GENERACIÓN DE NÚMEROS ALEATORIOS CON DISTRIBUCIÓN NORMAL SIMULADO EN EL LENGUAJE PYTHON

Castillo Flores Junior *, Calapuja Apaza Luis , Nalvarte Yantas Kevin, Huanca Huanca Valerio
Universidad Nacional de Ingeniería
Lima, Perú
e-mail:*juniorcastillon6@gmail.com, lcalapujaa@uni.pe, kevinnalvarte@hotmail.com, lhuancah@uni.pe

***Abstract*—El objetivo de este artículo es mostrar como se generan los números aleatorios por computadora, y dar una visión de cuáles son los algoritmos más adecuados para generarlos. En este artículo se implementa el método de congruencia lineal mixta que genera números pseudoaleatorios con la misma probabilidad de aparecer en una secuencia dada, estos números se trabajan para obtener una distribución normal.**

***Index Terms*—Distribución normal, congruencia lineal, pseudoaleatoriedad , estocástico,**

I. INTRODUCCIÓN

Antes de la aparición de las computadoras y su capacidad de cálculo en procesos estocásticos, existieron diferentes métodos para la generación de números aleatorios, estos estaban basados en procedimientos mecánicos que generaban enormes tablas de números (Tippett, Kendall y Babbington, Rand Corporation, etc). Más adelante al rededor de los anõs 40 del siglo pasado aparecen técnicas matemáticas para la simulación de variables aleatorias y procesos no deterministas, dicho método lleva por nombre Monte Carlo.

Por otro lado existen dos tipos de generadores para números aleatorios, los que se basan en fenómenos físicos tales como el ruido atmosférico, que tiene un alto grado de entropía (ya que no se conocen las condiciones iniciales que generan estos ruidos), estos son denominados generadores de verdaderos números aleatorios (TRNG), en contrariedad con este método los generadores pseudoaleatorios (PRNG), necesitan de un estado inicial para generar los números mediante una secuencia algorítmica, por esto carecen de entropía, un buen generador pseudoaleatorio debe de incorporar algún grado de entropía dentro de su ecuación, por lo tanto este tipo de técnica debe de incluir alguna complejidad adicional, algunas de estas técnicas se utilizan para asegurar que los números sean fiables para su uso en criptografía.

En este proyecto utilizaremos métodos de congruencia lineal simulados en lenguaje Python para la generación de números pseudoaleatorios y compararemos los datos obtenidos para cada uno de estos algoritmos.

II. ESTADO DEL ARTE

En el artículo Random Number Generation: Types and Techniques [1] se describe dos métodos de generación de números aleatorios, la primera se basa en tomar como modelo fenómenos del entorno físico cuyos patrones no son aleatorios, hay quienes discrepan sobre esta aleatoriedad, ya que para que se dé un fenómeno debe de existir efectos que lo causen, si se llegara a conocer las condiciones iniciales que causan estos fenómenos, estos dejarían de ser no deterministas, sin embargo, esto es casi imposible ya que los factores causantes de un fenómeno son prácticamente infinitos, esto se puede describir como un efecto mariposa, es decir, que si se produjera una pequeña perturbación en las condiciones iniciales los cambios en los fenómenos de la naturaleza serían enormes. Entre los fenómenos físicos tomados como fuente de aleatoriedad tenemos: ruido atmosférico, decaimiento radiactivo, lasers y circuitos osciladores.

Por otro lado también existen generadores que no dependen de fenómenos de la realidad, a estos se los conoce como generadores pseudoaleatorios, estos generadores en principio parecen producir secuencias aleatorias para cualquiera que no conozca el valor inicial secreto. En un generador pseudoaleatorio básico, el valor inicial es el único factor en que se introduce la entropía al sistema. A diferencia de los verdaderos generadores de números aleatorios (que toman la entropía de un fenómeno y lo transforman en números), un generador pseudoaleatorio necesita encontrar alguna entropía para mantenerse impredecible. Las tácticas clásicas para lograr esto incluyen tomar la hora del día, la ubicación del mouse o la actividad en el teclado, esto no nos asegura que alguien no pueda replicar una secuencia conociendo los valores iniciales, ya que un humano puede replicar estas tácticas. Un generador pseudoaleatorio confiable es aquel que no nos permite replicar con facilidad una secuencia de números, este debe de tener variables que no puedan determinarse en un proceso intermedio.

El método de congruencia lineal de Lehmer es uno de los métodos más conocidos para la generación de números pseudoaleatorios, consiste en escoger convenientemente una semilla $X_0 \geq 0$, un multiplicador $a \geq 0$, incremento $c \geq 0$ y un modulo $m > X_0$,a y c .

$$X_{j+1} = a.X_j + c.mod(m) \tag{1}$$

Se debe de tener cuidado ya que la generación de aleatorios puede degenerarse, para esto se escoge convenientemente ciertos números que generen un periodo máximo de tamaño m que se describen a detalle en el artículo [2], con esto lograremos que los números obtenidos sean igualmente probables, al aparecer como mínimo una vez en la secuencia.
Se describen más métodos que derivan de la congruencia lineal de Lehmer como: método mixto de congruencias, método multiplicativo de congruencias, generador Shift-Register, generador Lagger-Fibonnaci, generador de congruencia inversa, generador de congruencia lineal combinada.
Con los métodos mencionados anteriormente se obtienen números con distribuciones uniformes, estos pueden modificarse para generar otras distribuciones. Hay muchas distribuciones pero en nuestro estudio nos enfocaremos en la distribución normal, esto se describe con mas detalle en el capítulo 5 del libro Random Number Generation and Monte Carlo Methods [3].

III. Diseño del experimento

Nuestra misión será que dado como datos la media y la desviación típica, debemos de poder generar números aleatorios de tal forma que tengan distribución normal con la media y la desviación típica dada.

La función de distribución, de la distribución normal de $N(\mu, \sigma^2)$ está definida como:

$$\Phi_{\mu,\sigma^2}(z) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt, z \in \mathbb{R}$$

Y su función densidad de probabilidad esta dada por:

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, z \in \mathbb{R}$$

Esta función nos resulta en forma de campana, y nos muestra una distribución simétrica, la media, la mediana y la moda toman el mismo valor igual a μ . Esta distribución es bastante usada, debido a que permite modelar numerosos fenómenos naturales, sociales y psicológicos.

Entonces notemos que, si hacemos $\mu = 0, \sigma^2 = 1$,obtenemos $N(0, 1)$ la cual es denominada distribución normal estándar,

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dt, x \in \mathbb{R}$$

Debemos de notar que haciendo un cambio de variable en la función de distribución normal estándar

$$z = \frac{x - \mu}{\sigma}$$

podemos obtener nuevamente la distribución normal $N(\mu, \sigma^2)$

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, z \in \mathbb{R}$$

Por lo tanto, haremos el experimento usando la distribución normal estándar $N(0, 1)$.

Esto quiere decir que los números aleatorios deben generarse teniendo en cuenta que para cada tramo existe una probabilidad diferente, que depende de que tan cerca o lejos de la media se encuentre.

Entonces siguiendo con el experimento, podemos usar una distribución uniforme para construir nuestra distribución normal estándar. Encontramos material para esto en el capítulo 5 del libro Random Number Generation and Monte Carlo Methods citado anteriormente.

Por lo tanto, en primer lugar tendremos que conseguir un algoritmo eficiente para la generación de números aleatorios distribuidos uniformemente.

Existen varios métodos para conseguir esto, pero nosotros implementaremos el algoritmo más usado y recomendado en las bibliografías, por su simpleza, facilidad y eficacia, el cual es: el método de generación de números pseudoaleatorios usando generadores de congruencia lineal mixta, el cual se basa en el método de congruencia lineal de Lehmer.

El método de congruencia lineal de Lehmer pasa a llamarse método mixto de congruencias cuando $c \neq 0$ en la siguiente sucesión

$$X_{j+1} = a.X_j + c.mod(m) \tag{2}$$

Entonces el diseño del experimento se realizará en tres partes bien diferenciadas:

1) Implementar de manera óptima un algoritmo para conseguir la generación de números aleatorios en distribución uniforme.
2) Implementar la modificación y/o el uso del algoritmo de generación de números aleatorios en distribución uniforme para obtener un generador de números aleatorios en distribución normal estándar.
3) Hacer el cambio de variable y usar el algoritmo de generación de números aleatorios en distribución normal estándar para obtener un algoritmo de generación de números aleatorios en distribución normal el cual tendrá como argumentos adicionales la media (μ) y la desviación típica (σ^2).

Tambien utilizaremos la técnica de Box-Muller, que transforma dos distribuciones uniformes en una distribución normal bivariada.
El método consiste en los siguientes pasos:

1) Se generan dos números aleatorios r_1 y r_2 , $U(0, 1)$
2) Se transforman en dos variables aleatorias normales, cada una con media 0 y varianza 1, usando transformaciones directas:

$$z_1 = (-2 \ln r_1)^{1/2} \sin(2\pi r_2) \tag{3}$$

$$z_2 = (-2 \ln r_1)^{1/2} \cos(2\pi r_2) \tag{4}$$

3) Se calculan las variables aleatorias normales x_1 y x_2 de la siguiente forma:

$$x_1 = z_1\sigma + \mu \tag{5}$$

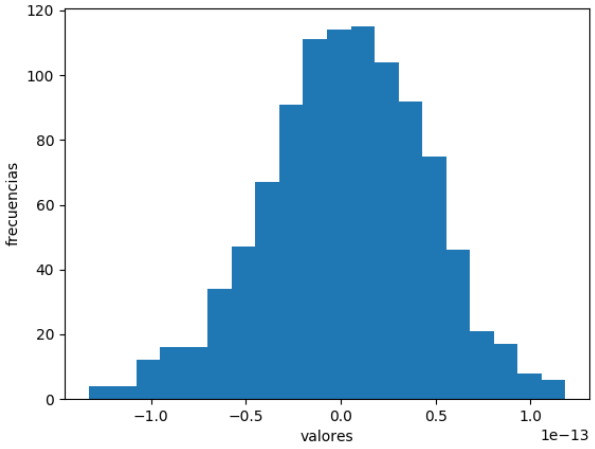
$$x_2 = z_2\sigma + \mu \tag{6}$$

IV. Experimentos y resultados

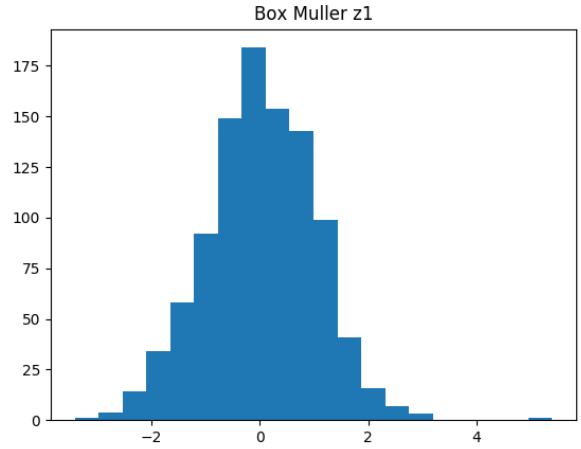
A partir del método de congruencia lineal, tomando como parámetros $a = 7^5$, $c = 630360016$ y $m = 2^{31} - 1$,elegidos convenientemente, generamos varias muestras pseudoaleatorias, dichas muestras se dividieron entre el módulo para uniformizarlas $U\ (0, 1)$, para luego aplicar el teorema del límite central que transforma esta distribución en una distribución normal según:

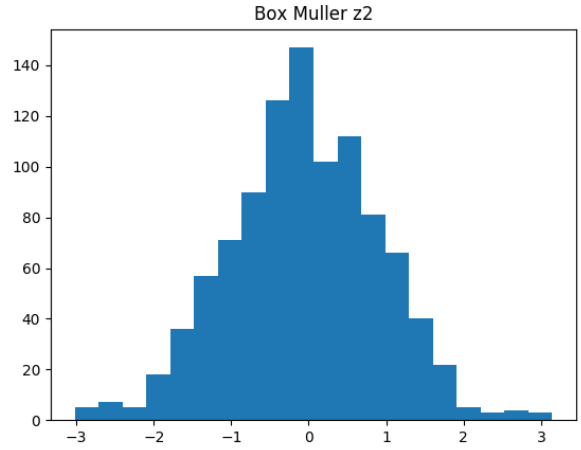
$$z = \frac{\sum_{i=1}^{n} r_i - n\mu}{(n\sigma^2)^{1/2}} \tag{7}$$

Donde r_i son los valores asociados a las variables aleatorias iid uniformes U_i en (0,1), asi obtenemos para n muestras, n valores para z , estos valores tiene una distribución $N(0, 1)$ como se obtuvo en el programa dibujando un histograma de frecuencias:

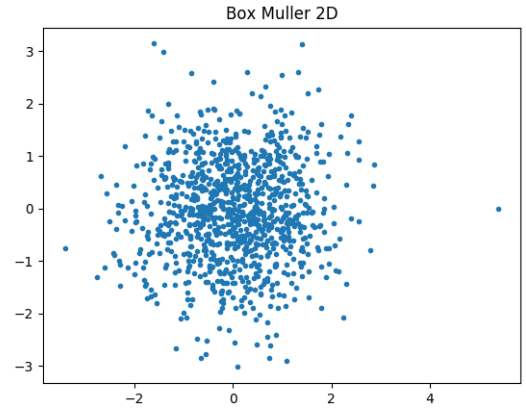


Para el método de Box-Muller utilizamos el mismo generador de congruencia lineal, y obtuvimos 2 variables aleatorias distribuidas normalmente, aplicando las ecuaciones (3) y (4):





Si graficamos esta distribución obtenida con el método de box-muller como una proyección en 2 dimensiones, obtenemos:



V. Discusión

Se debe de tener cuidado al momento de elegir los parámetros para el algoritmo de congruencia lineal, ya que debe de cumplir con ciertas condiciones para que no se degenere, es decir, su periodo máximo llegue a m-1.

En el método de convolución (teorema del limite central) nos arroja valores entre -1 y 1 de manera aleatoria, cuya distribución es normal cuando la cantidad de muestras es mayor que 5, nostros tomamos para 1000 muestras iid a partir de nuestro generador pseudoaleatorio.

El método de box-muller es un método directo de transformar 2 variables distribuidas uniformemente en U(0,1) en 2 variables aleatorias distribuidas normalmente. que en coordenadas polares nos da una distribucion bivariada normal. Nosotros graficamos la proyección en 2 dimensiones, y podemos observar que los datos se juntan hacia el centro, de la circunferencia, es decir, hay mas datos cerca de la media y a medida que nos alejamos del centro los puntos se dispersan.

VI. Conclusiones

Los métodos utilizados para la transformación de una distribucion uniforme a una con distribución normal son confialbes, siempre y cuando las muestras uniformes sean iid (independientes e identicamente distribuidas).

Se debe de elegir buenos parametros para el generador de congruencia lineal ,que cumplan las condiciones mencionadas, para que nuestros numeros pseudoaleatorios no degeneren.

Acknowledgment

References

[1] David DiCarlo (2012), Random Number Generation: Types and Techniques, [archivo PDF]. Disponible en: https://pdfs.semanticscholar.org

[2] Alfonso Mancilla (2000), Números Aleatorios ,[archivo PDF]. Disponible en: http://ciruelo.uninorte.edu.co/pdf/ingenieria_desarrollo/

[3] James E. Gentle, Random Number Generation and Monte Carlo Methods, 2da ed., Estados Unidos, 2005, pp.165–217.