Cambios de Variable y Generación de Números Aleatorios

Pablo L. De Nápoli

Departamento de Matemática
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Probabilidades y Estadística para Computación
Primer cuatrimestre de 2023

Parte I

Cambios de Variable

Planteo del problema general

Un problema que aparecemente en la teoría de probabilidades y sus aplicaciones es el siguiente:

Problema que vamos a estudiar hoy

Si X es una variable aleatoria (continua), y $\varphi : \mathbb{R} \to \mathbb{R}$ es una función ¿cuál es la distribución de $Y = \varphi(X)$?

El caso más sencillo: Cambios de variables monótonos

Supongamos que X es una variable aleatoria con una cierta función de distribución F como antes, y sea $Y = \varphi(X)$ donde $\varphi : \mathbb{R} \to \mathbb{R}$ es alguna función creciente, biyectiva y con una inversa $\varphi^{-1} : \mathbb{R} \to \mathbb{R}$ también creciente.

$$F_Y(y) = P\{Y \leq y\} = P\{\varphi(X) \leq y\} = P\{X \leq \varphi^{-1}(y)\} = F_X(\varphi^{-1}(y))$$

En particular si X es una variable continua, derivando vemos que

$$f_Y(y) = f_X(\varphi^{-1}(y)) \cdot (\varphi^{-1})'(y) = \frac{1}{\varphi(x)} f_Y(x) \text{ donde } x = \varphi^{-1}(y)$$

El caso más simple e importante, es el de un cambio de variable lineal dado por $\varphi(x) = ax + b$ con $a > 0$. Entonces $\varphi^{-1}(y) = \frac{y-b}{a}$.
Luego si $Y = aX + b$ entontramos que:

$$f_Y(y) = \frac{1}{a} f_X\left(\frac{y-b}{a}\right)$$

Cambios de variables lineales en la distribución normal

Supongamos que $X \sim N(\mu, \sigma^2)$ y hagamos un cambio de variable lineal, $Y = aX + b$ con $a > 0$ Según la fórmula que deudujimos recién

$$f_Y(y) = \frac{1}{a\sigma\sqrt{2\pi}} \exp\left\{ -\frac{\left(\frac{y-b}{a} - \mu\right)^2}{2\sigma^2} \right\} = \frac{1}{a\sigma\sqrt{2\pi}} \exp\left\{ -\frac{(y-(a\mu+b))^2}{2(a\sigma)^2} \right\}$$

Concluimos que $Y \sim N(a\mu + b, a^2\sigma^2)$.

Un caso particular es la estandarización de la normal: Si $X \sim N(\mu, \sigma^2)$, entonces

$$X^* = \frac{X - E(X)}{\sqrt{\mathrm{Var}(X)}} = \frac{X - \mu}{\sigma} \sim N(0, 1)$$

Esto lo vimos la clase pasada.

Otro ejemplo: la distribución log-normal

Supongamos que $X \sim N(\mu, \sigma^2)$. ¿Cuál es la distribución de $Y = e^X$?. Tomamos $\varphi(x) = e^x$, $\varphi : \mathbb{R} \to \mathbb{R}_{>0}$ es biyectiva y su inversa es $\varphi^{-1}(y) = \log y$. $\varphi^{-1} : \mathbb{R}_{>0} \to \mathbb{R}$. Recordamos que

$$f_Y(y) = f_X(\varphi^{-1}(y)) \cdot (\varphi{-}1)'(y) = \frac{1}{\varphi'(x)} f_X(x) \text{ donde } x = \varphi^{-1}(y)$$

Como $\varphi(x) = e^{\log y} = y$, encontramos que

$$f_Y(y) = \frac{1}{\sigma y \sqrt{2\pi}} \exp\left(-\frac{(\ln y - \mu)^2}{2\sigma^2}\right) \quad y > 0$$

Esta distribución se llama log normal porque si Y tiene distribución log-normal, entonces *log* $Y = X$ tiene distribución normal.

Un ejemplo de un cambio de variable no monótono

La situación es bastante más compleja si admitimos cambios de variables que no son monótonos o biyectivos.
Consideremos por ejemplo el cambio de variable $Y = X^2$. Entonces para $z > 0$ tenemos que:

$$F_Y(y) = P\{X^2 \leq y\} = P\{|X| \leq \sqrt{y}\} = P\{-\sqrt{y} \leq X \leq \sqrt{y}\} = \\ P\{X \leq \sqrt{y}\} - P\{Z < -\sqrt{y}\} = F_X(\sqrt{y}) - F_X(-\sqrt{y}^{-})$$

mientras que claramente $F_Y(y) = 0$ si $y < 0$.
En particular si X es una variable absolutamente continua con densidad f_X, encontramos (derivando como antes) que:

$$f_Y(y) = \frac{1}{2\sqrt{y}}\left[f_X(\sqrt{y}) + f_X(-\sqrt{y})\right] \quad (y > 0)$$

Un ejemplo: distribución del cuadrado de una variable normal

Sea $X \sim N(0,1)$ una variable aleatoria con distribución normal estándar. Utilizando la fórmula anterior, encontramos que $Y = X^2$ se distribuye según la densidad

$$f_Y(y) = \frac{1}{2\sqrt{y}}\left[f_X(\sqrt{y}) + f_X(-\sqrt{y})\right] = \frac{1}{2\sqrt{y}}\left[\frac{1}{\sqrt{2\pi}}e^{-y/2} + \frac{1}{\sqrt{2\pi}}e^{-y/2}\right]$$

o sea

$$f_Y(y) = \frac{1}{\sqrt{2\pi}}y^{-1/2}e^{-y/2} \quad (y > 0)$$

Esta densidad se conoce como la densidad χ^2 ("ji-cuadrado"] con un grado de libertad [abreviada χ_1^2]. Comparando con la definición de las distribuciones gama, y utilizando que $\Gamma(1/2) = \sqrt{\pi}$, vemos que coincide con la densidad $\Gamma\left(\frac{1}{2}, \frac{1}{2}\right)$.

En realidad, que $\Gamma(1/2) = \sqrt{\pi}$ sale de que hay una única constante que puedo poner para que la integral de f_Y en toda la recta sea 1.

Parte II

Generar números aleatorios con una distribución dada

Generar números aleatorios con una distribución dada

Supongamos que tenemos una distribución de probabilidad continua dada por una función de distribuciòn (acumulada) F estrictamente creciente y continua $F : \mathbb{R} \to [0,1]$.

Esta función tendrá una función inversa $F^{-1} : [0,1] \to \mathbb{R}$ dada por

$$F^{-1}(y) = x \Leftrightarrow y = F(x)$$

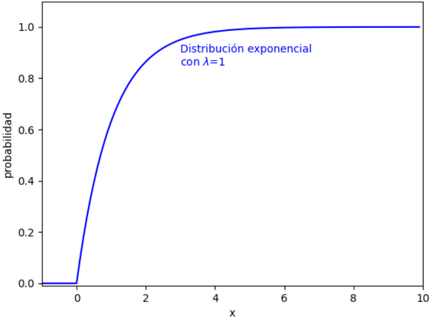
Entonces si sabemos simular la distribución uniforme podemos simular cualquier otra distribución de probabilidad F ya que si $U \sim \mathcal{U}(0,1)$, entonces $X = F^{-1}(U)$ es una variable aleatoria con distribución F.
Pues:

$$F_X(x) = P\{X \leq x\} = P\{F^{-1}(U) \leq x\} = P\{U \leq F(x)\} = F(x)$$

Un ejemplo: la distribución exponencial

La distribución exponencial $Exp(\lambda)$ de parámetro $\lambda > 0$ tiene como función de distribución acumulada

$$F(x) = \begin{cases} 1 - e^{-\lambda x} & \text{si} \quad x > 0 \\ 0 & \text{si} \quad x \leq 0 \end{cases}$$



Notamos que es estrictamente creciente y continua para $x > 0$.

Calculemos la inversa

Para calcular F^{-1} hay que resolver la ecuación

$$1 - e^{-\lambda x} = y$$

para cada $y \in (0, 1)$

$$\begin{aligned} 1 - e^{-\lambda x} = y &\Leftrightarrow e^{-\lambda x} = 1 - y \\ &\Leftrightarrow -\lambda x = \log(1 - y) \\ &\Leftrightarrow x = -\frac{\log(1 - y)}{\lambda} \end{aligned}$$

Luego

$$F^{-1}(y) = x = -\frac{\log(1 - y)}{\lambda} \; y \in (0, 1)$$

Entonces...

Conclusion

Si $U \sim \mathcal{U}(0,1)$, entonces

$$X = -\frac{\log(1-U)}{\lambda}$$

tendrá distribución exponencial $\mathrm{Exp}(\lambda)$.

¡En la computadora!

Simulación de la distribución exponencial a partir de la uniforme

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import scipy.stats as stats


# Simulamos la distribución exponencial

cuantas_veces = 1000000
U = stats.uniform.rvs(size=cuantas_veces)
X = -np.log(1 - U)

hist = sns.histplot(data=X, stat='density',
                    color='Orange', bins=50)
```

¡En la computadora! (2)

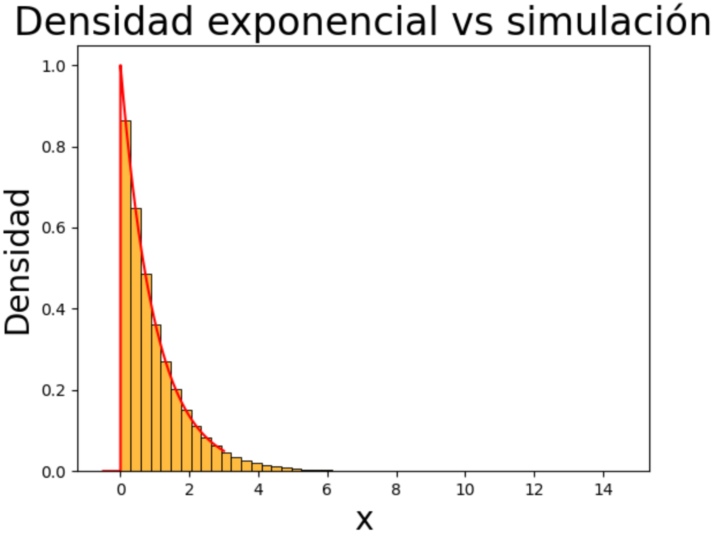
Simulación de la distribución exponencial a partir de la uniforme (2)

```
# continuamos: graficamos la densidad de probabilidad

xs = np.linspace(-0.5, 3, 100001)
distribución = stats.expon()
pdfs = distribución.pdf(xs)
plt.plot(xs, pdfs, color='red')

plt.title('Densidad exponencial vs simulación',
                        fontsize=24)
plt.xlabel('x', fontsize=20)
plt.ylabel('Densidad', fontsize=20)
plt.savefig('simular_exponencial.png')
plt.show()
```

Resultado de la simulación



Parte III

¿Y si queremos simular una distribución discreta?

La distribución acumulada de una variable discreta

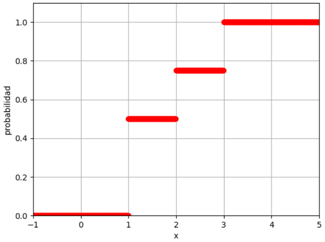
Supongamos que ahora queremos simular una variable aleatoria discreta con valores

$$x_1 < x_2 < \ldots < x_n$$

(los suponemos ordendados) distribución de probabilidades puntual

$$p_k = P\{X = x_k\}$$

La función de distribución acumulada de X es una función en escalera. Por ejemplo supongamos que X toma los valores 1,2,3 con probabilidades $1/2$, $1/4$ y $1/4$ respectivamente.



Simulando una variable aleatoria discreta

Notemos que esta función NO es biyectiva y no podemos hablar de su función inversa. Pero podemos utilizar una idea parecida:

Si $U \sim \mathcal{U}(0,1)$, podemos considerar las probabilidades acumuladas

$$a_0 = 0, \; a_k = p_1 + p_2 + \ldots + p_k \text{ si } k \geq 1$$

Notemos que son las alturas de los escalones en el gráfico anterior.

$$X = \begin{cases} x_1 & \text{si} \quad 0 = a_0 \leq U \leq a_1 \\ x_2 & \text{si} \quad a_1 < U \leq u_2 \\ x_2 & \text{si} \quad a_2 < U \leq u_3 \\ \ldots & \\ x_n & \text{si} \quad a_{n-1} < U \leq a_n = 1 \end{cases}$$

Entonces

$$P\{X = x_k\} = P\{a_{k-1} < U \leq a_k\} = a_k - a_{k-1} = p_k$$

¡En la computadora!

Simulación de una variable discreta a partir de la uniforme

```
xk = (1, 2, 3)
pk = (0.5, 0.25, 0.25)
n = len(xk)

a = {}
a[0] = 0
for k in range(1, n):
        a[k] = a[k - 1] + pk[k - 1]
a[n] = 1

import scipy.stats

def X():
        U = scipy.stats.uniform.rvs()
        for k in range(0, n):
                if a[k] < U and U <= a[k + 1]:
                        return xk[k]
```

Resultado de la simulación

Resultado de la simulación

```
xk= (1, 2, 3)

pk= (0.5, 0.25, 0.25)

a= {0: 0, 1: 0.5, 2: 0.75, 3: 1}

cuantas_veces= 10000

valor  frecuencia   f. relativa   probabilidad
1    4984    0.4984 0.5
2    2471    0.2471 0.25
3    2545    0.2545 0.25
```

Parte IV

¿Cómo hacen las computadoras para generar números pseudo-aleatorios?

Números Pseudo-Aleatorios

Como vimos anteriormente, para simular experimentos aleatorios en una computadora muchas veces se usa un generador de números pseudo-aleatorios. Estos números no son realmente aleatorios, sino que son generados mediante un algoritmo determinístico. Pero tienen la apariencia de ser aleatorios. Esto se comprueba mediante test estadísticos.

Usualmente los generadores de números pseudo aleatorios utilizan herramientas de la teoría de números como congruencias.

Aunque existen muchos métodos para esto, para darles alguna idea de como funcionan les voy a contar el que usaba mi primer computadora: la ZX-spectrum. Es el generador de números pseudo-aleatorios de Lehmer (1954).

Referencia:
`https://en.wikipedia.org/wiki/Lehmer_random_number_generator`

El generador de números pseudo-aleatorios de Lehmer

Este generador de números pseudo-aleatorios usa la secuencia

$$x_{k+1} \equiv g \cdot x_k \pmod{p}$$

donde p es un número primo y g es un número fijo convenientemente elegido. Si elegimos g y p convenientemente estos números pueden parecer aleatorios, y hacer que la sucesión x_k/p tenga aproximadamente la distribución uniforme.

Programita en Python para generar números pseudo-aleatorios

```
p= 2**16+1
g= 75
x= 1 #semilla del generador

def generador():
        global x
        x=(x*g)%p
        return x/p
```

Simulando la distribución uniforme

La ZX-spectrum entonces tenía una función RND que simulaba la distribución $\mathcal{U}(0,1)$ (aproximadamente)generando un número pseudo aleatorio x con este algoritmo, tomando

$$p = 2^{16} + 1 = 65537, g = 75$$