

Chapter 26 Random-Number Generation

- One key step in developing a simulation :
random number generation with a specific distribution
- Two steps
 - Random number generation :
Generate a uniformly distributed sequence $[0,1]$ (Ch.26)
* How well uniform ? (Ch.27)
 - Random variate generation :
Transform it to a specific distribution (Ch.28)
* Distributions (Ch.29)

Random Number Function

- Recursive equation : $x_n = f(x_{n-1}, x_{n-2}, \dots)$
 - (Example) $x_n = (5x_{n-1} + 1) \bmod 16$ & divide by 15, seed : x_0
cycle length of 16 and zero tail (initial non-repeated part)
- Good Generator Function
 - Uniformity in $[0,1]$ (mean=0.5, variance=1/12)
 - Reproducibility (deterministic or pseudo-random)
 - Efficiently computable
 - Long cycle length
 - Successive numbers should be independent
(correlation between successive numbers should be small)

Linear Congruential Generators (LCGs)

- $x_n = a^n \text{ mod } m$ or $x_n = ax_{n-1} \text{ mod } m$
 - D.H.Lehmer (1951) shows its good randomness property
 - Generally, $x_n = (ax_{n-1} + b) \text{ mod } m$

● Comments

- Choice of “m”
 - should be large for long cycle length ($\leq m$)
 - $m=2^k$ for easy computation
- maximum possible cycle length can be obtained iff
 - m and b are relatively prime
 - every prime # that is a factor of m is also a factor of a-1
 - $a=4p+1$ if $m=4q$ (p,q are integers)
- (Ex) All met if $m=2^k$, $a=4c+1$ and b is odd
 - $x_n = ((2^{34}+1)x_{n-1} + 1) \text{ mod } 2^{35}$: autocorrelation 0.25 (X)
 - $x_n = ((2^{18}+1)x_{n-1} + 1) \text{ mod } 2^{35}$: autocorrelation 2^{-18} (O)

Multiplicative LCG (b=0)

- $x_n = ax_{n-1} \bmod 2^k$
 - very efficient computation
 - max possible period (cycle length) = 2^{k-2}
if $a=8i \pm 3$ and the initial seed is odd
- $x_n = ax_{n-1} \bmod m$ ($m \neq 2^k$)
 - max possible period = $m-1$
iff a is a primitive root of m
iff $a^n \bmod m \neq 1$ for $n=1,2,\dots,m-2$
 - (Example) $x_n = ax_{n-1} \bmod 31$
 - primitive roots of 31 : 3, 11, 12, 13, 17, 21, 22, 24
 - $x_n = 3x_{n-1} \bmod 31$: period=30 with $x_0=1$
 - $x_n = 5x_{n-1} \bmod 31$: period=3 with $x_0=1$ ($5^3 \bmod 31 = 1$)

Multiplicative LCG (b=0) (cont'd)

- Issues : Round-off errors & Integer overflow
- Schrage's method (1979) - avoid "ax"
 - $ax \bmod m = g(x) + mh(x) \quad (<= m-1)$
 $g(x) = a(x \bmod q) - r(x \operatorname{div} q), \quad h(x) = (x \operatorname{div} q) - (ax \operatorname{div} m)$
 $q = m \operatorname{div} a, \quad r = m \bmod a \quad (\text{condition } r < q)$
 - $h(x) = (x \operatorname{div} (m \operatorname{div} a)) - (ax \operatorname{div} m)$
 $= 0 \text{ or } 1 \quad (g(x)'s \text{ sign can guide it})$
new $x = g(x) >= 0 ? g(x) : g(x) + m$
 - (Example) $x_n = 7^5 x_{n-1} \bmod (2^{31}-1)$
 $a=(7^5)=16807, m=(2^{31}-1)=2147483647, q=12773, r=2863$

Other Random # Generators

- Tausworthe Generators (1965)

- $b_n = c_{r-1} b_{n-1} \oplus c_{r-2} b_{n-2} \oplus c_{r-3} b_{n-3} \oplus c_{r-4} b_{n-4} \oplus \dots \oplus c_0 b_{n-r}$ ($c_i, b_i = 0$ or 1)
- (Example) $b_n = b_{n-4} \oplus b_{n-7}$
initial value $b_0 = b_1 = \dots = b_6 = 1$, then $b_7 = 0, b_8 = 0, \dots$
- Max period = $2^r - 1$ bits
- Simple hardware using LFSR (linear feedback shift registers)

- Extended Fibonacci Generators

- $x_n = (x_{n-1} + x_{n-2}) \bmod m$: high serial correlation
- $x_n = (x_{n-5} + x_{n-17}) \bmod 2^k$: by Marsaglia (1983)

- Combined Generators

- Add two : $w_n = (x_n + y_n) \bmod m$
- Exclusive-or two
- Shuffle with two

Good Random # Generators

- $x_n = 7^5 x_{n-1} \bmod (2^{31}-1)$: IBM SIMPL/I, APL systems (~'70),
PRIMOS OS, IMSL Library (~'80)
 - popular multiplicative LCG
- $x_n = 630360016 x_{n-1} \bmod (2^{31}-1)$: SIMSCRIPT II.5,
DEC-20 FORTRAN
- $x_n = (2^{16}+3) x_{n-1} \bmod 2^{31}$: known as RANDU
 - not used today : unsatisfactory 3-distributivity
- $x_n = (1103515245 x_{n-1} + 12345) \bmod 2^{32}$: used in UNIX
- Still need new ? - fast computers : 2^{31} becomes a short period
parallel computers : parallel algorithm for RNG

Seed Selection

- Do not use zero
- Avoid even values
- Do not use one stream for all variables
- Use nonoverlapping streams
 - e.g. u_0 for the seed of the first stream,
 u_{10000} for the seed of the second stream,
 u_{20000} for the seed of the third stream
- May reuse left-over seed in successive replications
- Do not use random seeds for reproducibility

Chapter 27 Testing RNGs

- Does it produce a sufficiently random stream ?
 - IID = Independently and identically distributed $U(0,1)$
 - Uniformity test & Independence test
- Level of significance α
 - Confidence level $100(1 - \alpha)\%$
 - $H_0 : R_i \sim U[0,1]$ & Independent \Leftarrow null hypotheses
 - $H_1 : R_i \neq U[0,1]$ & Dependent
 - $\alpha = \text{Prob}[\text{reject } H_0 \mid H_0 \text{ true}]$
 - does not tell about $\beta = \text{Prob}[\text{accept } H_1 \mid H_1 \text{ true}]$

Goodness-of-Fit Tests

- One-Dimensional Test
 - Chi-Square Test : check frequency (pdf)
 - Kolmogorov-Smirnov Test : check frequency (cdf)
 - Gap Test : check interarrival of the same data
 - Serial-Correlation Test : check autocovariance
- k-Dimensional Uniformity (k-Distributivity)
 - Serial Test
 - Spectral Test

Chi-Square Test

[kai]

● Method

- Histogram of observed data is prepared
- If $D = \sum_{i=1}^k (o_i - e_i)^2 / e_i < \chi^2_{[1-\alpha; k-1]}$, it is acceptable with sig. level α (see Table A.5)

k: # cells, e_i : expected frequency, o_i : observed frequency

● Example

- $x_n = (125x_{n-1} + 1) \bmod 2^{12}$, $x_0=1$
 - cell
- | | | | | | | | | |
|-----------------------|------|------|------|------|-----|------|------|-------|
| cell | 1 | 2 | 3 | 4 | ... | 9 | 10 | total |
| observed | 100 | 96 | 98 | 85 | ... | 107 | 94 | 1000 |
| expected | 100 | 100 | 100 | 100 | ... | 100 | 100 | 1000 |
| $(o_i - e_i)^2 / e_i$ | 0.00 | 0.16 | 0.04 | 2.25 | ... | 0.49 | 0.36 | 10.38 |
- $D=10.38 < 14.68 = \chi^2_{[1-0.1; 9]}$: uniform distribution with 90% CL

Chi-Square Test (cont'd)

- Sample size > 50, observations/cell > 5
- Example

- H_0 : arrival data follows Poisson distribution with $\lambda=3.64$
- $p(x) = (e^{-\lambda} \lambda^x)/x!$, $x=0, 1, 2, \dots$
0 , otherwise
- $p(0)=0.026, p(1)=0.096, p(2)=0.174, \dots, p(11)=0.001$

– xi	0	1	2	3	4	5	6	7	8	9	10	11	total
oi	12	10	19	17	10	8	7	5	5	3	3	1	100
ei	<u>2.6</u>	<u>9.6</u>	17.4	21.1	19.2	14.0	8.5	<u>4.4</u>	<u>2.0</u>	<u>0.8</u>	<u>0.3</u>	<u>0.1</u>	100.0
$(o_i - e_i)^2/e_i$	7.87	0.15	0.80	4.41	2.57	0.26				11.62			27.68

- at $\alpha=0.05$ (95% CL), $\chi^2_{[1-0.05;5]}=11.1 > 27.68$: H_0 is rejected

Kolmogorov-Smirnov Test

- Kolmogorov-Smirnov Test

- $K = \sqrt{n} \max |F_o(x) - F_e(x)|$

- n: # observations

- If $K < K_{[1-\alpha;n]}$, it is acceptable (see Table A.9)

- Comparison with Chi-Square Test

- Both can be applied to any distribution

- C-S Target: large samples & discrete distribution

- K-S Target: small samples & continuous distribution

- K-S : No cell size dependency as in Chi-Square Test

Gap Test

- Check the gaps between successive occurrence of the same data
- Example
 - random sequence of one digit numbers
 - 4, 1, 3, 5, 1, 7, 2, 8, 2, 0, 7, 9, 1, 3, 5, 2, 7, 9, 4, 1, 6, 3, (110 data)
 - “3” occurs 17 times in the list
 - first gap=10, second gap=7,...
 - $p[\text{gap length } 10 \text{ with value "3"}] = (p[\text{not } 3])^9 p[3] = (0.9)^9 \times 0.1$
 - $F(x) = p[\text{gap} \leq x] = 0.1 \times \sum_{i=0}^x (0.9)^i = 1 - 0.9^{x+1}$ if sufficiently random
 - there are 100 gap data out of 110 data : apply K-S Test

Gap Test (cont'd)

– gap length	0-3	4-7	8-11	12-15	44-47
frequency	35	22	17	9	1
relative freq	0.35	0.22	0.17	0.09	0.01
cumulative	0.35	0.57	0.74	0.83	1.00
$F_e(x)$	0.34	0.57	0.72	0.82	1.00
$ F_e(x)-cum. $	0.01	0.00	0.02	0.01	0.00

- $K = \sqrt{n} \max|F_o(x)-F_e(x)|$
 $= \sqrt{100} \times 0.0224$
 $= 0.224 < 1.36 = K_{[1-0.05;100]}$
 \Rightarrow The sequence is independent

Serial-Correlation Test

- Check the covariance

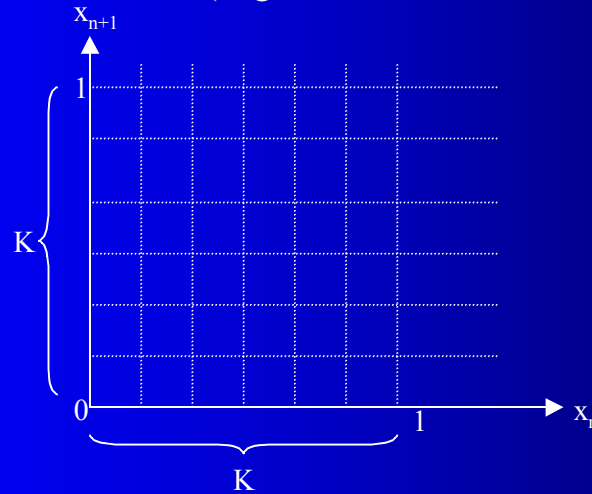
- COV between numbers that are k apart in a stream must be zero
- $R_k = (\sum_{i=1}^{n-k} (o_i - 0.5)(o_{i+k} - 0.5)) / (n-k)$
 - R_k is normally distributed with mean=0, variance= $1/(144(n-k))$
 - For uniform distribution $[0,1]$,
mean = $\int_0^1 t \cdot 1 dt = 0.5$,
variance = $\int_0^1 (t-1/2)^2 \cdot 1 dt = 1/12$
- 100(1- α)% confidence interval is $R_k \pm z/(12\sqrt{(n-k)})$
- If this CI does not include zero, the sequence has a significant correlation

k-Dimensional Uniformity (k-Distributivity)

- RNG Test : Uniformity & “Independence”
- If o_i is the i th random number with uniform dist. in $[0,1]$,
“independence” means $P\{o_{i+1}|o_i\} = P\{o_{i+1}\}$
 - $P\{o_i \text{ is in } [a_1, b_1]\} = b_1 - a_1$: 1-distributivity
 - $P\{o_{i+1} \text{ is in } [a_2, b_2) \text{ and } o_i \text{ is in } [a_1, b_1]\} = (b_1 - a_1)(b_2 - a_2)$: 2-distributivity
 - ...
 - $P\{o_{i+k-1} \text{ is in } [a_k, b_k) \dots \text{ and } o_i \text{ is in } [a_1, b_1]\} = (b_1 - a_1) \dots (b_k - a_k)$: k-dist.

Serial Test

- (Example) 2-distributivity
 - Given n random numbers, (x_1, x_2, \dots, x_n) , all between $[0, 1]$
 - Make a nonoverlapping $n/2$ pairs $(x_1, x_2), (x_3, x_4) \dots (x_{n-1}, x_n)$
(why nonoverlapping ?)
 - Count the # of numbers in each cell (K^2 cells)
 - Apply the C-S Test (degree of freedom = $K^2 - 1$)



Spectral Test

- k-tuples (x_1, x_2, \dots, x_k)
 - How uniformly the k-tuples can fill up the k-dimensional hyperspace
 - the k-tuples from an LCG fall on a finite number of parallel hyperplanes
 - Max distance between adjacent hyperplanes must be small
 - “RANDU”

