

congruencia-lineal

July 19, 2020

1 Aplicar la congruencia lineal para generar numeros randomicos

1.0.1 Alumno: Hernan Leon

1.0.2 Método de congruencias lineales [1]:

1.0.3 Pasos

1. Elegir una semilla x_0

2. Emplear la formula recursiva:

$$##### ** x_n = x_n = (a * x_0 + b) \bmod m$$

3. El numero pseudoaleatorio se obtiene con la siguiente formula

$$##### u_i = x_i / m$$

- Los elementos que intervienen en la fórmula de generación son los siguientes:
 - a es el multiplicador y es un entero positivo mayor que 0
 - m es un entero positivo y es el módulo
 - b es un entero positivo y representa el incremento.
- Si b toma un valor de 0, el generador se denomina multiplicativo, caso contrario se llama mixto.

Cómo elegir los parámetros:

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:

- a. Un generador congruencial multiplicativo no puede tener periodo completo.
- b. Un generador congruencial tiene periodo completo si y solo si b es impar

Numeros primos relativos entre si Por ejemplo, 6 y 35 son primos entre sí, al igual que, 7 y 17. En matemáticas, dos números enteros a y b son números primos entre sí (o coprimos, o primos relativos), si no tienen ningún factor primo en común, o, dicho de otra manera, si no tienen otro divisor común más que 1 y -1

```
[1]: import pandas as pd
import math
```

```
[35]: cantidad = 50
df = pd.DataFrame(index = range(cantidad+1), columns = ['Xn','Ui'])
a = 25214903917
b = 11 # o tambien c en Wikipedia
m = int(math.pow(2,48) - 1)
x0 = 342543 # semilla

df.iloc[0] = [x0,'--']

for i in range(cantidad):
    xn = (a*x0 + b) % m;
    ui = xn/m
    df.iloc[i+1] = [xn,ui]
    x0 = xn
df
```

```
[35]:
```

	Xn	Ui
0	342543	--
1	192939531121292	0.685459
2	253563094424980	0.900837
3	238493025723096	0.847297
4	62906473965458	0.223489
5	156920393682007	0.557493
6	157311748936500	0.558884
7	97396617953051	0.346022
8	206786124796738	0.734652
9	133042025808192	0.47266
10	99347644205240	0.352954
11	51559102895626	0.183175
12	27697493107953	0.0984013
13	170373948491702	0.60529
14	56589262792675	0.201045
15	53159919284616	0.188862
16	225782399071898	0.80214
17	35728602197152	0.126933
18	214349467583955	0.761522
19	76309580376446	0.271106
20	96409372451788	0.342515
21	194543075033262	0.691156
22	211610535301505	0.751792
23	146754659671036	0.521377
24	56873716312263	0.202056
25	267501501430937	0.950356
26	14180883089440	0.0503806

27	67007721298116	0.238059
28	255813127072808	0.908831
29	30486111888187	0.108308
30	31260141293175	0.111058
31	227465492997341	0.80812
32	86958230729038	0.308938
33	230523886715532	0.818985
34	183423710975390	0.651652
35	168350181967111	0.5981
36	31296116686503	0.111186
37	150762426938897	0.535616
38	180503578598410	0.641278
39	182783945703321	0.649379
40	12310726354868	0.0437365
41	17478867923017	0.0620974
42	91454748397170	0.324913
43	37907694014516	0.134675
44	222522401980183	0.790558
45	178409158491792	0.633837
46	119834215970900	0.425737
47	49568028944851	0.176101
48	108280417537233	0.384689
49	22263526704647	0.0790959
50	276974081125900	0.98401

2 Conclusiones

Este metodo es muy utilizado por las librerias de muchos lenguajes de programacion, como por ejemplo random en python o en java. Debido a su eficiencia es recomendado para generar numeros pseudoaleatorios.

Los numeros pseudoaleatorios que aparezcan dependeran de los parametros que se le de a la funcion, es por esto que las librerias de generacion de numeros aleatorios utilizan semillas muy grandes para garantizar una distribucion uniforme de los numeros aleatorios generados.

3 Bibliografia

[1] https://es.wikipedia.org/wiki/Generador_lineal_congruencial

[2] <https://www.monografias.com/trabajos108/numeros-primos-si-o-numeros-primos-relativos/numeros-primos-si-o-numeros-primos-relativos.shtml>