EJERCICIOS 11 Y 13 - PRÁCTICO 4 - 2019

TASAS DE RIESGO

OSCAR BUSTOS - PATRICIA KISBYE - GEORGINA FLESIA.

1. Introducción

Se trata de resolver el siguiente ejercicio:

Ejercicio 11: Sea X una variable aleatoria cuya distribución de probabilidad es $P(X = j) = p_j$ con $j = 1,2,\ldots$. Sea:

$$\lambda_n = P(X = n|X > n-1) = \frac{p_n}{1-\sum_{j=1}^{n-1} p_j} \ , \ n = 1,2,\ldots$$

Las cantidades λ_n, son las tasas discretas de riesgo. Considerando a X como el tiempo (discreto) de vida de algún artículo, λ_n representa la probabilidad de que habiendo sobrevivido hasta el tiempo $n-1$, muera en el tiempo n.

Mostrar que $p_1 = \lambda_1$ y que

$$p_n = (1-\lambda_1)(1-\lambda_2)\cdots(1-\lambda_{n-1})\lambda_n. \tag{1}$$

2. Resolución

Para resolver este ejercicio aplicamos el principio de inducción. Para $n = 1$, notemos que

$$p_1 = P(X = 1) = P(X = 1,\ X > 0) = P(X = 1 \mid X > 0)\cdot P(X > 0).$$

Como $P(X > 0) = 1$, entonces resulta:

$$p_1 = P(X = 1 \mid X > 0) = \lambda_1.$$

Para $n = 2$, de manera análoga vemos que

$$p_2 = P(X = 2) = P(X = 2,\ X > 1) = P(X = 2 \mid X > 1)\cdot P(X > 1) = \lambda_2\cdot(1-p_1).$$

Dado que $p_1 = \lambda_1$, resulta

$$p_2 = \lambda_2\cdot(1-\lambda_1).$$

Supongamos entonces que (1) se cumple para cierto $n \in \mathbb{N}$. Entonces, para p_{n+1} tenemos que:

$$\begin{aligned}
p_{n+1} &= P(X = n+1) = P(X = n+1 \mid X > n)\cdot P(X > n) \\
&= \lambda_{n+1}\cdot(1-\sum_{j=0}^{n} p_n) \\
&= \lambda_{n+1}\cdot(1-p_1-p_2-\cdots-p_n)
\end{aligned} \tag{2}$$

Ahora bien, por hipótesis inductiva tenemos que

$$p_n = (1-\lambda_1)(1-\lambda_2)\ldots(1-\lambda_{n-1})\lambda_n, \tag{3}$$

pero por definición también es cierto que

$$p_n = P(X = n \mid X > n-1)\cdot P(X > n-1) = \lambda_n\cdot(1-p_1-p_2-\cdots-p_{n-1}). \tag{4}$$

Luego de (3) y (4) resulta que

$$(1-\lambda_1)(1-\lambda_2)\dots(1-\lambda_{n-1}) = 1-p_1-p_2-\dots-p_{n-1}. \tag{5}$$

Volviendo a (2), tenemos que

$$\begin{aligned}
p_{n+1} &= (1-p_1-p_2-\dots-p_n)\cdot\lambda_{n+1}\\
&= (1-p_1-p_2-\dots-p_{n-1})\cdot\lambda_{n+1} - p_n\lambda_{n+1}\\
&= (1-\lambda_1)(1-\lambda_2)\dots(1-\lambda_{n-1})\cdot\lambda_{n+1} - p_n\lambda_{n+1} \qquad \text{por (5)}\\
&= (1-\lambda_1)(1-\lambda_2)\dots(1-\lambda_{n-1})\cdot\lambda_{n+1} - (1-\lambda_1)(1-\lambda_2)\dots(1-\lambda_{n-1})\lambda_n\lambda_{n+1} \qquad \text{por H.I.}\\
&= (1-\lambda_1)(1-\lambda_2)\dots(1-\lambda_{n-1})(1-\lambda_n)\cdot\lambda_{n+1}
\end{aligned}$$

Por lo tanto, para todo $n \geq 1$ se cumple que

$$p_n = (1-\lambda_1)(1-\lambda_2)\dots(1-\lambda_{n-1})\lambda_n,$$

3. Simulación de X conocidas las tasas de riesgo

Ejercicio 13: Suponiendo que $0 \leq \lambda_n \leq \lambda < 1$, para todo $n \geq 1$; considerar el siguiente algoritmo para generar una variable aleatoria con tasas dicretas de riesgo $\{\lambda_n\}$:

Paso 1: S = 0

Paso 2: Generar U, $Y = \text{Ent}\left[\dfrac{\log(U)}{\log(1-\lambda)}\right]+1$

Paso 3: S = S + Y

Paso 4: Generar U

Paso 5: Si U $\leq \lambda_s/\lambda$, tomar X = S y terminar. Caso Contrario, ir a Paso 2.

a) ¿Cuál es la distribución de Y en el paso 2?

b) Explique cómo funciona el algoritmo.

c) Argumentar que X resulta una variable aleatoria con tasas discretas de riesgo $\{\lambda_n\}$.

Solución:

La distribución de Y en el paso 2 es geométrica con probabilidad de éxito λ. Por ello debe ser claro en este algoritmo que $\lambda < 1$.

Este algoritmo genera sucesivamente valores de una variable aleatoria geométrica con probabilidad de éxito λ, y acumula los valores obtenidos en S. De esta manera, si Y toma sucesivamente los valores k_1, k_2, ..., k_n, entonces la variable S tomará los valores

$$k_1, \quad k_1+k_2, \quad k_1+k_2+k_3, \quad \dots \quad k_1+k_2+\dots+k_n.$$

El algoritmo acepta el valor $S = k_1$ con probabilidad λ_{k_1}/λ, y en tal caso devuelve ese valor: $X = k_1$. Si lo rechaza, entonces acepta el valor k_1+k_2 con probabilidad $(\lambda_{k_1+k_2}/\lambda)$, y en tal caso devuelve $X = k_1+k_2$. El proceso sigue hasta que eventualmente se han rechazado $n-1$ geométricas y se acepta el valor $S = k_1+k_2+\dots+k_n$ con probabilidad

$$\frac{\lambda_{k_1+k_2+\dots+k_n}}{\lambda}.$$

Para probar que el algoritmo genera valores de una variable con tasas de riesgo λ_n, veremos que la probabilidad de generar el valor $X = j$ es p_j, donde

$$p_j = (1-\lambda_1)(1-\lambda_2)\dots(1-\lambda_{j-1})\lambda_j.$$

Para mayor claridad, analizaremos los casos $j = 1$ hasta $j = 3$. Luego procederemos a comprobar el resultado por inducción en j.

3.1. Caso $j = 1$

Para $j = 1$, tenemos que el algoritmo devuelve el valor 1 si y sólo si Y toma el valor 1 y este valor es aceptado con probabilidad λ_1/λ. Luego:

$$P(X = 1) = P(Y = 1,\ U \leq \lambda_1/\lambda) = \lambda \cdot \frac{\lambda_1}{\lambda} = \lambda_1.$$

. Dado que $\lambda_1 = p_1$, el resultado es cierto para $j = 1$.

3.2. Caso $j = 2$

Para $j = 2$, existen dos casos en que el algoritmo devuelve el valor 2. Un caso es haber generado dos geométricas $Y_1 = 1$, $Y_2 = 1$ y en el cual se rechaza $S_1 = Y_1$ y se acepta $S_2 = Y_1 + Y_2 = 2$. El otro caso es generar una geométrica $Y_1 = 2$ y aceptar el valor $S_1 = Y_1 = 2$. Así:

$$\begin{aligned} P(X = 2) &= P(S_1 = 1,\ U_1 > \frac{\lambda_1}{\lambda},\ S_2 = 2, U_2 < \frac{\lambda_2}{\lambda}) + P(S_1 = 2, U_1 < \frac{\lambda_2}{\lambda}) \\ &= \underbrace{P(Y_1 = 1, U_1 > \frac{\lambda_1}{\lambda}, Y_2 = 1, U_2 < \frac{\lambda_2}{\lambda})}_{\lambda\cdot(1-\lambda_1/\lambda)\cdot\lambda\cdot\lambda_2/\lambda} + \underbrace{P(Y_1 = 2, U_1 < \frac{\lambda_2}{\lambda})}_{(1-\lambda)\lambda\cdot\lambda_2/\lambda} \\ &= (\lambda - \lambda_1)\lambda_2 + (1-\lambda)\lambda_2 = (1-\lambda_1)\lambda_2. \end{aligned} \tag{6}$$

Así queda probado el caso $j = 2$.

3.3. Caso $j = 3$

Para $j = 3$, los casos posibles contemplan la generación de tres geométricas con valor 1, o dos geométricas ($Y_1 = 1$, $Y_2 = 2$ o $Y_1 = 2$, $Y_2 = 1$), o una única geométrica $Y_1 = 3$. Así tenemos que:

$$\begin{aligned} P(X = 3) &= P(Y_1 = 1,\ U_1 > \frac{\lambda_1}{\lambda},\ Y_2 = 1,\ U_2 > \frac{\lambda_2}{\lambda},\ Y_3 = 1,\ U_3 < \frac{\lambda_3}{\lambda}) \\ &\quad + P(Y_1 = 1,\ U_1 > \frac{\lambda_1}{\lambda},\ Y_2 = 2,\ U_2 < \frac{\lambda_3}{\lambda}) \\ &\quad + P(Y_1 = 2,\ U_1 > \frac{\lambda_2}{\lambda},\ Y_2 = 1,\ U_2 < \frac{\lambda_3}{\lambda}) \\ &\quad + P(Y_1 = 3,\ U_1 < \frac{\lambda_3}{\lambda}). \end{aligned} \tag{7}$$

Notemos que cada renglón de la fórmula (7) corresponde a los valores de S obtenidos como:

$$S = 1+1+1, \qquad S = 1+2, \qquad S = 2+1, \qquad S = 3.$$

Así resulta:

$$\begin{aligned} P(X = 3) &= \lambda \cdot (1 - \frac{\lambda_1}{\lambda}) \cdot \lambda \cdot (1 - \frac{\lambda_2}{\lambda}) \cdot \lambda \cdot \frac{\lambda_3}{\lambda} \\ &\quad + \underbrace{\lambda}_{Y_1=1} \cdot (1 - \frac{\lambda_1}{\lambda}) \cdot \underbrace{(1-\lambda)\lambda}_{Y_2=2} \cdot \frac{\lambda_3}{\lambda} + \underbrace{(1-\lambda)\lambda}_{Y_1=2} \cdot (1 - \frac{\lambda_2}{\lambda}) \cdot \underbrace{\lambda}_{Y_2=1} \cdot \frac{\lambda_3}{\lambda} + \underbrace{(1-\lambda)^2\lambda}_{Y_1=3} \cdot \frac{\lambda_3}{\lambda} \\ &= (\lambda - \lambda_1) \cdot (\lambda - \lambda_2) \cdot \lambda_3 + (1-\lambda)(\lambda - \lambda_1) \cdot \lambda_3 + (1-\lambda)(\lambda - \lambda_2) \cdot \lambda_3 + (1-\lambda)^2 \lambda_3 \end{aligned} \tag{8}$$

Notemos que λ_3 es un factor común en todos los términos, y cada término tiene (además de λ_3) productos de 2 factores. Estos factores son de la forma $(1-\lambda)$ o $(\lambda - \lambda_k)$ con $k = 1, 2$.

Separemos los términos que contienen a $\lambda-\lambda_2$ de los que no lo tienen. Los que no tienen a $\lambda-\lambda_2$ necesariamente tienen al menos un factor de la forma $(1-\lambda)$. Tenemos entonces:

$$P(X=3)=\lambda_3\cdot\big((\lambda-\lambda_2)[\lambda-\lambda_1)+(1-\lambda)]+(1-\lambda)[(\lambda-\lambda_1)+(1-\lambda)]\big).$$

Las expresiones entre corchetes se reducen al factor que acompaña a λ_2 en el caso $j=2$ (fórmula (6)) y es igual a $1-\lambda_1$. Por lo tanto resulta:

$$P(X=3)=\lambda_3(\lambda-\lambda_2+1-\lambda)(1-\lambda_1)=(1-\lambda_1)(1-\lambda_2)\lambda_3,$$

como se quería probar.

3.4. Caso general

Para el caso general, observemos que el algoritmo devuelve el valor j con $n=1,2,\ldots,$ hasta $n=j$ iteraciones. Si realiza n iteraciones, significa que rechazó $n-1$ veces valores crecientes:

$$k_1,\quad k_1+k_2,\quad \ldots,\quad k_1+k_2+\cdots+k_{n-1}$$

y aceptó $k_1+k_2+\cdots+k_{n-1}+k_n=j$ con probabilidad λ_j/λ.

Dado que la variable Y puede tomar cualquier valor natural, entonces k_1, k_1+k_2, $\ldots$, $k_1+k_2+\cdots+k_n=j$ es cualquier secuencia creciente de n números naturales tal que el último valor es j. La denotamos entonces:

$$k_1=s_1,\quad k_1+k_2=s_2,\quad \ldots\quad k_1+k_2+\cdots+k_n=s_n=j.$$

Observemos que entonces

$$k_1=s_1,\qquad k_i=s_i-s_{i-1},\qquad 2\le i\le n.$$

Luego, la probabilidad de devolver j luego de exactamente n iteraciones es

$$\sum_{1\le s_1<s_2<\ldots<s_n=j} P(\text{rechazar } s_1,\text{rechazar } s_2,\ldots,\text{rechazar } s_{n-1},\text{aceptar } s_n=j).$$

Ahora, extendiendo a todos los valores de n posibles tenemos que:

$$P(X=j)=\sum_{n=1}^{j}\sum_{1\le s_1<s_2<\ldots<s_n=j} P(\text{rechazar } s_1,\text{rechazar } s_2,\ldots,\text{rechazar } s_{n-1},\text{aceptar } s_n=j).$$

Rechazar s_1 implica generar $Y_1=s_1$ con probabilidad $(1-\lambda)^{s_1-1}\lambda$ y rechazar $S_1=s_1$ con probabilidad λ_{s_1}/λ. Rechazar s_2 implica haber generado $Y_2=s_2-s_1$ con probabilidad $(1-\lambda)^{(s_2-s_1)-1}\lambda$ y rechazar $S_2=s_2$ con probabilidad λ_{s_2}/λ, y así siguiendo. Finalmente, aceptar $s_n=j$ implica haber generado $Y_n=s_n-s_{n-1}$ con probabilidad $(1-\lambda)^{s_n-s_{n-1}-1}\lambda$ y aceptar $S_n=s_n=j$ con probabilidad λ_j/λ. Luego:

$$\begin{aligned}
&P(\text{rechazar } s_1,\quad \ldots\quad ,\text{rechazar } s_{n-1},\text{aceptar } s_n=j)\\
&\quad= (1-\lambda)^{s_1-1}(1-\lambda)^{s_2-s_1-1}\ldots(1-\lambda)^{s_n-s_{n-1}-1}\lambda^n\cdot(1-\frac{\lambda_{s_1}}{\lambda})(1-\frac{\lambda_{s_2}}{\lambda})\ldots(1-\frac{\lambda_{s_{n-1}}}{\lambda})\cdot\frac{\lambda_{s_n}}{\lambda}\\
&\quad= (1-\lambda)^{j-n}(\lambda-\lambda_{s_1})(\lambda-\lambda_{s_2})\ldots(\lambda-\lambda_{s_{n-1}})\cdot\lambda_j
\end{aligned}$$

Así resulta:

$$\begin{aligned}
P(X=j) &= \sum_{n=1}^{j}(1-\lambda)^{j-n}\sum_{1\le s_1<s_2<\ldots<s_n=j}(\lambda-\lambda_{s_1})(\lambda-\lambda_{s_2})\ldots(\lambda-\lambda_{s_{n-1}})\cdot\lambda_j\\
&= \lambda_j\cdot\sum_{n=1}^{j}(1-\lambda)^{j-n}\sum_{1\le s_1<s_2<\ldots<s_n=j}(\lambda-\lambda_{s_1})(\lambda-\lambda_{s_2})\ldots(\lambda-\lambda_{s_{n-1}}) \qquad (9)
\end{aligned}$$

Afirmación: *Para todo $j \geq 1$, la fórmula (9) es igual a*

$$\lambda_j \cdot (\lambda - \lambda_1) \cdot (\lambda - \lambda_2) \cdots (\lambda - \lambda_{j-1}). \tag{10}$$

La afirmación ya ha sido probada para $j = 1$ hasta $j = 3$. Supongamos que la afirmación se cumple para $X = j - 1$. Probaremos por inducción que también vale para $X = j$.

Notemos que λ_j multiplica a una suma donde cada término es un producto de j factores, donde $j - n$ son de la forma $(1 - \lambda)$ y los n restantes son n distintos elegidos entre

$$(\lambda - \lambda_1), \quad (\lambda - \lambda_2), \quad \ldots \quad , (\lambda - \lambda_{j-1}).$$

Además estas posibilidades se cubren exhaustivamente.

En primer lugar separamos en la sumatoria sobre n en (9) a los términos que contienen al factor $(\lambda - \lambda_{j-1})$ de los que no lo tienen.

Para cada valor de $n = 2, 3, \ldots, j$, $(\lambda - \lambda_{j-1})$ es un factor en un término si y sólo si $s_{n-1} = j - 1$. Observemos que este factor no aparece en el término $n = 1$ pues es el caso de haber generado una sola geométrica $Y_1 = j$:

$$(1 - \lambda)^{j-1} \lambda_j.$$

Por lo tanto la parte de la sumatoria con este factor se puede escribir como:

$$\begin{aligned}
& \sum_{n=2}^{j} (1-\lambda)^{j-n} \sum_{s_1 < \ldots < s_{n-1} = j-1} (\lambda - \lambda_{s_1})(\lambda - \lambda_{s_2}) \ldots (\lambda - \lambda_{s_{n-2}})(\lambda - \lambda_{j-1}) \\
= \quad & (\lambda - \lambda_{j-1}) \sum_{n=2}^{j} (1-\lambda)^{(j-1)-(n-1)} \sum_{s_1 < \ldots < s_{n-1} = j-1} (\lambda - \lambda_{s_1})(\lambda - \lambda_{s_2}) \ldots (\lambda - \lambda_{s_{n-2}}) \\
= \quad & (\lambda - \lambda_{j-1}) \sum_{n=1}^{j-1} (1-\lambda)^{(j-1)-n} \sum_{s_1 < \ldots < s_n = j-1} (\lambda - \lambda_{s_1})(\lambda - \lambda_{s_2}) \ldots (\lambda - \lambda_{s_{n-1}}) \qquad (11)
\end{aligned}$$

La sumatoria en (11) es igual a $P(X = j-1)/\lambda_{j-1}$ que por hipótesis inductiva es $(1 - \lambda_1) \cdots (1 - \lambda_{j-2})$. Luego (11) es igual a:

$$(\lambda - \lambda_{j-1})(1 - \lambda_1) \cdots (1 - \lambda_{j-2}). \tag{12}$$

Los términos en (9) que no contienen a $\lambda - \lambda_{j-1}$ contienen n factores entre

$$(\lambda - \lambda_1), \quad (\lambda - \lambda_2), \ldots, (\lambda - \lambda_{j-2}),$$

y por lo tanto $n < j - 1$ necesariamente. En particular, $s_{n-1} \leq j - 2$ para todos los n donde esto tiene sentido: $n = 2, 3, \ldots, j - 1$. Recordemos que $n = 1$ es el caso $s_1 = j$.

Ahora bien, para cada $n = 2, 3, \ldots, j - 1$, se tiene una biyección entre los conjuntos:

$$\{(s_1, s_2, \ldots, s_{n-1}) \mid s_{n-1} \leq j - 2\} \qquad \text{y} \qquad \{(s_1, s_2, \ldots, s_{n-1}, s_n) \mid s_n = j - 1\}.$$

Así, la parte de la sumatoria en (11) sin el factor $\lambda - \lambda_{j-1}$ se puede escribir como:

$$\begin{aligned}
& (1-\lambda) \sum_{n=1}^{j-1} (1-\lambda)^{(j-1)-n} \sum_{s_1 < s_2 < \ldots < s_{n-1} \leq j-2} (\lambda - \lambda_{s_1})(\lambda - \lambda_{s_2}) \ldots (\lambda - \lambda_{s_{n-1}}) \\
= \quad & (1-\lambda) \sum_{n=1}^{j-1} (1-\lambda)^{(j-1)-n} \sum_{s_1 < s_2 < \ldots < s_n = j-1} (\lambda - \lambda_{s_1})(\lambda - \lambda_{s_2}) \ldots (\lambda - \lambda_{s_{n-1}}) \qquad (13)
\end{aligned}$$

y nuevamente, por hipótesis inductiva esto es

$$(1 - \lambda)(1 - \lambda_1) \cdots (1 - \lambda_{j-2}). \tag{14}$$

Sumando (12) y (14) y reemplazando en (9) obtenemos:

$$P(X = j) = \lambda_j \cdot (1 - \lambda_1)(1 - \lambda_2) \dots (1 - \lambda_{j-1}),$$

como queríamos probar.

Por lo tanto la **Afirmación** es válida para todo $j \geq 1$ y el algoritmo simula una variable aleatoria con tasas de riesgo λ_j.

4. Ejemplos

4.1. Binomial negativa, con $r = 2$

Si X es binomial negativa con número de éxitos $r = 2$, tenemos que

$$P(X = k) = (k-1) \cdot (1-p)^{k-2} p^2, \qquad k = 2, 3, \dots,$$

Por otra parte, $P(X > k - 1)$ es la probabilidad de que en los primeros $k - 1$ ensayos haya menos de dos éxitos: 0 éxitos o 1 éxito. Luego:

$$P(X > k-1) = P(0 \text{ éxitos}) + P(1 \text{ éxito}) = (1-p)^{k-1} + (k-1) \cdot (1-p)^{k-2} p = (1-p)^{k-2}(1 + (k-2)p).$$

Por lo tanto, las tasas de riesgo para $k = 2, 3, \dots,$ vienen dadas por:

$$\lambda_k = \frac{P(X = k)}{P(X > k-1)} = \frac{(k-1) \cdot (1-p)^{k-2} p^2}{(1-p)^{k-2}(1 + (k-2)p)} = \frac{(k-1)p^2}{1 + (k-2)p}.$$

En particular, tenemos que $\lambda_k < p$, para todo k. En efecto:

$$\lambda_k = \frac{(k-2)p + p}{(k-2)p + 1} \cdot p < 1 \cdot p = p.$$

Si extendemos la binomial negativa a $j = 1$ definiendo $P(X = 1) = 0$, entonces $\lambda_1 = 0$.

Dado que $\lambda_k / \lambda = \frac{(k-1)p}{1+(k-2)p}$, el algoritmo resulta:

```python
def binomial_negativa_2(p):
    "usando tasas de riesgo"
    S = 0
    lamda = p
    while 1:
        Y = int(log(1 - random()) / log(1 - lamda)) + 1
        S = S + Y
        lamda_S_sobre_lamda = (S - 1) * p / (1 - p + (S - 1) * p)
        U = random()
        if U < lamda_S_sobre_lamda:
            return S
```