RO05 - TP no 3A
Processus de risque en assurance

Haojie LU et Haifei ZHANG

0. Introduction

Dans ce projet, on analyse le modele d'assurance qui est décrit au cours de temps par le processus de risque de Cramér-Lunberg, $U = (U_t, t \geq 0)$:

$$U_t = u + ct - \sum_{k=1}^{N_t} Y_k, t \geq 0$$

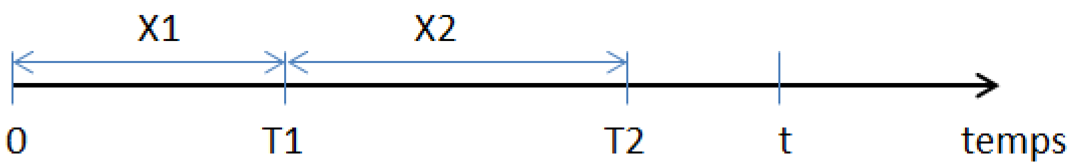
où $u > 0, c > 0, (N_t, t \geq 0)$ un processus de Poisson homogène de paramètres $\lambda > 0$ et $(Y_k, k \geq 1)$ un suite de v.a. positives i.i.d. de fonction de répartition commune F, indépendante du processus (N_t).

1. Exprimer l'événement A_i = "la ruine de la compagnie ait lieu avant t lors du i-ème sinistre", avec $t > 0$ fixé, et $i = 1, 2,$ en fonction du processus U et du temps T_1 du premier sinistre.

D'abord, on définit ddeux variables :

T_i : l'instant d'arrivée de i-ème sinistre,

$X_i, i \geq 0$: le temp d'inter-arrivées, $X_i = T_i - T_{i-1}$



Donc on a :

$$\begin{aligned} A_1 : U_{T_1} = u + cT_1 - Y_1 < 0 \quad aevc \quad T_1 < t, t > 0 \\ A_2 : U_{T_2} = u + cT_2 - (Y_1 + Y_2) < 0 \quad avec \quad T_2 < t, t > 0 \end{aligned}$$

2. Démonter que

$$P(A_1) = \lambda \int_0^t e^{-\lambda s} \bar{F}(u + cs) ds,$$

où $\bar{F} = 1 - F$.

$$\begin{aligned} P(A_1, s) &= P(u + cs - Y_1) \\ &= P(Y_1 > u + cs) \\ &= 1 - F(u + cs) \qquad \forall s \in (0, t) \end{aligned}$$

Ici, s est l'instant d'arrivée de premier sinistre, donc

$$s = X_1 = T_1$$
$$\therefore s \sim f(s) = \lambda e^{-\lambda s}$$

Car s est continu, donc

$$\begin{aligned} P(A_1) &= \int_0^t [1 - F(u+cs) \cdot f(s) ds] \\ &= \int_0^t [1 - F(u+cs) \cdot \lambda e^{-\lambda s} ds] \\ &= \lambda \int_0^t e^{-\lambda s} \cdot \bar{F}(u+cs) ds \end{aligned}$$

$$\text{où} \quad \bar{F}(u+cs) = 1 - F(u+cs)$$

3. Calculer la probabilité $P(A_2)$

Pour l'événement A_2, l'instant d'arrivée de deuxième sinistre est $s = X_1 + X_2 = T_2, T_2 \sim \Gamma(2, \lambda)$

La densité pour s est

$$\begin{aligned} f_2(s) &= \frac{(\lambda s)^{(2-1)}}{(2-1)!} \cdot \lambda e^{-\lambda s} \\ &= \lambda^2 s e^{-\lambda s} \qquad s \geq 0 \end{aligned}$$

On rappelle les formulaire : si F est la $f.r$ de la variable aléatoire X est G est la $f.r$ de la variable aléatoire Y, la lois de la variable aléatoire $X+Y$ est

$$P(X+Y < x) = F * G(x)$$

Donc la probabilité de A_2 à i'instant s est

$$\begin{aligned} P(A_2, s) &= P(u + cs - (Y_1 + Y_2) < 0) \\ &= P(Y_1 + Y_2 > u + cs) \\ &= 1 - P(Y_1 + Y_2 < u + cs) \\ &= 1 - F^{*2}(u+cs) \end{aligned}$$

Pour $\forall s \in (0, t)$, on fiat l'integration sur t:

$$\begin{aligned} P(A_2) &= \int_0^t [1 - F^{*2}(u+cs)] \cdot f_2(s) ds \\ &= \int_0^t \overline{F^{*2}}(u+cs) \lambda^2 s e^{-\lambda s} ds \\ &= \lambda^2 \int_0^t s e^{-\lambda s} \overline{F^{*2}}(u+cs) ds \end{aligned}$$

$$\text{où} \quad \overline{F^{*2}}(u+cs) = 1 - F^{*2}(u+cs)$$

4. Démontrer, par un calcul de l'espérance $E[U_t]$, qu'une condition de viabilité de la compagnie est que : $c - \lambda E[Y_1] > 0$.

Calculer l'espérance $E[Ut]$

$$\begin{aligned}
E[U_t] &= E[u + ct - \sum_{k=1}^{N_t} Y_k] \qquad t > 0 \\
&= u + ct - E[\sum_{k=1}^{N_t} Y_k] \\
&= u + ct - E[E[\sum_{k=1}^{N_t} Y_k | N_t]] \\
&= u + ct - E[N_t \cdot E[Y_1]] \\
&= u + ct - E[N_t] \cdot E[Y_1] \\
&= u + ct - \lambda t \cdot E[Y_1]
\end{aligned}$$

Si le cas de viabilité, $E[U_t] > u$

$$\begin{aligned}
&E[U_t] = u + ct - \lambda t \cdot E[Y_1] > u \\
&\Rightarrow c > \lambda E[Y_1]
\end{aligned}$$

5. Soit $r(t) := P(U_t < 0)$ la probabilité de ruine dans $[0, t]$. Démontrer que

$$r(t) := \sum_{n \geq 0} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \overline{F^{*n}}(u + ct)$$

Démonstration :

$$\begin{aligned}
r(t) &= P(U_t < 0) \\
&= P(u + ct - \sum_{k=1}^{N_t} Y_k < 0) \\
&= P(Y_1 + Y_2 + \ldots + Y_{N_t} > u + ct) \\
&= \sum_{n=0}^{\infty} [1 - P(Y_1 + Y_2 + \ldots + Y_n \leq u + ct)] \cdot P(N_t = n) \\
&= \sum_{n=0}^{\infty} \overline{F^{*n}}(u + ct) \cdot e^{-\lambda t} \frac{(\lambda t)^n}{n!} \\
&= \sum_{n=0}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} \cdot \overline{F^{*n}}(u + ct)
\end{aligned}$$

6. Démontrer que si

$$\frac{1}{t} \sum_{k=1}^{N_t} Y_k \to \rho, \qquad t \to \infty,$$

alors $\rho = \lambda E[Y_1]$. La constante ρ est appelée *moyenne des indemnités par unité de temps*.La quantité $\eta := (c - \rho)/\rho$ est appelée *charge de sécurité*.

$$\frac{1}{t}\sum_{k=1}^{N_t} Y_k \to \rho = \frac{N_t}{t}\cdot\frac{1}{N_t}\cdot\sum_{k=1}^{N_t} Y_k$$
$$\because E[N_t] = \lambda t$$
$$\therefore \frac{N_t}{t} \quad \xrightarrow{p.s} \quad \lambda$$
$$\frac{1}{N_t}\cdot\sum_{k=1}^{N_t} Y_k \quad \xrightarrow{p.s} \quad \frac{1}{N_t}\cdot E[N_t]\cdot E[Y_1] = E[Y_1]$$
$$\therefore \frac{1}{t}\sum_{k=1}^{N_t} Y_k \quad \xrightarrow{p.s} \quad \lambda\cdot E[Y_1] = \rho, \qquad t\to\infty$$

7. SIMULATION STOCHASTIQUE Soient : $\lambda = 0.01$, F est la f.r. de la loi log-normale de paramètres $\mu = 1,\ et\ \sigma = 0.5, u = 100\ etc = 1.$

(a). Réaliser une trajectoire du processus Ut pour $0 \leq t \leq min\{1000, Truine\}$. Faire une

Après nombreux de simulations, on a trouvé que avec les paramètres $\mu = 1$, $\sigma = 0.5$, il est presque impossible de ruiner. Car le résultat n'été pas sensible, on a changé le paramètre de $\mu = 1$ à $\mu = 3$. ci-dessous est une figure de notre simulation.

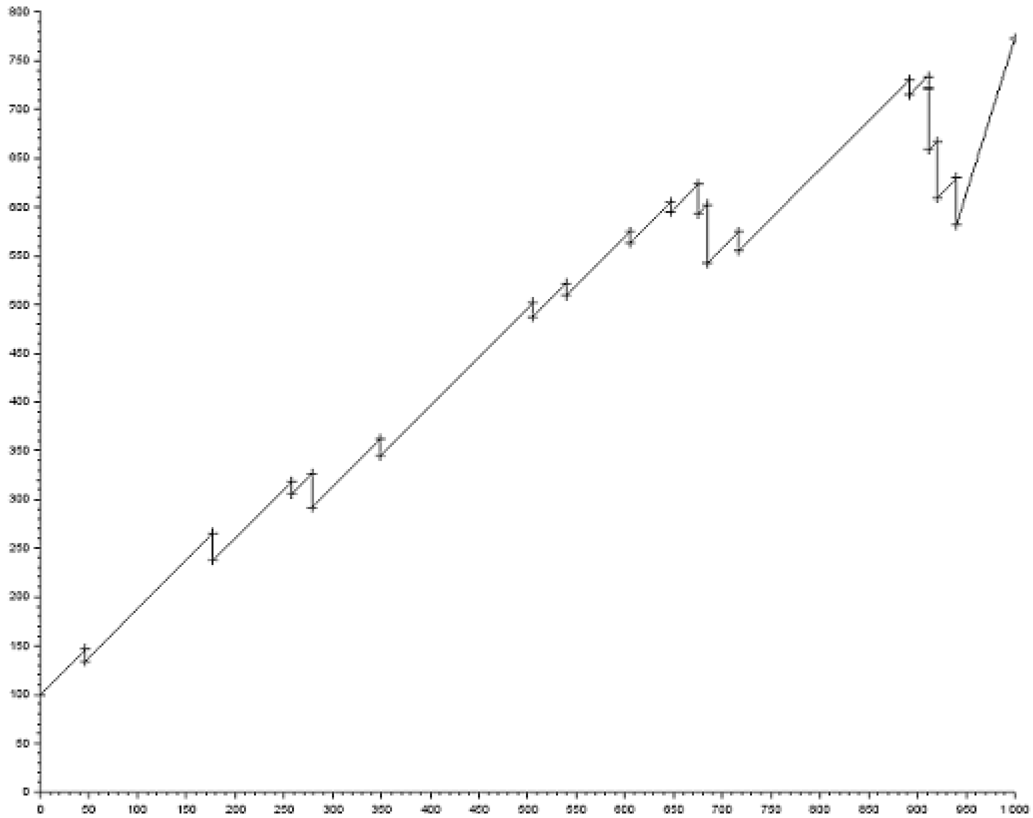


Figure1 Réalisation d'une trajectoire

(b). En réalisant N trajectoire de U_t, estimer par Monte Carlo le temps moyen de ruine de la compagnie d'assurance.

En réalisant 1000 trajectoires, on a obtenu le temps moyen de ruine comme ci-dessous. Dans ce problème, on ne prend que les trajectoires où la ruine apparait.

$$TMoyen_{ruine} = \frac{\sum_{i=0}^{1000} TRuine_i \cdot I\{dans\ trjectoire_i\ ruine\ appara\hat{\imath}t\}}{\sum_{i=0}^{1000} I\{ruineappara\hat{\imath}t\}}$$

```
-->nombre_ruine
 nombre_ruine  =

    5.

-->T_moyen_ruine
 T_moyen_ruine  =

    415.85285
```

Figure2 Le temps moyen de ruine

(c).Calculer N pour que la précision soit $(\alpha, \theta) = (0,01; 0,01)$.

Pour ce problème, on a retirer 1000 trajectoires avec ruine avant $t = 1000$. On a obtenu l'espérance et la variance comme ci-dessous :

```
-->mean(Temps_ruine)
 ans  =

    440.413

-->variance(Temps_ruine)
 ans  =

    195735.5

-->
```

Figure3 L'espérance et la variance de temps moyen ruine de 1000 trajectoires avec ruine

Calcul de N_0

$$N_0 = \frac{var((Temps_ruine))}{\theta\alpha^2} = \frac{195735.5}{10^{-6}} \approx 1.96 \times 10^{11}$$

(d).Partant d'une réalisation sans ruine de Ut, sur un intervalle de temps $[0, T]$, estimer ρ et η.

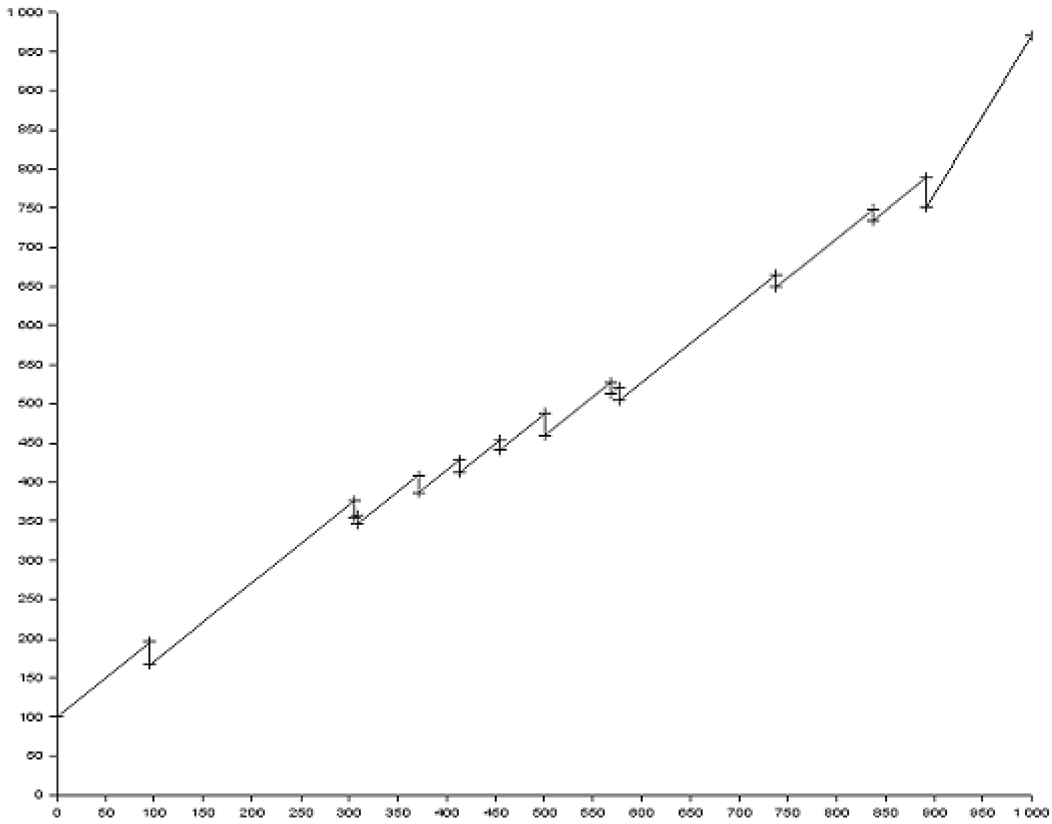


Figure4 Un trajectoire sans ruine

Dans cette réalisation, il y a 12 sinistres, donc on a obtenue les estimateurs comme ci-dessous :

$$\hat{\rho} = \frac{1}{t}\sum_{k=1}^{N_t} Y_k = 0.262$$
$$\hat{\eta} = \frac{c-\hat{\rho}}{\hat{\rho}} = \frac{1-0.262}{0.262} = 2.82$$

ANNEX

La fonction de simulation :

```
function [T, U, Truine]=simulation()
lamda=0.01
mu=3
sigma=0.5
u=100
c=1

//generer deux suites de 20 valeurs qui suivent la loi unif(0,1)
unif_1=rand(20,1)
unif_2=rand(20,1)

//generer  20 valeurs qui suivent la loi normal(0,1) par Box-Muller
y_k=zeros(20,1)
for j=1:20
    y_k(j)=sqrt(log(unif_1(j))*-2)*cos(2*%pi*unif_2(j))
end
```

```
//generer  20 valeurs qui suivent la loi log-normal(mu,sigma^2)
y=%e**(y_k*sigma+mu)

//generer 20 valeurs qui suivent la loi exp(lamda)
xi=log(1-unif_1)/(-1*lamda)

//calculer les points du temps ou les sinstres ont lieu
T=zeros(41,1)
for i=2:2:40
   T(i)=T(i-1)+xi(i/2)
   T(i+1)=T(i)
end

U=zeros(41,1)
U(1)=u
Truine=41
for i=2:2:40
    U(i)=U(i-1)+c*(T(i)-T(i-1))
    U(i+1)=U(i)-y(i/2)
    if (U(i+1)<0|T(i+1)>1000)
        Truine=i+1
        break
    end
end

if T(Truine)>1000
   T(Truine)=1000
   T(Truine-1)=1000
   U(Truine)=U(Truine-1)+c*(1000-T(Truine-2))
   U(Truine-1)=U(Truine)
end

endfunction
```

La fonction pour tracer les trajectoires :

```
[T,U,Truine]=simulation()
plot2d(0,0,1,rect=[0,0,12,10], frameflag=3)
xpoly(T(1:Truine),U(1:Truine),"lines")
f=gce()
set(f,"mark_style",1)
```

La fonction pour calculer le temps moyen de ruine pendant 1000 simulations :

```
nombre_ruine=0
T_moyen_ruine=0
for i=1:1000
    [T,U,Truine]=simulation()
    if(T(Truine)<1000)
        nombre_ruine=nombre_ruine+1
        T_moyen_ruine=T_moyen_ruine+T(Truine)
     end
end
T_moyen_ruine=T_moyen_ruine/nombre_ruine
```

La fonction pours stimuler 1000 réalisations avec ruine avant $t = 1000$:

```
n=0
Temps_ruine=zeros(1000,1)
while(n<=1000)
    [T,U,Truine]=simulation()
    if(T(Truine)<1000)
        n=n+1
        Temps_ruine(n)=T(Truine)
     end
end
```