

MÁSTER EN INTELIGENCIA ARTIFICIAL

MÉTODOS DE SIMULACIÓN

Práctica Grupo 3

Autores

LUIS COUTO SELLER
IRENE MARBÁN ÁLVAREZ
AÍDA MUÑOZ MONJAS

October 24, 2022

Contenidos

1	Generación de números y variables aleatorias	2
1.1	Enunciado	2
1.2	El método Monty Python	2
1.3	Comparación de métodos	4
1.3.1	Resultados en velocidad.	4
1.3.2	Resultados en el test de bondad de ajuste χ^2	5
1.3.3	Resultados en el test de valores altos de sigma.	5
1.3.4	Resultados del test de aleatoriedad	5
1.3.5	Resultados del test de correlación interbloque.	5
1.3.6	Conclusiones	5
	Bibliografía	6

1 Generación de números y variables aleatorias

1.1 Enunciado

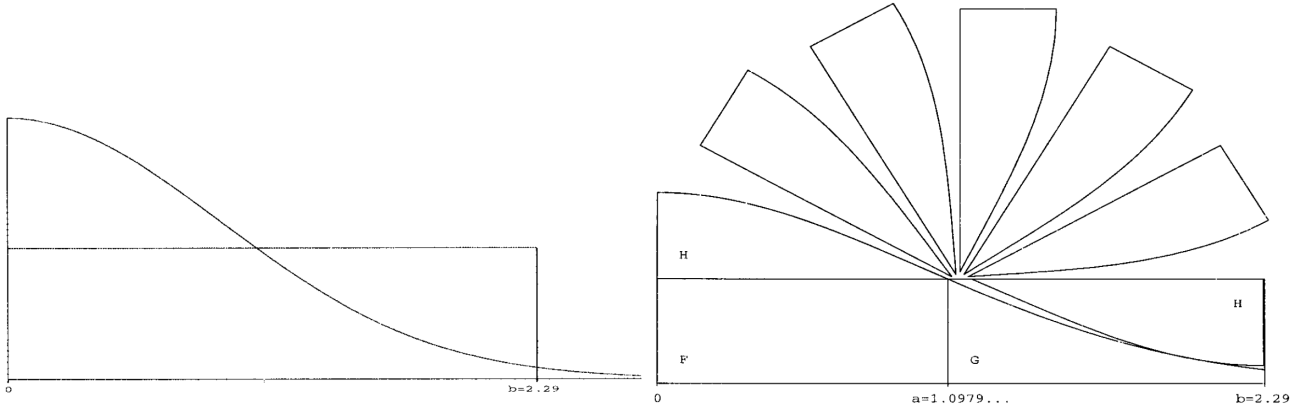
Describir el método *Monty Python* para la distribución normal y compararlo con otros métodos para la generación de valores de la normal.

1.2 El método Monty Python

El objetivo del método Monty Python es lograr generar números aleatorios cuya frecuencia se aproxime a aquella de una distribución normal. Para ello, emplea la similitud entre la expresión de la función de densidad de la distribución normal y una sigmoide decreciente, así como se aprovecha del bajo coste computacional de la generación de números aleatorios en un rectángulo.

$$N(0, 1) \sim f(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-x^2/2}$$

Tomando $x > 0$, se traza un rectángulo de base b y altura $1/b$ sobre la sigmoide. Con un giro de π radianes y un desplazamiento sobre la región exterior al rectángulo se pretende representar la mayor superficie posible de la sigmoide en el interior del rectángulo de área 1. En las imágenes siguientes, podemos observar el procedimiento descrito tomando un $b = 2.29$.



(a) Sigmoide con el rectángulo de base b . [1]

(b) Giro y desplazamiento gráficamente. [1]

Una vez descrito el área bajo la función de densidad en el interior del rectángulo, todo número aleatorio generado en ese rectángulo pertenecerá a las regiones F , G , H o a la región comprendida entre G y H correspondiente a la cola de la sigmoide. La probabilidad de obtener un número en la cola es de un 0.022 [1].

Siendo (x, y) el valor aleatorio generado y (x', y') su valor correspondiente en la sigmoide,

$$(x', y') = \begin{cases} (x, y) & \text{si } x \in F \text{ ó } G \\ (b - x, 2/b - y) & \text{si } x \in H \end{cases}$$

Si el valor aleatorio generado no pertenece a F , G ó H , pertenecerá a la cola de la distribución, por lo que basta con devolver una variante de la cola normal mediante el método de Marsaglia o el método de la cola general de Marsaglia y Tsang.

La elección del valor de b no es crítica, pero elegir un valor de b demasiado grande implica que habrá solapamiento al realizar el giro y desplazamiento descrito, mientras que si el valor de b es demasiado pequeño, será necesario realizar frecuentemente el método para hallar los valores de la cola. En caso de utilizar el método anterior para "ajustar" la sigmoide al rectángulo, la elección de $b = 2.29$ es prácticamente la máxima posible [1].

En lugar de una rotación y un desplazamiento de la región exterior al rectángulo, "estirar" la región H permite un ajuste mejor, manteniendo constante el área de la sigmoide, y por tanto siendo este un método más complejo pero más eficiente para realizar dicha aproximación.

Definiendo el factor de estiramiento s y la función de densidad de la región H , $f_H(x)$,

$$s = \frac{a}{b-a} \quad f_H(x) = f(x) - \frac{1}{b} \text{ con } 0 < x < a$$

se tiene que la región H girada y estirada tiene una función de densidad $f_{H'}(x)$.

$$f_{H'}(x) = \frac{1}{b} - s \left[f(s(b-x)) - \frac{1}{b} \right]$$

La siguiente figura representa gráficamente las transformaciones matemáticas descritas compuestas con la rotación aplicada a la sección H .

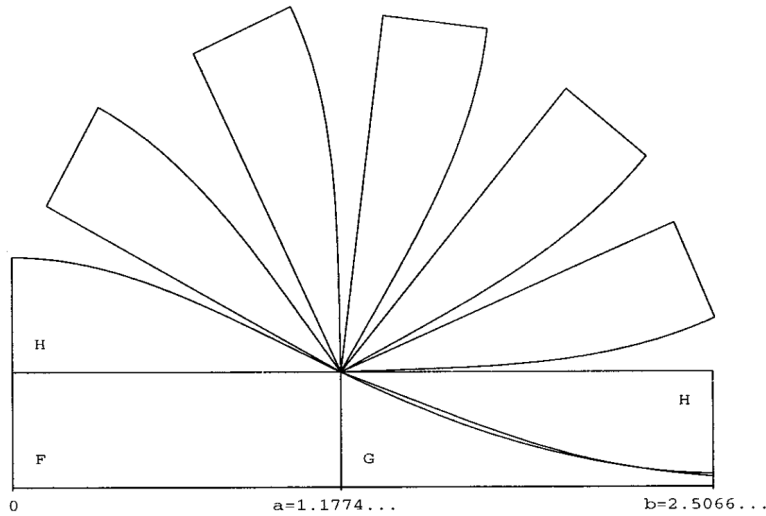


Figure 2: Descripción gráfica del estiramiento descrito. [1]

Para este segundo caso, la elección habitual para el valor de b es $b = \sqrt{2\pi}$. Esta elección no es crítica, cualquier valor entre 2.506 y 2.5074 es válido [1], pero el par $b = \sqrt{2\pi}$, $a = \sqrt{\ln 4}$ son opciones fácilmente identificables con una precisión ilimitada.

1.3 Comparación de métodos

A continuación, se compararán diferentes métodos para la generación de valores de la normal con el método Monty Python. Para ello, nos basaremos en la publicación citada [2]. Teniendo en cuenta que tanto el método generador de las secciones F , G y H , como el método generador de las colas de la sigmoide son relevantes, el artículo divide la evaluación en cuatro secciones.

- Tests de bondad de ajuste. Para evaluar los algoritmos utilizados, se utiliza el test χ^2 . El número de intervalos en los que se divide la distribución para realizar el test es k , que se define mediante la siguiente expresión, siendo n el número de muestras.

$$k = \lceil n^{3/5} \rceil$$

- Test para valores altos de sigma. Este test mide principalmente el comportamiento de los generadores en las colas de la distribución. El principal reto es la baja probabilidad de obtener muestras que se encuentren en las colas. Se utiliza por tanto un algoritmo para "forzar" generación de muestras que sean grandes múltiplos de sigma.
- Test de aleatoriedad. Para este test, se usó la batería Crush, que forma parte de los *TestU01*.
- Test de correlación interbloque. Este test mide como afecta la distribución de cada una de las muestras a la muestra siguiente, considerándose un buen resultado si la distribución de cada una de las muestras es independiente de las anteriores.

También se evalúa el número de operaciones realizadas y la velocidad de ejecución.

1.3.1 Resultados en velocidad.

La velocidad se mide relativa a la velocidad del método de Rechazo Polar. El método Monty Python es el cuarto más rápido en comparación con los 16 otros métodos de generación de variables aleatorias estudiados [2], siendo 1.61 veces más rápido que el método de Rechazo Polar. Destaca además que el número de operaciones realizadas es bajo en comparación con el método Wallace, que está por encima del método Monty Python en rapidez. Este último tiene también menos cantidad de constantes que el método Ziggurat, que está también por encima del Monty Python en cuestión de velocidad.

Nombre del método	Velocidad
Wallace (qual=1)	6.41
Ziggurat	4.29
Wallace (qual=4)	2.48
Monty Python	1.61
PPND7	1.16
Mixture-of-triangles	1.14
Polar	1.00

1.3.2 Resultados en el test de bondad de ajuste χ^2 .

En el test de bondad de ajuste χ^2 , el método Monty Python no pasa el test al utilizar muestras de valores mayores a 2^{34} . De nuevo, los métodos Wallace y Ziggurat obtienen mejores resultados, ya que no fallan al utilizar muestras de valor igual o mayor a 2^{36} . El algoritmo PPND7 falla en la misma prueba que el método Monty Python, pero este algoritmo además es 3 veces más lento.

1.3.3 Resultados en el test de valores altos de sigma.

Los resultados de este test muestran el múltiplo más alto de sigma donde el test se realiza con éxito. El método Monty Python obtiene un valor de 8.27, siendo en este caso mejor que los métodos Ziggurat y Wallace. En comparación con el resto de generadores, el Monty Python obtiene muy buenos resultados, situándose el cuarto mejor de todos los generadores comparados.

1.3.4 Resultados del test de aleatoriedad

El generador Monty Python no tuvo fallos al realizarse el test Crush.

1.3.5 Resultados del test de correlación interbloque.

Este test solo se realizó en los generadores Ziggurat, Wallace y Monty Python, dado que fueron los más rápidos y en este test se necesitan realizar pruebas con muestras muy grandes. Los generadores de Ziggurat y Monty Python pasaron todos los tests con éxito, mientras que el generador Wallace (tanto el de baja calidad como el de alta), falla cuando las iteraciones están por debajo de 8. Esto se debe a que el método Wallace es recursivo.

1.3.6 Conclusiones

Si bien es cierto que el método Wallace es el más rápido, presenta desventajas evidentes como la correlación. El método Ziggurat, que es más lento que el Wallace, no tiene problemas de correlación, pero si que falla en el test Crush con una colisión identificada en los tests de doble precisión. Además, utiliza un total de 388 constantes, lo que puede ser problemático en algunos entornos. El método Monty Python es el tercero más rápido después del Wallace y el Ziggurat, y presenta ventajas evidentes, como los buenos resultados en los tests de aleatoriedad y correlación, así como el bajo número de operaciones y constantes necesarias. La limitación del método Monty Python está en el uso de muestras cuya n sea mayor que 2^{36} .

Bibliografía

References

- [1] G. Marsaglia and W. W. Tsang, "The monty python method for generating random variables," vol. 24, no. 3, p. 341–350, sep 1998. [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/292395.292453>
- [2] D. Thomas, W. Luk, P. Leong, and J. Villasenor, "Gaussian random number generators," *ACM Comput. Surv.*, vol. 39, 11 2007.