrale sous forme d'une espérance mathématique.

En particulier,

$$p = \mathbb{E}[\mathbb{1}_{\{Z>t\}}]$$

Cette méthode repose sur la loi forte des grands nombres, dans le cas de l'estimation de p on peut l'appliquer car, $\mathbb{E}(|\mathbb{1}_{\{Z>t\}}|^2) < \infty$.

1. Via la factorisation de Cholesky

Soit $(Z_i)_{1\leq i\leq N}$ une suite de variables aléatoires indépendantes de même loi que Z. L'estimation de p se fait alors via l'estimateur sans biais, ([2] p.13)

$$\theta_N = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{Z_i>t\}}$$

Pour simuler numériquement cet estimateur, on commence par simuler un N-échantillon de variables $(Z_1, Z_2, \ldots, Z_N)$ selon la loi de Z.

On introduira alors une variable, appelée `cmpt`, ayant pour rôle de sommer les valeurs de l'indicatrice de $\{Z > t\}$, celle-ci sera donc incrémentée de 1 à chaque Z_k simulé tel que $Z_k > t$, pour $k = 1, \ldots, N$.

Ainsi, notre estimateur prendra pour valeur le rapport entre la variable de comptage et N.

En reprenant la simulation précédente, sous python cela nous donne,

```
N = 1000 #taille de l'echantillon
t = 2 #a faire varier
cmpt = 0 #variable de comptage
Zn = []
K = [ [1, rho], [rho, 1] ] #matrice de covariance de X
for i in range (N) :
    #Simulation d'un vecteur gaussien X (N fois)
    X = npr.multivariate_normal([0,0], K)
    #Simulation d'un N-ech. selon la loi de Z
    Zn.append(l1*np.exp(b1*X[0]) + l2*np.exp(b2*X[1]))
    if (Zn[i] > t ) :
        cmpt = cmpt + 1   #Incrementation de la variable de comptage
theta = cmpt/N #estimateur empirique
```

On s'intéresse maintenant à la précision de cette méthode, celle-ci est donnée par le théorème "central limit". D'après ce théorème on a,

$$\sqrt{N}\left(\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\{Z_i>t\}} - \mathbb{E}[\mathbb{1}_{\{Z>t\}}]\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}(0, \mathbb{V}(\mathbb{1}_{\{Z>t\}}))$$

Puisque $\mathbb{1}_{\{Z_i>t\}}$ suit une loi de Bernoulli de paramètre $p = \mathbb{P}(Z > t)$, on a,

$$\mathbb{V}(\mathbb{1}_{\{Z>t\}}) = p(1-p)$$

Donc,

$$\sqrt{N}(\theta_N - p) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}(0, p(1-p))$$

Ici on peut directement appliquer le résultat théorique et calculer l'écart-type de la méthode comme suit.

```
var = theta*(1-theta)
sd = np.sqrt(var)
```

Numériquement on peut aussi directement obtenir l'écart-type avec la fonction suivante,

```
numpy.std(a) #a : tableau des valeurs
```

Ce qui nous sera utile pour les méthodes qui vont suivre.

2. Nombre de tirages moyens dans une méthode de Monte-Carlo standard

Comme explicité précedemment, la précision des méthodes de Monte-Carlo repose sur le théorème "central limit", en particulier, on peut construire un intervalle de confiance de niveau $1-\alpha$ comme suit.

D'après le théorème "central limit",

$$\lim_{n\to\infty} \mathbb{P}\left(z_{\frac{\alpha}{2}} \leq \sqrt{N}\frac{\theta_N - p}{\sqrt{p(1-p)}} \leq z_{1-\frac{\alpha}{2}} \right) = 1-\alpha$$

où, $z_{\frac{\alpha}{2}}$ (resp. $z_{1-\frac{\alpha}{2}}$) est le quantile d'ordre $\frac{\alpha}{2}$ (resp. $1-\frac{\alpha}{2}$) de la loi normale $\mathcal{N}(0,1)$.

Puisque $z_{\frac{\alpha}{2}} = -z_{1-\frac{\alpha}{2}}$, ceci nous donne,

$$\lim_{n\to\infty} \mathbb{P}\left(|\theta_N - p| \leq z_{1-\frac{\alpha}{2}}\sqrt{\frac{p(1-p)}{N}} \right) = 1-\alpha$$

Par la loi forte des grands nombre θ_N est un estimateur convergent presque-sûrement vers p. En utilisant alors le théorème de l'application continue, $x \mapsto \sqrt{x}$ continue, on a,

$$\sqrt{\theta_N(1-\theta_N)} \quad \xrightarrow{\mathbb{P}} \quad \sqrt{p(1-p)}$$

Puis avec le théorème de Slutsky, on en déduit un intervalle de confiance de niveau $1-\alpha$,

$$I_{1-\alpha}(p) = \left[\theta_N - \sqrt{\frac{\theta_N(1-\theta_N)}{N}}; \theta_N + \sqrt{\frac{\theta_N(1-\theta_N)}{N}} \right]$$

On cherche alors un ordre de grandeur de N, le nombre de tirages à effectuer, pour pouvoir affirmer avec une une confiance $1-\alpha$ que $p \in \left[\frac{1}{2}\times 10^{-7}; \frac{3}{2}\times 10^{-7}\right]$.

Cela se traduit par une erreur inférieure à $\frac{1}{2}\times 10^{-7}$.

C'est-à-dire qu'on cherche N tel que,

$$z_{1-\frac{\alpha}{2}}\sqrt{\frac{\theta_N(1-\theta_N)}{N}} < \frac{1}{2}\times 10^{-7}$$

Il faut alors,

$$\sqrt{N} > z_{1-\frac{\alpha}{2}}\frac{2\sqrt{\theta_N(1-\theta_N)}}{10^{-7}} \quad \Leftrightarrow \quad N > z^2_{1-\frac{\alpha}{2}}\frac{4\,\theta_N(1-\theta_N)}{10^{-14}}$$

Avec une confiance de $1-\alpha = 0.95$, on a $z_{1-\frac{\alpha}{2}} = 1.96$. On suppose aussi que t est tel que p est de l'ordre de 10^{-7} donc $p(1-p)$ est de l'ordre de 10^{-7} aussi. On obtient,

$$N > 1.96^2 \times \frac{4\times 10^{-7}}{10^{-14}} = 3.8416 \times \frac{4}{10^{-7}} > 1.5\times 10^8 \text{ tirages.}$$

Ainsi, un ordre de grandeur du nombre de tirages à effectuer, pour pouvoir estimer p avec une erreur inférieure à $\frac{1}{2}\times 10^{-7}$, à l'aide d'une méthode de Monte-Carlo standard, est de 10^8 tirages.

3. Technique d'échantillonnage préférentiel

La méthode utilisée précédemment, dans le cas de l'estimation d'une probabilité p très petite, savère être non-adaptée. Pour corriger cela, et réduire la variance tout en ayant un nombre de simulations raisonnable, on applique une méthode d'échantillonnage préferentiel.

On cherche alors à réécrire p sous la forme d'une espérance, mais cette fois-ci à l'aide d'une autre densité.

On note,

$$g(X_1, X_2) = \lambda_1 e^{\beta_1 X_1} + \lambda_2 e^{\beta_2 X_2}$$

Dans cette partie on suppose que X_1 et X_2 sont deux gaussiennes centrées réduites, et indépendantes. On notera f_{X_1} (resp. f_{X_2}) la densité de X_1 (resp. X_2).

On a alors,

$$\begin{aligned} p = \mathbb{P}(g(X_1, X_2) > t) &= \mathbb{E}[\mathbb{1}_{\{g(X_1,X_2)>t\}}] \\ &= \iint_{\mathbb{R}^2} \mathbb{1}_{\{g(X_1,X_2)>t\}} f_{X_1}(x_1) f_{X_2}(x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2 \end{aligned}$$

Puisque X_1 et X_2 sont indépendantes la loi jointe est égale au produit des lois marginales.

D'autre part, on sait que si $X \sim \mathcal{N}(0,1)$ alors, $X + m \sim \mathcal{N}(m, 1)$, pour un nombre réel m.
De plus, puisque X_1, X_2 sont indépendantes, $X_1 + m$ et $X_2 + m$ le sont aussi. On note alors, f_{X_1+m} et f_{X_2+m} les densités respectives de $X_1 + m$ et $X_2 + m$, et on a,

$$\begin{aligned} p &= \iint_{\mathbb{R}^2} \mathbb{1}_{\{g(X_1,X_2)>t\}} \frac{f_{X_1}(x_1) f_{X_2}(x_2)}{f_{X_1+m}(x_1) f_{X_2+m}(x_2)} f_{X_1+m}(x_1) f_{X_2+m}(x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2 \\ &= \mathbb{E}\left[\mathbb{1}_{\{g(X_1+m,X_2+m)>t\}} \frac{f_{X_1}(X_1+m) f_{X_2}(X_2+m)}{f_{X_1+m}(X_1+m) f_{X_2+m}(X_2+m)}\right] \\ &= \mathbb{E}\left[\mathbb{1}_{\{g(X_1+m,X_2+m)>t\}} \frac{2\pi e^{-\frac{1}{2}(X_1+m)^2} e^{-\frac{1}{2}(X_2+m)^2}}{2\pi e^{-\frac{1}{2}(X_1+m-m)^2} e^{-\frac{1}{2}(X_2+m-m)^2}}\right] \\ &= \mathbb{E}\left[\mathbb{1}_{\{g(X_1+m,X_2+m)>t\}} e^{-\frac{1}{2}(X_1+m)^2 - \frac{1}{2}(X_2+m)^2 + \frac{1}{2}(X_1^2+X_2^2)}\right] \\ &= \mathbb{E}\left[\mathbb{1}_{\{g(X_1+m,X_2+m)>t\}} e^{-\frac{1}{2}(X_1^2+2X_1m+m^2+X_2^2+2X_2m+m^2) + \frac{1}{2}(X_1^2+X_2^2)}\right] \\ &= \mathbb{E}\left[\mathbb{1}_{\{\lambda_1 e^{\beta_1(X_1+m)} + \lambda_2 e^{\beta_2(X_2+m)} > t\}} e^{-(X_1+X_2)m - m^2}\right] \end{aligned}$$

Ainsi, p peut s'écrire sous la forme :

$$\mathbb{E}\left[\phi(X_1, X_2)\mathbb{1}_{\{\lambda_1 e^{\beta_1(X_1+m)} + \lambda_2 e^{\beta_2(X_2+m)} > t\}}\right]$$

où, $\phi(X_1, X_2) = e^{-(X_1+X_2)m - m^2}$

On souhaite maintenant majorer la probabilité suivante,

$$\begin{aligned}
\mathbb{P}(g(X_{1+m}, X_{2+m}) > t) &= \mathbb{P}(\lambda_1 e^{\beta_1(X_1+m)} + \lambda_2 e^{\beta_2(X_2+m)} > t) && (1)\\
&\geq \mathbb{P}\left(\left\{\lambda_1 e^{\beta_1(X_1+m)} > \frac{t}{2}\right\}, \left\{\lambda_2 e^{\beta_2(X_2+m)} > \frac{t}{2}\right\}\right) && (2)\\
&= \mathbb{P}\left(\lambda_1 e^{\beta_1(X_1+m)} > \frac{t}{2}\right) \times \mathbb{P}\left(\lambda_2 e^{\beta_2(X_2+m)} > \frac{t}{2}\right) && (3)
\end{aligned}$$

En effet, (1) est majoré par (2) puisque l'évenement $\{\lambda_1 e^{\beta_1(X_1+m)} > \frac{t}{2}, \lambda_1 e^{\beta_1(X_1+m)} > \frac{t}{2}\}$ est inclus dans $\{g(X_1, X_2) > t\}$, c'est un cas particulier de la réalisation.

Par indépendance des variables aléatoires X_1 et X_2 on a (2) qui est égal à (3), car l'intersection est égale au produit des probabilités.

On cherche alors un m vérifiant,

$$\mathbb{P}(\lambda_1 e^{\beta_1(X_1+m)} + \lambda_2 e^{\beta_2(X_2+m)} > t) \geq \frac{1}{4}$$

On va donc résoudre le système d'inéquations suivant,

$$\begin{cases} \mathbb{P}\left(\lambda_1 e^{\beta_1(X_1+m)} > \frac{t}{2}\right) \geq \frac{1}{2} \\ \mathbb{P}\left(\lambda_2 e^{\beta_2(X_2+m)} > \frac{t}{2}\right) \geq \frac{1}{2} \end{cases} \quad \Leftrightarrow \quad \begin{cases} \mathbb{P}\left(X_1 > \frac{\ln(t) - \ln(2\lambda_1)}{\beta_1} - m\right) \geq \frac{1}{2} \\ \mathbb{P}\left(X_2 > \frac{\ln(t) - \ln(2\lambda_2)}{\beta_2} - m\right) \geq \frac{1}{2} \end{cases}$$

Rappelons que si $X \sim \mathcal{N}(m, 1)$ alors, $\mathbb{P}(X > m) = \frac{1}{2}$.

Ainsi, puisque $X_1 \sim \mathcal{N}(0, 1)$ et $X_2 \sim \mathcal{N}(0, 1)$, on veut m tel que,

$$\begin{cases} \frac{\ln(t) - \ln(2\lambda_1)}{\beta_1} - m \leq 0 \\ \frac{\ln(t) - \ln(2\lambda_2)}{\beta_2} - m \leq 0 \end{cases} \quad \Leftrightarrow \quad \begin{cases} m \geq \frac{\ln(t) - \ln(2\lambda_1)}{\beta_1} \\ m \geq \frac{\ln(t) - \ln(2\lambda_2)}{\beta_2} \end{cases}$$

On peut proposer le choix suivant pour m,

$$m = \frac{\ln(t) - \ln(2\lambda_1)}{\beta_1} + \frac{\ln(t) - \ln(2\lambda_2)}{\beta_2}$$

On peut alors à nouveau appliquer une méthode d'échantillonnage moyen pour estimer p.

L'estimateur sans biais est donné par,

$$\hat{\theta}_N = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{\lambda_1 e^{\beta_1(X_1^i+m)} + \lambda_2 e^{\beta_2(X_2^i+m)} > t\}} e^{-(X_1^i+X_2^i)m - m^2}$$

où, $(X_1^i)_{1\leq i\leq N}$, (resp. $(X_2^i)_{1\leq i\leq N}$) est une suite de variables aléatoires indépendantes de densité f_{X_1} (resp. f_{X_2}).

Pour simuler ce processus, en prenant soin de définir en amont les paramètres, on simule N variables de loi normale centrée réduite. On gardera alors en mémoire les valeurs sommées pour pouvoir calculer numériquement la variance.

On définira aussi la fonction suivante, pour alléger le code,

```
m = (np.log(t)-np.log(2*l1))/b1 + (np.log(t)-np.log(2*l2))/b2
def f(x,l,b) :
    return l*np.exp(b*(x+m)) #m fixe
```

Alors le processus sera implémenté comme suit, ‘

```
Xn = []
for i in range (N) :
    #Simulation de 2 v.a de loi normale centree reduite
    X = npr.normal(0,1,2)
    if ( (f(X[0],l1,b1) + f(X[1],l2,b2)) > t) :
        Xn.append(np.exp(-(X[0]+X[1])*m-m**2))
    else :
        Xn.append(0)
theta = np.mean(Xn) #estimateur empirique
```

On obtient l'écart-type en utilisant la fonction python dédiée,

```
sd = np.std(Xn) #ecart-type empirique
```

La réduction de la variance dépend ici de m, c'est pourquoi on a cherché m tel que, $\mathbb{P}(\lambda_1 e^{\beta_1(X_1+m)} + \lambda_2 e^{\beta_2(X_2+m)} > t) \geq \frac{1}{4}$. Cette condition assure que la variance de cette méthode sera plus petite que la variance de $\mathbb{1}_{\{Z_i>t\}}$, (où $(Z_i)_{1\leq i\leq N}$ i.i.d, selon la loi de Z).

4. Propriétés de l'espérance et de la variance

Soit (X, Y) deux variables aléatoires indépendantes de lois données par les densités p_X et p_Y. Montrons que, si f est une fonction bornée, alors,

$$\mathbb{E}[f(X,Y)] = \mathbb{E}[h(Y)]$$

avec $h(y) = \mathbb{E}[f(X, y)]$

Notons que la loi jointe est le produit des lois marginales puisque les variables X et Y sont indépendantes. On peut donc écrire, via le lemme de transfert,

$$\mathbb{E}\left[f(X,Y)\right] = \iint_{\mathbb{R}^2} f(x,y)p_X(x)p_Y(y)\,\mathrm{d}x\,\mathrm{d}y$$

En utilisant Fubini-Tonelli, puisque les densités sont positives et que f est bornée, on a,

$$\begin{aligned}\mathbb{E}\left[f(X,Y)\right] &= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} f(x,y)p_X(x)\,\mathrm{d}x\right) p_Y(y)\mathrm{d}y \\ &= \int_{\mathbb{R}} h(y)p_Y(y)\mathrm{d}y = \mathbb{E}[h(Y)]\end{aligned}$$

En effet, $h(y) = \int_{\mathbb{R}} f(x,y)p_X(x)\,\mathrm{d}x$, puisque y est fixé, on intègre seulement contre la loi de X.

De la même façon avec f^2, en appliquant le lemme de transfert on peut voir que,

$$\mathbb{E}\left[f^2(X,Y)\right] = \iint_{\mathbb{R}^2} f^2(x,y)p_X(x)p_Y(y)\,\mathrm{d}x\,\mathrm{d}y$$

Par les mêmes arguments [2] que précedemment, on obtient,

$$\begin{aligned}\mathbb{E}\left[f^2(X,Y)\right] &= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} f^2(x,y)p_X(x)\,\mathrm{d}x\right) p_Y(y)\mathrm{d}y \\ &= \int_{\mathbb{R}} g(y)p_Y(y)\mathrm{d}y\end{aligned}$$

où, $g(y) = \int_{\mathbb{R}} f^2(x,y)p_X(x)\,\mathrm{d}x$

On peut alors en déduire la propriété suivante de la variance,

$$\mathbb{V}(h(Y)) \leq \mathbb{V}(f(X,Y))$$

En décomposant la variance on a naturellement,

$$\begin{aligned}\mathbb{V}(f(X,Y)) &= \mathbb{E}[f^2(X,Y)] - \mathbb{E}[f(X,Y)]^2 &(4)\\ &= \mathbb{E}[f^2(X,Y)] - \mathbb{E}[h(Y)]^2 &(5)\\ &= \mathbb{E}[f^2(X,Y)] - \Big(\mathbb{E}[h^2(Y)] - \mathbb{V}(h(Y))\Big) &(6)\end{aligned}$$

De ce qui précède, $\mathbb{E}[h(Y)] = \mathbb{E}[f(X,Y)]$ donc (4) implique (5).
De plus on a, $\mathbb{V}(h(y)) = \mathbb{E}[h^2(y)] - \mathbb{E}[h(y)]^2$, ainsi (5) nous donne (6).

En regroupant les variances du même coté de l'équation on a,

$$\mathbb{V}(f(X,Y)) - \mathbb{V}(h(Y)) = \mathbb{E}[f^2(X,Y)] - \mathbb{E}[h^2(Y)]$$

Or,

$$h^2(Y) = \mathbb{E}[f(X,y)]^2$$

On peut donc écrire,

$$\begin{aligned}\mathbb{E}[h^2(Y)] = \int_{\mathbb{R}} h^2(y)p_Y(y)\mathrm{d}y &= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} f(x,y)p_X(x)\,\mathrm{d}x\right)^2 p_X(y)\mathrm{d}y &(7)\\ &\leq \int_{\mathbb{R}} \left(\int_{\mathbb{R}} f^2(x,y)p_X(x)\,\mathrm{d}x\right) p_Y(y)\mathrm{d}y = \int_{\mathbb{R}} g(y)p_Y(y)\mathrm{d}y = \mathbb{E}[f^2(X,Y)] &(8)\end{aligned}$$

En utilisant l'inégalité de Jensen, avec la fonction convexe $x \mapsto x^2$, on a (7) $\leq$ (8), car $\mathbb{E}[f(X,y)]^2 \leq \mathbb{E}[f^2(X,y)]$.

Ainsi, puisque $\mathbb{E}[h^2(Y)] \leq \mathbb{E}[f^2(X,Y)]$, on peut conclure que,

$$\mathbb{V}(f(X,Y)) - \mathbb{V}(h(Y)) \geq \mathbb{E}[f^2(X,Y) \geq 0 \quad \Leftrightarrow \quad \mathbb{V}(f(X,Y)) \geq \mathbb{V}(h(Y))$$

2. f bornée implique f^2 bornée

5. Relation entre les fonctions de répartitions

On suppose ici, que X_2 est une variable aléatoire réelle dont la fonction de répartition est donnée par F_2. On cherche à calculer la fonction de répartition, notée G_2, de $\lambda_2 e^{\beta_2 X_2}$.

$$\begin{aligned} G_2(t) &= \mathbb{P}(\lambda_2 e^{\beta_2 X_2} \leq t) = \mathbb{P}\left(e^{\beta_2 X_2} \leq \frac{t}{\lambda_2}\right) = \mathbb{P}\left(\beta_2 X_2 \leq \ln\left(\frac{t}{\lambda_2}\right)\right) \\ &= \mathbb{P}\left(X_2 \leq \frac{\ln(t) - \ln(\lambda_2)}{\beta_2}\right) = F_2\left(\frac{\ln(t) - \ln(\lambda_2)}{\beta_2}\right) \end{aligned}$$

Ceci grâce à l'indépendance de X_1 et X_2

6. Méthode de Monte-Carlo utilisant les propriétés précédentes

Toujours sous les condtions d'indépendance de X_1 et X_2 on a,

$$\begin{aligned} p = \mathbb{P}(Z > t) &= 1 - \mathbb{P}(Z \leq t) = 1 - \mathbb{P}(\lambda_1 e^{\beta_1 x_1} + \lambda_2 e^{\beta_2 x_2} \leq t) \\ &= 1 - \iint_{\mathbb{R}^2} \mathbb{1}_{\{\lambda_1 e^{\beta_1 x_1} + \lambda_2 e^{\beta_2 x_2} \leq t\}} f_{X_1}(x_1) f_{X_2}(x_2)\, \mathrm{d}x_1\, \mathrm{d}x_2 \\ &= 1 - \int \left(\int_{\mathbb{R}^2} \mathbb{1}_{\{\lambda_2 e^{\beta_2 x_2} \leq t - \lambda_1 e^{\beta_1 x_1}\}} f_{X_2}(x_2)\, \mathrm{d}x_2 \right) f_{X_1}(x_1)\, \mathrm{d}x_1 \\ &= 1 - \mathbb{E}[G_2(t - \lambda_1 e^{\beta_1 X_1})] \end{aligned}$$

Ceci en utilisant Fubini-Tonelli, puisque les densités sont positives, et l'indicatrice est bornée, puis en se ramenant à la définition de la fonction de répartition. On obtient alors,

$$p = \mathbb{E}\left[1 - G_2(t - \lambda_1 e^{\beta_1 X_1})\right]$$

Puisque 1 est une constante.

Il faut maintenant vérifier qu'on a encore réduit la variance, en comparaison de la première méthode proposée, ie que cette méthode est plus précise.

On note $\psi(X_1, X_2) = \mathbb{1}_{\{\lambda_1 e^{\beta_1 X_1} + \lambda_2 e^{\beta_2 X_2} > t\}}$, puisque ψ est une indicatrice, elle est bornée. En utilisant les propriétés démontrées en partie 4., on a,

$$\mathbb{V}\left(\psi(X_1, X_2)\right) \geq \mathbb{V}\left(\mathbb{E}[\psi(x_1, X_2)]\right)$$

Or,

$$\begin{aligned} \mathbb{E}[\psi(x_1, X_2)] &= \int_{\mathbb{R}} \mathbb{1}_{\{\lambda_1 e^{\beta_1 x_1} + \lambda_2 e^{\beta_2 x_2} > t\}} f_{X_2}(x_2)\, \mathrm{d}x_2 \\ &= \int_{\mathbb{R}} \mathbb{1}_{\{\lambda_2 e^{\beta_2 x_2} > t - \lambda_1 e^{\beta_1 x_1}\}} f_{X_2}(x_2)\, \mathrm{d}x_2 \\ &= 1 - \int_{\mathbb{R}} \mathbb{1}_{\{\lambda_2 e^{\beta_2 x_2} \leq t - \lambda_1 e^{\beta_1 x_1}\}} f_{X_2}(x_2)\, \mathrm{d}x_2 \\ &= 1 - G_2(t - \lambda_1 e^{\beta_1 X_1}) \end{aligned}$$

Ainsi, on a bien,

$$\mathbb{V}\left(\mathbb{1}_{\{Z>t\}}\right) \geq \mathbb{V}\left(1 - G_2(t - \lambda_1 e^{\beta_1 X_1})\right)$$

À nouveau on propose une méthode d'échantillonnage moyen, p sera estimé par le processus suivant.

Soit $(X_1^i)_{1\leq i\leq N}$, une suite de variables aléatoires indépendantes de densité f_{X_1}, un estimateur sans biais de p est,

$$\tilde{\theta}_N = \frac{1}{N}\sum_{i=1}^{N}\left(1 - G_2(t - \lambda_1 e^{\beta_1 X_1^i})\right)$$

Pour la simulation de la fonction G_2 on utilisera la partie 5. On a,

$$G_2(t - \lambda_1 e^{\beta_1 X_1}) = F_2\left(\frac{1}{\beta_2}\ln\left(\frac{t - \lambda_1 e^{\beta_1 X_1}}{\lambda_2}\right)\right)$$

On définira une fonction G comme suit, pour simplifier le code,

```
def G(x) :
    return norm.cdf(1/b2*np.log(np.abs((t-l1*np.exp(b1*x))/l2)),
                    loc=0, scale=1)
```

Il y a plusieurs choses à remarquer sur l'implémentation de G,

- $G(x) = G_2(t - \lambda_1 e^{\beta_1 x})$, avec t fixé.
- La fonction de répartition F_2 de la variable X_2 est la fonction de répartition d'une loi normale $\mathcal{N}(0,1)$, dû à la complexité de son expression on utilisera directement la fonction dédiée de python pour obtenir sa valeur :

  ```
  scipy.stats.norm.cdf(t, loc=0, scale=1) #P(X <= t) ou X ~ N(0,1)
  ```

- Le logarithme appelé dans la fonction ne doit jamais prendre de valeur négative ou nulle, ce qui peut être le cas dépendemment des paramètres définis. L'ajout de la valeur absolue est donc utile, et n'impactera pas l'estimation.

On implémentera alors le processus de la façon suivante,

```
Xn = []
for i in range (N) :
    X = npr.normal(0,1)
    Xn.append(1-G_2(X))
theta = np.mean(Xn) #estimateur empirique
```

Une idée de la précision de la méthode est donnée par,

```
sd = np.std(Xn) #ecart-type empirique
```

7. Méthode de Monte-Carlo et fonction de répartition dans le cas gaussien

On se place à nouveau dans le cas où $X = (X_1, X_2)$ est un vecteur gaussien centré tel que $\mathbb{V}(X_1) = \mathbb{V}(X_2) = 1$ et $\mathrm{Cov}(X_1, X_2) = \rho$, avec $|\rho| \leq 1$.

On souhaite retravailler dans le même esprit que la partie 6., mais cette fois-ci les variables ne sont plus indépendantes.

Pour se ramener à un cas où les variables sont indépendantes, on utilise le fait que $X_2 - \rho X_1$ est indépendant de X_1.

$$p = \mathbb{P}(Z > t) = 1 - \mathbb{P}(Z \leq t) = 1 - \mathbb{P}(\lambda_1 e^{\beta_1 X_1} + \lambda_2 e^{\beta_2 X_2} \leq t)$$

On effectue le changement de variables suivant,

$$\begin{cases} U_1 = X_1 \\ U_2 = X_2 - \rho X_1 \end{cases} \quad \Leftrightarrow \quad \begin{cases} X_1 = U_1 \\ X_2 = U_2 + \rho U_1 \end{cases}$$

On cherche alors à isoler U_2, en utilisant l'indépendance de U_1 et U_2, ainsi que le théorème de Fubini-Tonelli, on peut écrire,

$$\begin{aligned}
\mathbb{P}(\lambda_1 e^{\beta_1 U_1} + \lambda_2 e^{\beta_2 (U_2 + \rho U_1)} \leq t) &= \iint_{\mathbb{R}^2} \mathbb{1}_{\{\lambda_1 e^{\beta_1 u_1} + \lambda_2 e^{\beta_2 (u_2 + \rho u_1)} \leq t\}} f_{U_1}(u_1) f_{U_2}(u_2) \, \mathrm{d}u_1 \, \mathrm{d}u_2 \\
&= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \mathbb{1}_{\{e^{\rho\beta_2 u_1} \left(\lambda_1 e^{(\beta_1 - \rho\beta_2) u_1} + \lambda_2 e^{\beta_2 u_2}\right) \leq t\}} f_{U_2}(u_2) \, \mathrm{d}u_2 \right) f_{U_1}(u_1) \, \mathrm{d}u_1 \\
&= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \mathbb{1}_{\{\lambda_2 e^{\beta_2 u_2} \leq t e^{-\rho\beta_2 u_1} - \lambda_1 e^{(\beta_1 - \rho\beta_2) u_1}\}} f_{U_2}(u_2) \, \mathrm{d}u_2 \right) f_{U_1}(u_1) \, \mathrm{d}u_1 \\
&= \mathbb{E}\left[G_2(t e^{-\rho\beta_2 U_1} - \lambda_1 e^{(\beta_1 - \rho\beta_2) U_1}) \right] \\
&= \mathbb{E}\left[G_2(t e^{-\rho\beta_2 X_1} - \lambda_1 e^{(\beta_1 - \rho\beta_2) X_1}) \right]
\end{aligned}$$

De plus, en utilisant la partie 5., on a alors,

$$G_2(t e^{-\rho\beta_2 X_1} - \lambda_1 e^{(\beta_1 - \rho\beta_2) X_1}) = F_2\left(\frac{1}{\beta_2} \ln\left(\frac{t e^{-\rho\beta_2 X_1} - \lambda_1 e^{(\beta_1 - \rho\beta_2) X_1}}{\lambda_2} \right) \right)$$

Ainsi, on en déduit,

$$p = \mathbb{E}\left[1 - F_2\left(\frac{1}{\beta_2} \ln\left(\frac{t e^{-\rho\beta_2 X_1} - \lambda_1 e^{(\beta_1 - \rho\beta_2) X_1}}{\lambda_2} \right) \right) \right]$$

On met alors en place une nouvelle méthode d'échantillonnage moyen.

On estime p via l'estimateur sans biais,

$$\check{\theta}_N = \frac{1}{N} \sum_{i=1}^{N} \left(1 - F_2\left(\frac{1}{\beta_2} \ln\left(\frac{t e^{-\rho\beta_2 X_1^i} - \lambda_1 e^{(\beta_1 - \rho\beta_2) X_1^i}}{\lambda_2} \right) \right) \right)$$

où $(X_1^i)_{1 \leq i \leq N}$, une suite de variables aléatoires indépendantes de loi $\mathcal{N}(0, 1)$.

On implémente le processus de la même façon que pour la partie 6., c'est-à dire qu'on définit, ici aussi, une fonction F telle que,

```
def F(x) :
    return norm.cdf(1/b2*np.log(np.abs((t*np.exp(-rho*b2*X)
                    -l1*np.exp((b1-rho*b2)*X))/l2)),
                    loc=0, scale=1)
```

Puis on simule notre N-échantillon, et on stocke les $(1 - F(X_k)$, pour $k = 1, \ldots, N$

```
Xn = []
for i in range (N) :
    X = npr.normal(0,1)
    Xn.append(1-F(X))
theta = np.mean(Xn) #estimateur empirique
```

Brève conclusion. On peut conclure cette étude en remarquant que les dernières méthodes de Monte-Carlo, reposant sur les propriétés des fonctions de répartition, donnent des estimations plus précises que les premières méthodes proposées.

En effet dans le cas où la probabilité p est très petite, les méthodes proposées de "prime-abord" sont peu efficaces, le travail sur la réduction de variance est donc important ici.

Références

[1] Samuel Herrmann. Cours rédigé : Théorie des probabilités, 2015-2016.

[2] Samuel Herrmann. Cours rédigé : Algorithmes Stochastiques, 2021-2022.