IFT-3655: Modèles Stochastiques **A-2024**

Prof. Pierre L'Ecuyer

DEVOIR 8

Devoir à remettre le *vendredi 29 novembre 2024, avant 10h30.*

Les devoirs doivent être faits individuellement: un devoir par étudiant. Il est très important de bien expliquer tout ce que vous faites. Dans la correction, on accordera beaucoup d'importance à la clarté des explications. Attention au plagiat: il n'est pas permis de copier et/ou modifier des solutions venant de quelqu'un d'autre, ou prises sur Internet ou dans des livres. Rendez votre devoir dans Studium sous forme d'un fichier .pdf. Aucune programmation à faire pour ce devoir.

---

1. *(13 points)*

Des événements surviennent selon un processus de Poisson de taux $\lambda > 0$. Chaque fois qu'un événement survient, on peut décider de s'arrêter ou de continuer. Le but du jeu est de s'arrêter au dernier événement qui a lieu avant le temps T, où $T > 1/\lambda$ est une constante fixée à l'avance et connue. Si on s'arrête lors d'un événement qui a lieu au temps t pour $0 \le t \le T$ et si aucun autre événement n'a lieu durant l'intervalle $(t, T]$, alors on gagne, sinon (dans tous les autres cas) on perd. En particulier, si aucun événement n'a lieu durant $[0, T]$, on perd. On peut définir une stratégie qui consiste à s'arrêter au premier événement qui a lieu après le temps s, où s est une constante de notre choix, $0 \le s \le T$.

(a) Quelle est la probabilité de gagner si on utilise cette stratégie, en fonction de s?

(b) Quelle est la valeur optimale de s, qui maximise notre probabilité de gagner?

(c) Montrez que la probabilité de gagner en utilisant cette stratégie optimale est $1/e$, où e est la constante d'Euler.

2. *(12 points)*

Supposons que les durées (en heures) entre les arrivées des autobus successifs à un arrêt donné suivent la loi uniforme sur $(0, 1)$ et sont indépendantes. Les passsagers arrivent selon un processus de Poisson de taux λ par heure. Supposons qu'un autobus vient tout juste de quitter l'arrêt. Soit X le nombre de personnes qui vont prendre le prochain autobus. Trouvez $\mathbb{E}[X]$ et $\mathrm{Var}(X)$. Comment peut-on calculer toute la loi de probabilité de X, i.e., $\mathbb{P}[X = x]$ pour chaque entier $x \ge 0$?

Aide: On sait que

$$\int_0^1 y^n e^{-ay} \mathrm{d}y = \frac{n!}{a^{n+1}} \left(1 - e^{-a} \sum_{i=0}^{n} \frac{a^i}{i!}\right).$$

Numéro #1

On cherche à s'arrêter au dernier événement entre 0 et T. Si on s'arrête à "t", alors il ne devrait avoir aucun événement entre t+1 et T sinon on a perdu.

(a) $N(T)$: nbr d'événements entre $[0,T]$
$N(s)$: nbr d'événements entre $[0,s]$

Selon la stratégie adoptée, on s'arrête au premier événement après le temps s, disons au temps t. $(t > s)$

Probabilité qu'il n'y a aucun événement entre (t,T) est $\mathbb{P}[N(T) - N(t) = 0] = e^{-\lambda(T-t)}$ qui représente la probabilité cumulative des événements de t+1 à T.

Probabilité d'atterir au 1er événement après le temps s est $\mathbb{P}[N(t) - N(s) = 1] = \lambda e^{-(t-s)}$ qui représente la probabilité fixe qu'un seul événement se produit.

Il faut regarder tous les temps $s < t < T$:

$$\mathbb{P}\{\text{Gagner}\} = \int_s^T \lambda e^{-\lambda(t-s)} \cdot e^{-\lambda(T-t)}\, dt$$
$$= \lambda \int_s^T e^{-\lambda(T-s)}\, dt$$
$$= \lambda e^{-\lambda(T-s)}(T-s)$$

(b) Pour maximiser la probabilité de gagner, on va dériver par rapport à s et poser égale à 0 :

$$\Rightarrow \frac{d}{ds}\left[\lambda e^{-\lambda(T-s)}(T-s)\right] = 0$$

$$\Rightarrow \lambda \cdot \left(\frac{d}{ds}\left[e^{-\lambda(T-s)}\right] \cdot (T-s) + e^{-\lambda(T-s)} \cdot \frac{d}{ds}\left[T-s\right]\right) = 0$$

$$\Rightarrow \lambda \cdot \left(\lambda e^{-\lambda(T-s)}(T-s) - e^{-\lambda(T-s)}\right) = 0$$

$$\Rightarrow \lambda e^{-\lambda(T-s)}\left(\lambda(T-s) - 1\right) = 0$$

$$\Rightarrow \lambda(T-s) = 1$$

$$\Rightarrow -s = \frac{1}{\lambda} - T$$

$$\Rightarrow s = T - \frac{1}{\lambda}$$

(c) Prenons notre probabilité de gagner et la valeur optimale de "s" :

$$\mathbb{P}\{\text{Gagner avec stratégie}\} = \lambda e^{-\lambda(T-s)}(T-s)$$
$$= \lambda e^{-\lambda(T-T+\frac{1}{\lambda})}\left(T-T+\frac{1}{\lambda}\right)$$
$$= \frac{\lambda}{\lambda} e^{-\frac{\lambda}{\lambda}}$$
$$= \frac{1}{e}, \text{ avec } s = T - \frac{1}{\lambda}$$

Numéro #2

La variable $Y_j \sim \mathcal{U}(0,1)$ est le temps d'attente avant l'arrivée du prochain autobus après le $(j-1)$ ème autobus (indep.).

Le nombre de passagers qui va arriver d'ici le prochain autobus dépend du temps $Y = y$.
Donc, cela se traduit conditionnellement comme

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y = y]] = \sum_{i=1}^{\lambda} \mathbb{E}[Y_i] = \lambda \mathbb{E}[Y_i]$$

étant donné que les Y_i sont indep. du taux λ.

Puisque $Y_i \sim \mathcal{U}(0,1)$, $\mathbb{E}[Y_i] = \frac{1}{2}$.

Donc,

$$\mathbb{E}[X] = \frac{\lambda}{2}$$

Pour la variance, le processus est semblable :

$$\begin{aligned} \mathbb{V}[X] &= \mathbb{E}[\mathbb{V}[X \mid Y]] + \mathbb{V}[\mathbb{E}[X \mid Y]] \\ &= \lambda \mathbb{E}[Y] + \lambda^2 \mathbb{V}[Y] \\ &= \frac{\lambda}{2} + \lambda^2 \mathbb{V}[Y] \end{aligned}$$

Puisque $Y_i \sim \mathcal{U}(0,1)$, $\mathbb{V}[Y_i] = \frac{1}{12}$

Donc,

$$\mathbb{V}[X] = \frac{\lambda}{2} + \frac{\lambda^2}{12}$$

On connait $\mathbb{E}[X]$ et $\mathbb{V}[X]$ mais toujours pas la loi de probabilité de X. Sauf qu'en conditionnant sur Y, on peut trouver.

$$\begin{aligned} \mathbb{P}[X = x] &= \mathbb{E}[\mathbb{P}[X = x \mid Y]] \\ &= \int_0^1 \mathbb{P}[X = x \mid Y = y] \cdot \underbrace{f_Y(y)}_{\frac{1}{1-0}} \, dy \\ &= \int_0^1 \frac{(\lambda y)^x e^{-\lambda y}}{x!} \, dy \\ &= \frac{\lambda^x}{x!} \left(\frac{x!}{\lambda^{x+1}} \left(1 - e^{-\lambda} \sum_{i=0}^{x} \frac{\lambda^i}{i!} \right) \right) \\ &= \frac{1}{\lambda} \left(1 - e^{-\lambda} \sum_{i=0}^{x} \frac{\lambda^i}{i!} \right) \end{aligned}$$

parce que le nombre d'arrivées de passagers dans l'intervalle de $Y = y$ suit une loi de Poisson avec un taux λy.

3. *(10 points)*

Une compagnie d'assurance reçoit deux types de réclamations, les types 1 et 2. Soit $N_i(t)$ le nombre de réclamations de type i reçues durant l'intervalle de temps $[0, t]$, pour $i = 1, 2$. Supposons que $\{N_1(t),\, t \geq 0\}$ et $\{N_2(t),\, t \geq 0\}$ sont des processus de Poisson independents de taux $\lambda_1 = 10$ et $\lambda_2 = 1$, et que les montants des réclamations de type i sont des variables aléatoires indépendantes exponentielles de moyenne μ_i, avec $\mu_1 = \$1000$ et $\mu_2 = \$5000$. On vient de recevoir une réclamation de \$4000. Quelle est la probabilité que cette réclamation soit de type 1?

4. *(15 points)*

Supposons que les clients arrivent à un magasin de détail selon un processus de Poisson de taux B de 9 heures à midi, $2B$ de midi à 14 heures, puis de $1.2B$ de 14 heures à 17 heures. Ici, B est une variable aléatoires de loi connue qui ne prend que des valeurs positives et prend la même valeur pour toute la journée. Le taux d'arrivée des clients est donc lui-même aléatoire. Une plus grande valeur de B correspondra à une journée plus achalandée.

On dispose d'une fonction pour générer B selon la bonne loi, et d'une autre fonction pour générer des (imitations de) variables aléatoires indépendantes uniformes sur $(0, 1)$. Donnez un algorithme permettant de simuler les instants des arrivées successives pour ce processus, pour une journée, en utilisant ces deux ingrédients.

Numéro #3

La probabilité que la réclamation de 4000$ reçue est un évènement du processus de Poisson N_1 est :

$$\mathbb{P}[T=1 \mid C=4000] = \frac{\mathbb{P}[C=4000 \mid T=1]\cdot\mathbb{P}[T=1]}{\mathbb{P}[C=4000]}$$

où $\mathbb{P}[C=4000] = \sum_{i=1}^{2} \mathbb{P}[C=4000 \mid T=i]\cdot\mathbb{P}[T=i]$

Aussi, $\mathbb{P}[C=4000 \mid T=i] = \frac{1}{\mu_i} e^{-\frac{1}{\mu_i}\cdot 4000}$ et

$$\mathbb{P}[T=i] = \frac{\lambda_i}{\sum_{j=1}^{2}\lambda_j}$$

La probabilité qu'une réclamation provient d'un type "i" est la proportion du taux d'arrivées des deux car si on observe une réclamation, la proba qu'elle soit de type 1 est $\propto$ au "poids" de λ_1 sur $\lambda_1+\lambda_2$.

Et $\mathbb{P}[C=4000 \mid T=i]$ c'est juste proba qu'une réclamation fixe durant $[0,t]$ soit 4000, donc $Exp\left(\frac{1}{\mu_i}\right)$.

Donc on a,

$$\mathbb{P}[C=4000 \mid T=1]\cdot\mathbb{P}[T=1] = \frac{1}{1000}e^{-4}\cdot\frac{10}{10+1} = \frac{1}{1100}e^{-4}$$

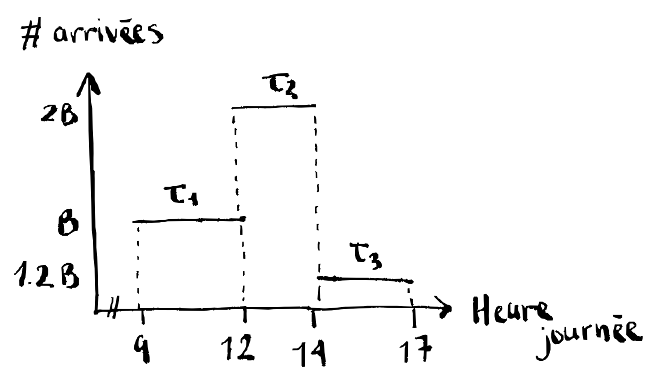
$$\mathbb{P}[C=4000 \mid T=2]\cdot\mathbb{P}[T=2] = \frac{1}{5000}e^{-\frac{4}{5}}\cdot\frac{1}{10+1} = \frac{1}{55000}e^{-\frac{4}{5}}$$

Finalement,

$$\mathbb{P}[T=1 \mid C=4000] = \frac{\frac{1}{1100}e^{-4}}{\frac{1}{1100}e^{-4}+\frac{1}{55000}e^{-\frac{4}{5}}} \approx 0,671$$

Il y a 67,1% de chances que la réclamation provient du type 1. Bien que la moyenne des montants de réclamation du type 1 soit loin de 4000$, il faut savoir que le taux du processus est 10 fois plus élevé que celui du type 2. Ce taux compense la valeur faible des montants du type 1.

Numéro #4



Chaque intervalle τ_i pour $i=1,2,3$ contient des instants d'arrivées T_j que nous devons simulés.

Ce que j'ai pensé à faire au début était de générer un nombre b_i d'uniforme (ex: $b_2 = 2B$) et de poser $X_j = \min(\tau_i) + \max(\tau_i) \cdot \mathcal{U}_j$ mais cela <u>ne fonctionne pas</u> parce que cela veut dire que certains intervalles seraient plus propice à un # élevé d'arrivées.

Au lieu, je vais procéder de la manière suivante.

Pour chaque intervalle τ_i, je vais construire la fonction de taux cumulé correspondante. On obtiendra des équations affines $a_i(\ell)$ qui formeront un $a(\ell)$ linéaire par morceaux. À savoir que ℓ est la différence entre un temps t et 9 ($t-9$, ex: $t=12$, $\ell=3$). En fait on change de référencement.

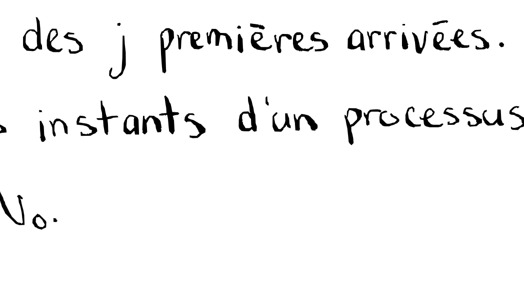
1. Cas $\ell = [0,3]$: $a_1(\ell) = b_1\ell$
2. Cas $\ell = [3,5]$: $a_2(\ell) = b_2(\ell-3) + a_1(3)$
3. Cas $\ell = [5,8]$: $a_3(\ell) = b_3(\ell-5) + a_2(5)$

Donc,

$$a(\ell) = \begin{cases} b_1\ell, & \text{si } 0 \le \ell \le 3 \\ b_2(\ell-3) + 3b_1, & \text{si } 3 \le \ell \le 5 \\ b_3(\ell-5) + 2b_2 + 3b_1, & \text{si } 5 \le \ell \le 8 \end{cases}$$

où $b_1 = b$, $b_2 = 2b$ et $b_3 = 1.2b$ avec b la valeur générée fixe de la loi B.

Avec notre fonction de taux cumulés, on peut générer nos T_j comme à la diapo 7, c'est à dire par inversion avec nos X_j, qui sont les temps d'arrivées des client. Ces temps Z_j suivent des exponentielles de paramètre $\lambda = 1$. Ils représentent la durée entre les arrivées.



et les X_j sont les instants soient les sommes cumulatives des j premières arrivées. En fait, ce sont les instants d'un processus de Poisson standard N_0.

Donc,

$$X_1 = Z_1$$
$$X_2 = Z_1 + Z_2$$
$$\vdots$$
$$X_j = \sum_{k=1}^{j} Z_k \qquad \text{où } Z_k \sim Exp(1)$$

Ces instants Z_j peuvent être générer encore par inversion.

Inverse de la PDF de Z_j :

$$f(Z_j) = \mathcal{U}_j$$
$$\lambda e^{-\lambda Z_j} = \mathcal{U}_j$$
$$e^{-Z_j} = \mathcal{U}_j$$
$$-Z_j = \ln[\mathcal{U}_j]$$
$$Z_j = -\ln[\mathcal{U}_j]$$

On se retrouve avec

$$X_j = -\sum_{k=1}^{j} \ln[\mathcal{U}_k]$$

On peut désormais simuler nos T_j avec $T_j = a^{-1}(X_j)$ où l'inverse de $a(\ell)$ est

$$a^{-1}(x_j) = \begin{cases} \dfrac{x_j}{b_1}, & \text{si } 0 \le x_j \le 3b_1 \\ \dfrac{x_j - 3b_1}{b_2} + 3, & \text{si } 3b_1 \le x_j \le 3b_1 + 3b_2 \\ \dfrac{x_j - (2b_2 + 3b_1)}{b_3} + 5, & \text{si } 3b_1 + 2b_2 \le x_j \le 3b_1 + 2b_2 + 3b_3 \end{cases}$$

L'algorithme final ressemblerait à ça (ne pas si fier complètement mais plutôt au processus décris plus haut)

```
fonction simulationArriveesClients()
  générer valeur de B avec genererValeurB()
  poser b1 = B
  poser b2 = 2*B
  poser b3 = 1.2*B

  fonction a(t)
    si 0 <= t <= 3, alors retourner b1*t
    sinon si 3 <= t <= 5, alors retourner b1*3 + b2*(t-3)
    sinon si 5 <= t <= 8, alors retourner b1*3 + b2*2 + b3*(t-5)

  fonction a_inv(y)
    si 0 <= y <= 3*b1, alors retourner y/b1
    sinon si 3*b1 < y <= 3*b1 + 2*b2, alors retourner ((y-3*b1)/b2) + 3
    sinon si 3*b1 + 2*b2 < y <= 3*b1 + 2*b2 + 3*b3, alors retourner ((y-3b1-2b2)/
b3) + 5

  poser N = genererPoisson(a(8))
  poser Xs = []

  répéter
    poser U_j = genererUniforme(0,1)
    poser Z_j = -ln(U_j)
    poser X_j = 0

    pour i de 1 à longueur(Xs)
      X_j += Xs[i]

    X_j += Z_j
    Xs.ajouter(X_j)

  poser Ts = []
  pour i de 1 à longueur(Xs)
    poser T_j = a_inv(Xs[i])
    Ts.ajouter(T_j)

  trier(Ts)
  retourner Ts
```