Simulación Estocástica

Generación de números aleatorios

Vanda Inácio de Carvalho

Primer Semestre 2015

Introducción

- Grande parte de este curso estará dedicada al estudio de modelos estadísticos utilizando simulaciones hechas en un computador.
- Usamos el termino 'modelo estadístico' para designar cualquier modelo matemático que incluya una componente aleatoria.
- Nuestro interés, ahora, es la simulación de tal componente aleatoria.
- El nucleo basico de tales simulaciones es la capacidad de generar números aleatorios en un computador.
- Vamos a dividir el problema de generación de números aleatorios en dos subproblemas distintos
  1. generación de números aleatorios independientes y uniformemente distribuídos en el intervalo $(0, 1)$;
  2. generación de números aleatorios de diferentes distribuciones, usando para eso números aleatorios independientes y uniformemente distribuídos en el intervalo $(0, 1)$.

Generación de números seudo aleatorios

- Hay, fundamentalmente, dos clases de métodos distintos para generar números aleatorios

  (a) Verdaderos números aleatorios son generados usando algún dispositivo físico. Ejemplos clásicos incluyen tirar una moneda o lanzar un dado. Todavía, las simulaciones modernas requieren cientos de miles o millones de números aleatorios, por lo que estos métodos suelen ser demasiado lentos para ser de uso práctico;

  (b) números seudo aleatorios son generados por los computadores. Si bien estos métodos son rápidos, un desafío con este enfoque es que los programas de computador son inherentemente determinísticos y, por lo tanto, no pueden producir output verdaderamente aleatorio.

Generación de números seudo aleatorios

Generador congluencial lineal

- El generador congluencial lineal es un ejemplo sencillo de un generador de números seudo aleatorios.
- Aún que este generador ya no sea de importancia práctica, comparte características importantes con los generadores más complicados usados en la práctica hoy en día y lo estudiaremos como un ejemplo accesible.

Algoritmo

$$X_{n+1} = (aX_n + c) \mod m, \quad n \geqslant 1$$

donde

- $m > 1$ *es el modulo,*
- $a \in \{1, 2, \ldots, m-1\}$ *es el multiplicador,*
- $c \in \{0, 1, \ldots, m-1\}$ *es el incremento,*
- $X_1 \in \{0, 1, \ldots, m-1\}$ *es la semilla.*

El output es una secuencia $X_1, X_2, X_3, \ldots$ *de numeros seudo aleatorios. Para converter al interval* $(0, 1)$, *hacer* $U_n = X_n/m$.

Generación de números seudo aleatorios

Generador congluencial lineal

- $\mod$ denota el operador modulo (o modulus).
- $n \mod m$ es el resto de la división entera de n por m. Por ejemplo, $8 \mod 5$ es igual o 'congruente' a 3.
- Por la definición del operador mod, $n \mod m \in \{0, 1, \ldots, m-1\}$.
- Así, la secuencia obtenida mediante el uso del generador congruencial lineal consiste de números enteros X_n en el rango $\{0, 1, \ldots, m-1\}$.
- El output (es decir, la secuencia generada) depende de los parámetros m, a, c, y de la semilla X_1.
- Veremos que si m, a y c se eligen cuidadosamente, la secuencia resultante se comporta de manera similar a una secuencia de variables independientes y uniformemente distribuídas.
- Escogiendo diferentes valores para la semilla X_1, diferentes secuencia de números seudo aleatorios serán obtenidas.

Generación de números seudo aleatorios

Generador congluencial lineal

Ejemplo

Para los siguientes pará metros $m = 8$, $a = 5$, $c = 1$, *y semilla* $X_1 = 0$, *el generador produce la siguiente secuencia*

n	$\mathbf{X_n}$
1	*0*
2	*1*
3	*6*
4	*7*
5	*4*
6	*5*
7	*2*
8	*3*
9	*0*
10	*1*
11	*6*

- En R, la secuencia anterior puede ser obtenida usando, por ejemplo, el siguiente código

```
a=5; c=1; m=8; n=11
x=numeric(n); x[1]=0
for(i in 1:(n-1)){
x[i+1]=(a*x[i]+c)%%m
}
u=x/m
```

- El output del generador congruencial lineal aunque parezca aleatorio, dado la manera como es generado, esta claro que tiene varias propiedades que lo hacen diferente de secuencias verdaderamente aleatorias.

- Por ejemplo, dado que cada nuevo valor X_{n+1} es calculado a partir de X_n, una vez que la secuencia generada alcanza un valor X_n que ha sido generado antes, el output se empieza a repetir.

- En el ejemplo, esto sucede para $X_1 = X_9$ y obtenemos $X_{10} = X_2$, $X_{11} = X_5$, etc.

- Dado que X_n puede sólo tomar m valores diferentes, el output se empieza a repetir después de un máximo de m pasos; la secuencia es entonces periódica, con período máximo m.

- En el ejemplo anterior la secuencia generada tiene, portanto, el máximo período posible, 8, para la configuración de parámetros dada.

- Ni siempre es así, como ilustra el próximo ejemplo.

Ejemplo

Parámetros: $m = 16$, $a = 11$, $c = 0$, *y* $X_1 = 1$. *Secuencia generada*

n	$\mathbf{X_n}$
1	*1*
2	*11*
3	*9*
4	*3*
5	*1*

La secuencia tiene período 4, aún que el período máximo posible seria de 16.

- Por veces, la periodicidad de una secuencia de números seudo aleatorios puede causar problemas, pero, por otra parte, si el tamaño del período es superior a la cantidad de números aleatorios que necesitamos, entonces la periodicidad no afecta el nuestro resultado.
- Por esta razón, es necesário elegir cuidadosamente los parámetros m, a y c a fin de lograr un período suficientemente largo.
- Un criterio para la elección de m, a, y c está dado en el siguiente teorema.

Generación de números seudo aleatorios

Generador congluencial lineal

Teorema

El generador congruencial lineal tiene período m, si y solo si las siguientes tres condiciones están satisfechas:

(a) *m y c tienen máximo común divisor de 1 (m y c son relativamente primos).*

(b) *Si m es multiplo de 4, entonces* $a - 1$ *también es multiplo de 4.*

(c) *Si q es un numero primo que divide m, entonces q también divide* $a - 1$.

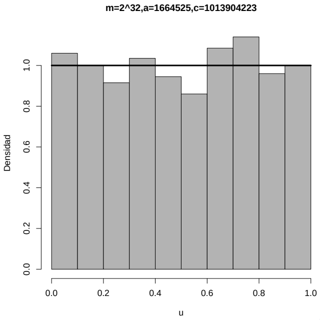
En estas condiciones, el período no depende de la semilla X_1.

- Se puede ver que $m = 8$, $a = 5$, y $c = 1$, en el primero ejemplo satisfacen las tres condiciones mencionadas en el teorema.
- Ya en el segundo ejemplo, al menos la condición (b) no está satisfecha, ya que $m = 16$ es multiplo de 4 y $a - 1 = 10$ no lo es. Así, la secuencia generada no tiene período máximo. Además, el período de dicha secuencia puede cambiar con la semilla.

Generación de números seudo aleatorios

Evaluación de la calidad de los generadores de números seudo aleatorios

- Iremos ahora ver criterios que cualquier generador de números seudo aleatorios debe satisfacer y ejemplificaremos usando el generador congruencial lineal.
- Se supone que los generadores generan números seudo aleatorios independientes y uniformemente distribuídos en el intervalo $(0, 1)$.
- El requisito más básico es el histograma asemejarse a la densidad de una variable aleatoria $U(0, 1)$.



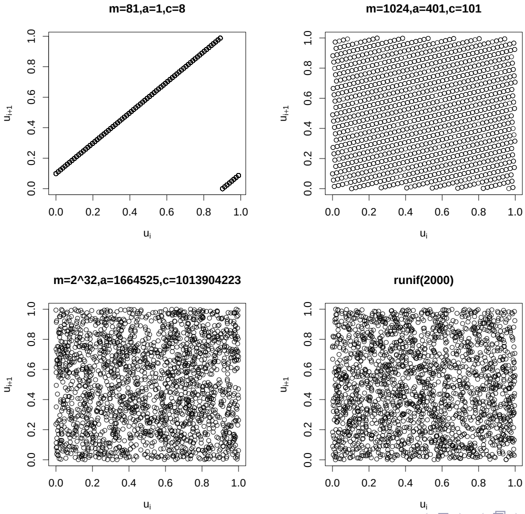
- El supuesto de uniformidad de la secuencia generada puede ser probado más formalmente usando testes estadísticos como el test del chi-cuadrado o el test de Kolmogorov–Smirnov.

- Otro aspecto de la calidad de los generadores de números seudo aleatorios es la posibilidad de dependencia entre valores consecutivos.

- Por ejemplo, en el generador congruencial lineal cada valor generado es una función deterministica del valor anterior y así valores consecutivos son claramente dependientes.

- En cierta medida, este problema es compartido por todos los generadores de números seudo aleatorios.

- Una manera fácil de visualizar la dependencia entre pares de valores consecutivos es hacer un grafico de dispersión de los puntos (U_i, U_{i+1}), $i = 1, 2, \ldots, n-1$.


m=81,a=1,c=8
m=1024,a=401,c=101
m=2^32,a=1664525,c=1013904223
runif(2000)
u_{i+1}
u_i
0.0
0.2
0.4
0.6
0.8
1.0

- En las figuras anteriores se mostra los grficos de dispersión de los puntos (U_i, U_{i+1}), $i = 1, 2, \ldots, n-1$ para diferentes configuraciones de m, a, y c.
- En las dos figuras de la primera fila claramente se observa una fuerte dependencia entre valores consecutivos.
- El la primera figura de la segunda fila tal dependencia ya no es visible, aún que dada la forma como los valores fueran generados, hay siempre alguna dependencia.
- Sin embargo, al comparar esta figura con la segunda figura de la segunda fila, la cual fue obtenida usando el generador de uniformes de `R`, se ve un comportamiento similar.