Tema 1

Generación de números aleatorios

1.1. Introducción

Los números aleatorios son la base esencial de la simulación. Usualmente, toda la aleatoriedad involucrada en el modelo se obtiene a partir de un generador de números aleatorios que produce una sucesión de valores que supuestamente son realizaciones de una secuencia de variables aleatorias independientes e idénticamente distribuidas (i.i.d.) $U(0,1)$. Posteriormente estos números aleatorios se transforman convenientemente para simular las diferentes distribuciones de probabilidad que se requieran en el modelo. En general, la validez de los métodos de transformación dependen fuertemente de la hipótesis de que los valores de partida son realizaciones de variables aleatorias iid $U(0,1)$, pero esta suposición realmente no se cumple, puesto que los generadores de números aleatorios son simplementes programas determinísticos que intentan reproducir una sucesión de valores que parezca aleatoria.

Números pseudoaleatorios

Si decidiésemos realizar el sorteo de Navidad de Lotería Nacional mediante ordenador, seguramente la gente no confiaría en la aleatoriedad del ordenador y se quejaría. En su lugar, se prefiere un método físico y sencillo de entender, como extraer bolas de un bombo. Incluso este tipo de métodos requiere tomar ciertas precauciones: todas las bolas debe tener idéntico peso, deben de estar bien mezcladas en el bombo y se deben cambiar regularmente para reducir las posibilidades de que unas aparezcan más que otras. Claramente este procedimiento no es práctico para una simulación computacional que requiere la generación de cientos de miles de números aleatorios.

El método más conveniente y más fiable de generar números aleatorios es utilizar algoritmos determinísticos que posean alguna base matemática solida. Estos algoritmos producen una sucesión de números que se asemeja a la de una sucesión de realizaciones de variables aleatorias iid $U(0,1)$, aunque realmente no lo sea. Es por ello que este tipo de números se denominan pseudo-aleatorios y el algoritmo que los produce se llama generador de números pseudo-aleatorios.

Definición 1.1. *Un generador de números (pseudo)aleatorios es una estructura* $\mathcal{G} = (X, x_0, T, U, g)$*, donde* X *es un conjunto finito de estados,* $x_0 \in X$ *es el estado inicial (semilla), la aplicación* $T : X \to X$ *es la función de transición,* U *es el conjunto finito de posibles observaciones, y* $G : X \to U$ *es la función de salida.*

Básicamente, el funcionamiento de un generador de números pseudo-aleatorios es el siguiente. Se elige una semilla inicial cualquiera x_0, y se genera una sucesión de valores x_n mediante una relación de recurrencia $x_n = T(x_{n-1})$. Cada uno de estos valores proporciona un número pseudo-aleatorio u_n definido a través de alguna relación $u_n = g(x_n)$. Claramente, la sucesión de estados es periódica, puesto que X es finito. En algún momento, ocurrirá que $x_j = x_i$ para algún $j > i$, y a partir de ese instante,

$x_{j+k} = x_{i+k}$, y por lo tanto, $u_{j+k} = u_{i+k}$, para todo $k \geq 0$. El *periodo* es el menor entero $\rho > 0$ tal que para algún entero $\tau \geq 0$, se verifica que $x_{\rho+k} = x_k$, para todo $k \geq \tau$. Claramente, el periodo de un generador no puede exceder el cardinal del espacio de estados. Una buena propiedad para un generador es que su periodo esté cercano a $|X|$.

Un buen generador de números pseudo-aleatorios debería tener las siguientes propiedades:

- Por encima de todo, la sucesión de valores que proporcione debería asemejarse a una sucesión de realizaciones independientes de una variable aleatoria $U(0, 1)$.
- Los resultados deben ser reproducibles, en el sentido de que comenzando con las mismas condiciones iniciales debe ser capaz de reproducir la misma sucesión. Esto nos puede permitir depurar fallos del modelo o simular diferentes alternativas del modelo en las mismas condiciones obteniendo una comparación más precisa. Los procedimientos físicos no permiten que los resultados sean reproducibles.
- la sucesión de valores generados debe tener un ciclo no repetitivo tan largo como sea posible
- el generador debe ser rápido y ocupar poca memoria interna

1.2. Métodos de generación de números pseudo-aleatorios

1.2.1. Método de los cuadrados medios - Midsquare Method

Este método se debe fue propuesto en los años 40 por los matemáticos John von Neumann y Nicholas Metropolis. El método comienza tomando un número al azar, x_0,

de $2n$ cifras (originalmente los autores proponían 4 cifras) que al elevarlo al cuadrado resulta un número de hasta $4n$ cifras. Si es necesario se añaden ceros a la izquierda para que el número resultante tenga exactamente $4n$ cifras. Sea x_1 el número resultante de seleccionar las $2n$ cifras centrales de x_0^2; el primer número aleatorio u_1 se obtiene poniendo un punto decimal delante las $2n$ cifras de x_1. A continuación x_2 y u_2 se generan a partir de x_1 del mismo modo. Así sucesivamente.

Este método tiene dos inconvenientes principales:

- tiene una fuerte tendencia a degenerar a cero rápidamente (probar por ejemplo con $x_0 = 1009$)
- los números generados pueden repetirse cíclicamente después de una secuencia corta.

Ejemplo 1.2.

$$x_0 = 3708 \Rightarrow x_0^2 = 13|7492|64 \Rightarrow x_1 = 7492 \Rightarrow u_1 = 0.7492$$
$$x_1 = 7492 \Rightarrow x_1^2 = 56|1300|64 \Rightarrow x_2 = 1300 \Rightarrow u_2 = 0.1300$$
$$x_2 = 1300 \Rightarrow x_2^2 = 1|6900|00 \Rightarrow x_3 = 6900 \Rightarrow u_3 = 0.6900$$
$$x_3 = 6900 \Rightarrow x_3^2 = 47|6100|00 \Rightarrow x_4 = 6100 \Rightarrow u_4 = 0.6100$$
$$x_4 = 6100 \Rightarrow x_4^2 = 47|2100|00 \Rightarrow x_5 = 2100 \Rightarrow u_5 = 0.2100$$
$$x_5 = 2100 \Rightarrow x_5^2 = 4|4100|00 \Rightarrow x_6 = 4100 \Rightarrow u_6 = 0.4100$$
$$x_6 = 4100 \Rightarrow x_6^2 = 16|8100|00 \Rightarrow x_7 = 8100 \Rightarrow u_7 = 0.8100$$
$$x_7 = 8100 \Rightarrow x_7^2 = 65|6100|00 \Rightarrow x_8 = 6100 \Rightarrow u_8 = 0.6100$$

1.2.2. Métodos congruenciales

Los principales generadores de números pseudo-aleatorios utilizados hoy en día son los llamados generadores congruenciales lineales, introducidos por Lehmer en 1951. Un método congruencial comienza con un valor inicial (semilla) x_0, y los sucesivos valores $x_n, n \geq 1$ se obtienen recursivamente con la siguiente fórmula:

$$x_n = ax_{n-1} + b \text{ módulo } m,$$

donde a, m y b son enteros positivos que se denominan, respectivamente, el multiplicador, el módulo y el incremento. Si $b = 0$, el generador se denomina multiplicativo; en caso contrario se llama mixto. La sucesión de números pseudo-aleatorios $u_n, n \geq 1$ se obtiene haciendo $u_i = \frac{x_i}{m}$.

Como el siguiente resultado demuestra, cada x_i está completamente caracterizado por a,b,m y x_0.

Proposición 1.3. *Los valores generados por un método congruencial verifican:*

$$x_n = a^n x_0 + b\frac{a^n - 1}{a - 1} \mod m$$

Demostración.

Para $n = 1$, tenemos que $x_1 = ax_0 + b \mod m$, lo que implica que existe $k \in \mathbb{Z}_+$ tal que $km + x_1 = ax_0 + b$. Ahora, para $n = 2$, se tiene que

$$\begin{aligned} x_2 &= ax_1 + b \mod m \\ &= a(ax_0 + b - km) + b \mod m \\ &= a^2x_0 + b(a + 1) - akm \mod m \\ &= a^2x_0 + b(a + 1) \mod m. \end{aligned}$$

Por recurrencia, se tiene que

$$\begin{aligned} x_3 &= a^3 x_0 + b(a^2 + a + 1) \mod m \\ &\dots \\ x_n &= a^n x_0 + b(a^n - 1 + \dots + a + 1) \mod m \\ &= a^n x_0 + b\frac{a^n - 1}{a - 1} \mod m \end{aligned}$$

■

Por lo tanto, la primera objeción que se le puede hacer a este método, objeción común a todos los generadores de números pseudo-aleatorios, es que la sucesión de los valores x_n no es en absoluto aleatoria. Sin embargo, posteriormente veremos que si elegimos las parámetros iniciales convenientemente, la sucesión $\{u_n\}$ puede asemejarse a una sucesión de números aleatorios.

La segunda objeción es que los valores u_i pueden tomar sólo los valores $0, \frac{1}{m}, \frac{2}{m}, \dots, \frac{m-1}{m}$, luego no hay posibilidad de generar un valor, por ejemplo entre $\frac{2}{m}$ y $\frac{3}{m}$. Tomando m suficientemente grande $m \geq 10^9$, el conjunto de posibles valores es suficientemente denso en $[0, 1]$ como para que la sucesión asemeje a la de una variable continua uniforme en dicho intervalo.

Ejemplo 1.4. Consideremos el generador congruencial $x_n = 5x_{n-1} + 1 \mod 9$ con $x_0 = 1$.

n	x_n	u_n	n	x_n	u_n	n	x_n	u_n
0	1	-	2	3	0.333	4	0	0
1	6	0.6666	3	7	0.7777	5	1	0.1111

Ejemplo 1.5. Consideremos el generador congruencial $x_n = 5x_{n-1} + 3 \mod 16$, con $x_0 = 7$.

n	x_n	u_n	n	x_n	u_n	n	x_n	u_n	n	x_n	u_n
0	7	-	5	10	0.625	10	9	0.563	15	4	0.25
1	6	0.375	6	5	0.3125	11	0	0	16	7	0.4375
2	1	0.0625	7	12	0.75	12	3	0.1875	17	6	0.375
3	8	0.5	8	15	0.9375	13	2	0.125	18	1	0.0625
4	11	0.6875	9	14	0.875	14	13	0.8125	19	8	0.5

Se observa que el generador del ejemplo 1.4 no es adecuado, pues produce un ciclo de longitud $6 < m = 9$ (la longitud de ciclo de un operador se denomina periodo). Sin embargo, en el generador del ejemplo 1.5 la longitud de ciclo coincide exactamente con el módulo, lo cuál es inevitable, pues es evidente que el periodo nunca puede exceder al modulo. En este caso, se dice que el generador es de ciclo completo. Obsérvese que un generador sea de ciclo completo es independiente de la semilla que se utilice; sea cual sea $x_0 \in \{1, \ldots, m\}$, el ciclo siempre tendrá periodo m y se producirá en el mismo orden. Sin embargo, si un generador no es de ciclo completo, la longitud de ciclo puede depender de la semilla utilizada. Si en el ejemplo 1.4 utilizamos $x_0 = 5$ o $x_0 = 8$, el periodo resultante es 2.

De los ejemplos anteriores se desprende que una cuestión de interés es cómo elegir los parámetros del generador de forma que este tenga ciclo completo. El siguiente teorema, propuesto por Hull y Dobell (1962) proporciona un caracterización en este sentido.

Teorema 1.6. *Un generador congruencial tiene periodo completo si y sólo si se cumplen las siguientes condiciones:*

1. *m y b son primos entre sí*
2. *Si q es un número primo que divide a m, entonces q divide a $a - 1$.*
3. *Si 4 divide a m, entonces 4 divide a $a - 1$.*

Corolario 1.7. *Un generador congruencial multiplicativo no puede tener periodo completo*

Es evidente que es necesario que m sea grande con el fin de un periodo largo y una alta densidad en el intervalo $[0, 1]$. Sin embargo, la operación de dividir por m y calcular el resto es relativamente lenta. Una elección de m adecuada computacionalmente es $m = 2^k$, donde k-bits es el tamaño de palabra (unidad básica de trabajo) del microprocesador. El hecho de que esta elección sea ventajosa reside en que nos podemos aprovechar del desbordamiento de datos (integer overflow) para no tener que realizar la operación del generador explícitamente. La cuestión es que si tenemos una máquina de k-bits, entonces el mayor entero que puede ser representado es $2^k - 1$ y en cualquier intento de representar un entero mayor H, que ocuparía $h > k$ dígitos binarios, se perderían los $h - k$ dígitos binarios más a la izquierda, y los k dígitos que quedan se corresponden precisamente con $H \mod 2^k$.

Ejemplo 1.8. Para ilustrar como aprovechar el desbordamiento de datos en el generador del ejemplo 1.5, veamos cómo generar $x_{10} = 9$ a partir de $x_9 = 14$.

Observemos que $5x_{14} + 3 = 73$, que en binario es 1001001. Como nuestra capacidad es de 4 bits, el desbordamiento de datos produce que se pierdan los tres primeros dígitos, quedando 1001 que es la representación binaria de $x_{10} = 9$.

Corolario 1.9. *Un generador congruencial con $m = 2^k \geq 4$ tiene periodo completo si y sólo si b es impar y $1 = a \mod 4$.*

Los generadores multiplicativos tienen la ventaja computacional de que no es necesario la suma de la constante b, pero, como hemos visto, no pueden tener periodo completo. Afortunadamente, es posible obtener periodo $m - 1$ si m y a se eligen adecuadamente. Los generadores multiplicativos se introdujeron antes que los mixtos, han sido objeto de más estudio y actualmente se utilizan más que los mixtos.

Al igual que los generadores mixtos, la mejor forma de elegir el modulo desde el punto de vista computacional es tomar $m = 2^k$. Sin embargo, en este caso, el periodo máximo posible será una cuarta parte del modulo.

Proposición 1.10. *El periodo máximo de un generador congruencial multiplicativo con* $m = 2^k \geq 16$ *es* 2^{k-2}. *Dicho periodo maximal se alcance si y sólo si* x_0 *es impar y* $3 = a \mod 8$ *o* $5 = a \mod 8$

Un generador multiplicativo muy utilizado, conocido como RANDU, tomaba $m = 2^{31}$ y $a = 2^{16} + 3$. Sin embargo, se ha demostrado que tiene propiedades estadísticas bastante malas.

Debido a los problemas que surgen de elegir $m = 2^k$ en los generadores multiplicativos, se han investigado otras alternativas para el valor de m. La más satisfactoria es la propuesta de Hutchinson en 1966, que consiste en tomar como m el mayor primo menor que 2^k. Por ejemplo, si $k = 31$, el mayor primo menor que 2^{31} es $2^{31} - 1$. A este respecto se tiene el siguiente resultado.

Teorema 1.11. *Sea* t *la longitud de un ciclo maximal de un generador congruencial multiplicativo. Se verifica que:*

1. *Si* $t = m - 1$, *entonces* m *es primo*
2. *Si* m *es primo, entonces* t *divide a* $m - 1$.
3. *Si* m *es primo, entonces* $t = m - 1$ *si y sólo si* a *es una raíz primitiva de* m *(*a *es una raíz primitiva de* m *si* $a \neq 0$ *y no existe ningún factor primo* p *de* $m - 1$ *tal que* $1 = a^{\frac{m-1}{p}} \mod m$*)*

Los generadores multiplicativos más famosos utilizados por IBM tomaban $m = 2^{31} - 1$ y $a_1 = 7^5$ o $a_2 = 630360016$.

Ejemplo 1.12. Consideremos un generador congruencial multiplicativo con $m = 31$. Encontrar los periodos correspondientes a los multiplicadores $a_1 = 3$, $a_2 = 4$.

Como $m = 31$ es primo, entonces la longitud maximal de ciclo t debe dividir a $m - 1 = 30 = 2 \cdot 3 \cdot 5$. Luego t puede valer 1, 2, 3, 5, 6, 10, 15 y 30.

- $a_1 = 3$. Veamos si 3 es raíz primitiva de 31. Para ello hay que comprobar que no se verifican ninguna de las siguientes relaciones: $1 = 3^{15} \mod 31$, $1 = 3^{10} \mod 31$ y $1 = 3^6 \mod 31$.

  - $3^{15} \mod 31 = 14348907 \mod 31 = 30 \neq 1$
  - $3^{10} \mod 31 = 59049 \mod 31 = 25 \neq 1$
  - $3^6 \mod 31 = 729 \mod 31 = 16 \neq 1$

  Por lo tanto, aplicando el tercer supuesto del teorema 1.11 se sigue que el generador tiene ciclo maxima de longitud 30.

- $a_1 = 3$. Veamos si 4 es raíz primitiva de 31. Para ello hay que comprobar que no se verifican ninguna de las siguientes relaciones: $1 = 4^{15} \mod 31$, $1 = 4^{10} \mod 31$ y $1 = 4^6 \mod 31$.

  - $4^{15} \mod 31 = 1073741824 \mod 31 = 1$

  Por lo tanto, aplicando el tercer supuesto del teorema 1.11 se sigue que el generador no tiene ciclo máximo de longitud 30. Se puede comprobar que en este caso hay seis ciclos de longitud 5 y uno de longitud 1.

1.2.3. Combinación de algoritmos congruenciales

Los algoritmos congruenciales se pueden combinar para aumentar el periodo del ciclo de generación. Estas combinaciones se basan en los siguientes resultados:

- Si $U_1, \ldots U_k$ son variables aleatorias iid $\mathcal{U}(0,1)$, entonces la parte fraccional de $U_1 + \ldots U_k$ también sigue una distribución $\mathcal{U}(0,1)$

$$U_1 + U_2 + \cdots + U_k - [U_1 + U_2 + \ldots U_k] \sim \mathcal{U}(0,1)$$

- Si $u_1, u_2, \ldots, u_k$ están generados por algoritmos congruenciales con ciclos de periodo $c_1, c_2, \ldots, c_k$, respectivamente, entonces la parte fraccional de $u_1 + u_2 + \cdots + u_k$ tiene un ciclo de periodo m.c.m.$\{c_1, c_2, \ldots, c_k\}$.

El algoritmo combinado de Wichmann y Hill (1982,1984) tiene un periodo de orden 10^{12}. El generador es:

$$\begin{aligned} x_i &\equiv 171 x_{i-1} \mod 30269 \\ y_i &\equiv 172 y_{i-1} \mod 30307 \\ z_i &\equiv 170 z_{i-1} \mod 30323 \end{aligned}$$

y tomar

$$u_i = \left(\frac{x_i}{30269} + \frac{y_i}{30307} + \frac{z_i}{30323}\right) - \left[\frac{x_i}{30269} + \frac{y_i}{30307} + \frac{z_i}{30323}\right]$$

1.2.4. Otros generadores más complicados

Aunque los generadores congruenciales son los más utilizados en la práctica, se han desarrollado otros tipos de generadores con la intención de obtener periodos más largos y mejores propiedades estadísticas. A menudo, sin embargo, un generador congruencial con parámetros elegidos adecuadamente puede funcionar tan bien que otras alternativas más complicadas.

Los generadores congruenciales pueden generalizarse a recursiones lineales de orden mayor, considerando la siguiente relación

$$x_n = (a_1 x_{n-1} + \cdots + a_k x_{n-k}) \bmod m,$$

donde el orden k y el módulo m son enteros positivos, y los coeficientes a_j son enteros variando entre $-(m-1)$ y $(m-1)$. En la n-ésima iteración, el estado es el vector $(x_n, x_{n-1}, \ldots, x_{n+k-1}) \in \mathbb{Z}_m^k$. La función de salida se puede definir simplemente como $u_n = \frac{x_n}{m}$.

El estudio de este generador se asocia al estudio del polinomio característico

$$P(z) = z^k - a_1 z^{k-1} - \cdots - a_k$$

sobre el cuerpo finito $\mathbb{Z}_m$. Cuando m es primo y el polinomio es primitivo sobre $\mathbb{Z}_m$, el periodo del generador es $m^k - 1$ (periodo máximo posible en esta clase de generadores).

1.3. Tests para la comprobación de la uniformidad y la aleatoriedad

1.3.1. Comprobación de la uniformidad

A continuación, se muestran algunos procedimientos para verificar la uniformidad de los números aleatorios generados por algún método. Se trata de decidir si los números generados se pueden considerar como una realización de una muestra aleatoria simple de una distribución $\mathcal{U}(0,1)$.

1.3.1.1. Contraste de Kolmogorov-Smirnov

El test de K-S es un test de bondad de ajuste que se utiliza para determinar si los datos de una determinada muestra se ajustan a una hipotética distribución.

Dada una muestra aleatoria simple $x_1, \ldots, x_n$, la función de distribución empírica de la muestra es

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{x_i \leq x} = \frac{|\text{valores } x_i \leq x|}{n}$$

.

Las hipótesis son:

$$H_0 : F_n = F_0 \qquad\qquad H_1 : F_n \neq F_0,$$

donde F_0 es la hipotética función de distribución. El estadístico de contraste para el test K-S es

$$D_n = \sup_{x \in \mathbb{R}} |F_n(x) - F_0(x)|.$$

La distribución de D_n está tabulada. La región crítica del contraste es $[D_{n,\alpha}, +\infty)$, donde $D_{n,\alpha}$ es el percentil de orden $1-\alpha$ de la distribución de D_n.

En nuestro caso particular, F_0 es la función de distribución de una v.a. $\mathcal{U}(0,1)$, y por lo tanto, se puede comprobar que el estadístico de Kolmogorov-Smirnov para contrastar la uniformidad de la muestra $u_1, \ldots, u_n$ viene dado por:

$$D_n = \max_{1 \leq i \leq n} \left\{ \text{máx} \left\{ \left| \frac{i}{n} - u_{(i)} \right|, \left| \frac{i-1}{n} - u_{(i)} \right| \right\} \right\}$$

siendo $u_{(i)}$ el i-ésimo menor valor de la muestra.

1.3.1.2. Test de la χ^2

Dada la muestra $u_1, \ldots, u_n$ y un nivel de significación α, el test consiste en los siguientes pasos:

1. Dividir el intervalo $(0,1)$ en k clases disjuntas de la misma amplitud, $\frac{1}{k}$. Para cada clase C_j, contar el número de elementos O_j que cae en dicha clase.

2. Comparamos las frecuencias observadas en cada clase con las que corresponderían según la distribución teórica. Se considera el estadístico:

$$T = \sum_{j=1}^{k} \frac{\left(O_j - \frac{n}{k}\right)^2}{\frac{n}{k}}.$$

   Se demuestra que para n grande, T sigue una distribución χ^2 con $k-1$ grados de libertad.

3. Se rechaza la hipótesis de uniformidad si $T > \chi^2_{k-1,\alpha}$, donde $\chi^2_{k-1,\alpha}$ es el percentil de orden $1-\alpha$ de la distribución χ^2_{k-1} .

1.3.1.3. Contraste de los pares consecutivos no solapados

Dada la muestra $u_1, \ldots, u_n$, n par, y un nivel de significación α, el test consiste en los siguientes pasos:

1. Dividir el intervalo $(0,1)$ en k clases disjuntas de la misma amplitud, $\frac{1}{k}$.

2. Categorizamos la muestra $u_1, \ldots u_n$ asociando a cada u_i el índice de la clase a la que pertenece. Sea $y_1, \ldots, y_n$ la muestra discretizada ($y_i = j \Leftrightarrow u_i \in C_j$).

3. Agrupamos los valores de la muestra discretizada en pares consecutivos no solapados

$$(y_1, y_2), (y_3, y_4), \ldots (y_{n-1}, y_n)$$

4. Sea O_{ij} el número de veces que aparece el par (i,j). Si la hipótesis de uniformidad es cierta, entonces $O_{ij} \sim \text{Bi}\left(\frac{n}{2}, p = \frac{1}{k^2}\right)$. Por tanto, $E[O_{ij}] = E_{ij} = \frac{n}{2m^2}$.

5. Se evalúa el estadístico de contraste

$$T = \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{(O_i j - E_{ij})^2}{E_{ij}}.$$

Se demuestra que para n grande, T sigue una distribución χ^2 con $m^2 - 1$ grados de libertad.

6. Se rechaza la hipótesis de uniformidad si $T > \chi^2_{m^2-1,\alpha}$, donde $\chi^2_{m^2-1,\alpha}$ es el percentil de orden $1 - \alpha$ de la distribución $\chi^2_{m^2-1}$.

1.3.2. Contraste de aleatoriedad

A continuación, mostraremos un procedimiento para comprobar la aleatoriedad de la muestra de numeros generados.

1.3.2.1. Test de rachas

Dada la muestra $u_1, \ldots, u_n$ se construye una sucesión formada por 0 y 1 del siguiente modo: en la posición i se coloca un 0 si $x_{i+1} < x_i$ y se coloca un 1 si $x_{i+1} > x_i$. Cada grupo consecutivo se 0's o 1's se denomina racha. Se demuestra que para n suficientemente grande, el número de rachas R sigue una distribución normal

$$R \sim \mathcal{N}\left(\frac{2n-1}{3}, \sigma^2 = \frac{16n-29}{90}\right)$$

El estadístico de contraste sería

$$Z = \frac{R - \frac{2n-1}{3}}{\sqrt{\frac{16n-29}{90}}}$$

que sigue una distribución $\mathcal{N}(0,1)$.

Para un nivel de significación α, se rechaza la hipótesis de aleatoriedad si $|Z| > Z_{\alpha/2}$.

1.4. Simulación Montecarlo

Se entiende por simulación Montecarlo al uso de números aleatorios (realizaciones de variables $\mathcal{U}(0,1)$ para resolver ciertos problemas estocásticos o determinísticos en los que el paso del tiempo no juega un papel esencial (problemas de tipo estático).

1.4.1. Cálculo de series

Supongamos que deseamos calcular $\theta = \sum_{x\in D} g(x)$. Consideremos X una variable aleatoria discreta con soporte D y función puntual de probabilidad $p(x)$. Entonces,

$$\sum_{x\in D} g(x) = \sum_{x\in D} \frac{g(x)}{p(x)} p(x) = E\left(\frac{g(X)}{p(X)}\right).$$

Si $\{X_i\}_{i=1}^{\infty}$ es una sucesión de variables aleatorias iid con función puntual de probabilidad $p(x)$, entonces $\left\{\frac{g(X_i)}{p(X_i)}\right\}_{i=1}^{\infty}$ es una sucesión de variables aleatorias con media θ. Por la ley de los grandes números, se tiene que

$$\frac{1}{k}\sum_{i=1}^{k} \frac{g(X_i)}{p(X_i)} \to \theta \qquad \text{cuando } k \to \infty$$

Consecuentemente, simulando una muestra $(x_1, \ldots, x_k)$ de la v.a. X, podemos estimar θ mediante $\frac{1}{k}\sum_{i=1}^{k} \frac{g(x_i)}{p(x_i)}$.

1.4.2. Aproximación de integrales

1.4.2.1. Método 1. General

Deseamos estimar el valor de una integral $\theta = \int_C g(x)dx$. Siguiendo un esquema similar al de la estimación de series podemos considerar una variable aleatoria continua

con soporte en C y función de densidad $f(x)$. Entonces,

$$\theta = \int_C g(x)dx = \int_C \frac{g(x)}{f(x)} f(x)dx = E\left(\frac{g(X)}{f(X)}\right).$$

De nuevo, como aplicación de la ley de los grandes números, se tiene que simulando una muestra $(x_1, \ldots, x_k)$ de la v.a. X, podemos estimar θ mediante $\frac{1}{k}\sum_{i=1}^{k} \frac{g(x_i)}{f(x_i)}$.

1.4.2.2. Método 2. Utilización de números aleatorios

Una de las primeras aplicaciones de los números aleatorios fue la aproximación de integrales definidas. Supongamos que deseamos obtener

$$\theta = \int_0^1 g(x)dx$$

Para obtener el valor de θ. podemos observar que si $U \sim \mathcal{U}(0,1)$, entonces podemos expresar $\theta = E[g(U)]$. Si $U_1, \ldots, U_k$ son v.a. iid $\mathcal{U}(0,1)$, se sigue que las variables aleatorias $g(U_1), \ldots, g(U_k)$ son iid con media θ. Por lo tanto, por la ley fuerte de los grandes numeros, se sigue que

$$\sum_{i=1}^{k} \frac{g(U_i)}{k} \to E[g(U)] = \theta \qquad \text{cuando } k \to \infty$$

Consecuentemente, podemos aproximar θ generando una cantidad suficientemente grande de números aleatorios u_i y tomando el promedio de los valores $g(u_i)$.

Si deseamos calcular

$$\theta = \int_a^b g(x)dx,$$

basta hacer el cambio de variable $y = \frac{x-a}{b-a}$, con lo cual

$$\theta = \int_0^1 (b-a)g(a + [b-a]y]dy = \int_0^1 h(y)dy.$$

donde $h(y) = (b-a)g(a + [b-a]y]$.

Del mismo modo, si deseamos aproximar

$$\theta = \int_0^{\infty} g(x)dx,$$

basta hacer el cambio de variable $y = \frac{1}{x+1}$, con lo cual

$$\theta = \int_0^1 h(y)dy.$$

donde $h(y) = \frac{g\left(\frac{1}{y}-1\right)}{y^2}$.

1.4.2.3. Método 3. Acierto o fallo

Supongamos que deseamos calcular $\int_a^b g(x)dx$, donde la función g está acotada en (a, b). El problema se puede transformar equivalentemente en calcular la integral $\int_0^1 g(x)dx$, donde la función g verifica $0 \leq g(x) \leq 1$, para todo $x \in (0, 1)$. El valor de dicha integral es el área de superficie encerrada por la curva dentro del cuadrado $[0, 1] \times [0, 1]$, lo que equivale a calcular $P((X, Y) \in A)$, donde (X, Y) es una variable aleatoria bidimensional distribuida uniformemente en el cuadrado unidad (equivalentemente, X e Y variables $\mathcal{U}(0, 1)$ independientes) y A es el conjunto de puntos (x, y) del cuadrado unidad tales que $y < g(x)$.

Un método para estimar la probabilidad de dicho suceso es simular una muestra $\{(x_i, y_i)\}_{i=1}^n$ de puntos de cuadrado unidad y calcular la proporción muestral de los que pertenecen al conjunto A.