CPSC 531: Random Numbers

Jonathan Hudson

Department of Computer Science

University of Calgary

http://www.ucalgary.ca/~hudsonj/531F17

Introduction

- In simulations, we generate random values for variables with a specified distribution
  - E.g., model service times using the exponential distribution
- Generation of random values is a two step process
  1. Random number generation: Generate random numbers uniformly distributed between 0 and 1
  2. Random variate generation: Transform the above generated random numbers to obtain numbers satisfying the desired distribution

Pseudo Random Numbers

- Common pseudo random number generators determine the next random number as a function of the previously generated random number (i.e., recursive calculations are applied)

$$x_n = f(x_{n-1}, x_{n-2}, x_{x-3}, \dots)$$

- Random numbers generated, are therefore, **deterministic**. That is, sequence of random numbers is known **a priori (BEFORE)** given the starting number (called the seed). For this reason, random numbers are known as **pseudo random**.
  - True random number generator's would produce numbers that are independent of those previous
  - We can determine quality of **uniformity** and **independence** of pseudo RNG with statistical tests

A Sample Generator

$$x_n = (5x_{n-1} + 1)\ mod\ 16$$

A Sample Generator

$$x_n = (5x_{n-1} + 1)\ mod\ 16$$

- Starting with $x_0 = 5$:
- The first 32 numbers obtained by the above procedure
  10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, *10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5.*

A Sample Generator

$$x_n = (5x_{n-1} + 1)\ mod\ 16$$

- Starting with $x_0 = 5$:

- The first 32 numbers obtained by the above procedure
10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, *10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5.*

- By dividing x's by 16:
0.6250, 0.1875, 0.0000, 0.0625, 0.3750, 0.9375, 0.7500, 0.8125, 0.1250, 0.6875, 0.5000, 0.5625, 0.8750, 0.4375, 0.2500, 0.3125, *0.6250, 0.1875, 0.0000, 0.0625, 0.3750, 0.9375, 0.7500, 0.8125, 0.1250, 0.6875, 0.5000, 0.5625, 0.8750, 0.4375, 0.2500, 0.3125.*

A Sample Generator

$$x_n = (5x_{n-1} + 1)\ mod\ 16$$

- Starting with $x_0 = 5$:
- The first 32 numbers obtained by the above procedure
  10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, *10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5.*
- The length of the sequence before full repetition is known as the cycle length (**period**) This example has a period of 16
  - Some generators do not repeat an initial portion of the sequence referred to as the “tail” of the sequence

Desirable Properties

Random number generation routines should be:

- Computationally efficient
- Portable
- Have sufficiently long cycle
- Replicable (given the same seed)
  - Helps program debugging
  - Helpful when comparing alternative system design
- Should have provision to generate several streams of random numbers
- Closely approximate the ideal statistical properties of **uniformity** and **independence** .

Linear Congruential Generator (LCG)

- Commonly used algorithm
- A sequence of integers $x_1, x_2, \dots$ between 0 and m-1 is generated according to

$$x = (a * x_{i-1} + c) \; mod \; m$$

  - where **multiplier** a and **increment** c are constants, m is the **modulus** and x_0 is the **seed** (or starting value)

- Random numbers $u_1, u_2, \dots$ are given by $u_i = \frac{X_i}{m} \quad i = 1,2, \dots$
- The sequence can be reproduced if the seed is known

More on LCG

- Selection of the values of a, c, m, and X_0 affects the statistical properties of the generator and its cycle length.

- If $c = 0$, the generator is called **Multiplicative** LCG. (Ex Lehmer page 39)

$$x_n = (5 * x_{n-1}) \ mod \ 2^5$$

- If $c \neq 0$, the generator is called **Mixed** LCG

$$x_n = ((2^{34} + 1) * x_{n-1} + 1) \ mod \ 2^{35}$$

Even more on LCG

- Can have at most m distinct integers in the sequence
  - As soon as any number in the sequence is repeated, the whole sequence is repeated
  - **Period**: number of distinct integers generated before repetition occurs
- Problem: Instead of continuous, the u_i's can only take on discrete values $0, 1/m, 2/m, \ldots, (m-1)/m$
  - Solution: m should be selected to be very large in order to achieve the effect of a continuous distribution (typically, $m > 10^9$)
- Most digital computers use a binary representation of numbers
  - Speed and efficiency are aided by a modulus, m, to be (or close to) a power of 2

Seed Selection

$$x_n = (5 * x_{n-1})\ mod\ 2^5$$

- Using a seed of $x_0 = 1$:

  5, 25, 29, 17, 21, 9, 13, 1, 5,…

  Period = 8

- With $x_0 = 2$:

  10, 18, 26, 2, 10,…

  Period is only 4

- Possible period 32

Note: Full period is a nice property but uniformity and independence are more important

Seed Selection

- Seed selection
  - Any value in the sequence can be used to “seed” the generator
- Do not use random seeds: such as the time of day
  - Cannot reproduce. Cannot guarantee non-overlap.
- Do not use zero:
  - Fine for mixed LCGs
  - But multiplicative LCGs will stuck at zero
- Avoid even values:
  - For multiplicative LCG with modulus $m=2^k$, the seed should be odd
- Do not use successive seeds
  - May result in strong correlation

Example RNGs

- A currently popular multiplicative LCG is:

$$x_n = (7^5 * x_{n-1})\, mod(2^{31} - 1)$$

  - 2^{31}-1 is a prime number and 7^5 is a primitive root of it
    → Full period of 2^{31}-2.
- This generator has been extensively analyzed and shown to be rather good
  - Modulus is largest 32 bit integer prime
  - $a = (7^5)\, mod(2^{31} - 1) = 16807$
- $a = 48271$ has been shown to generate slightly more random sequences

Myths About Random-Number Generation

- A complex set of operations leads to random results.
  It is better to use simple operations that can be analytically evaluated for randomness.

- Random numbers are unpredictable.
  Easy to compute the parameters, a, c, and m from a few numbers => LCGs are unsuitable for cryptographic applications

Myths (Cont)

- Some seeds are better than others. May be true for some.

$$x_n = (9806 \ * x_{n-1} + 1) \ mod \ (2^{17} - 1)$$

  - Works correctly for all seeds except x_0 = 37911
  - Stuck at x_n= 37911 forever
  - Such generators should be avoided
  - Any nonzero seed in the valid range should produce an equally good sequence
  - Generators whose period or randomness depends upon the seed should not be used, since an unsuspecting user may not remember to follow all the guidelines

$$2^{17} - 1 = 131071$$

Myths (Cont)

- Accurate implementation is not important.
  - RNGs must be implemented without any overflow or truncation
    For example:

$$x_n = 1103515245x_{n-1} + 12345 \ mod \ 2^{31}$$

Straightforward multiplication above may produce overflow.

$$2^{31} = 2147483648$$

Testing Random Number Generators

- Two categories of test
  - Test for **uniformity**
  - Test for **independence**
- Passing a test is only **a necessary** condition and **not a sufficient** condition
  - i.e., if a generator fails a test it implies it is bad but if a generator passes a test it does not necessarily imply it is good.

More on Testing …

- Testing is not necessary if a well-known simulation package is used or if a well-tested generator is used

- In what follows, we focus on “empirical” tests, that is tests that are applied to an actual sequence of random numbers
  - Chi-Square Test
  - KS Test

Chi-Square Test

- Prepare a histogram of the empirical data with k cells
- Let O_i and E_i be the observed and expected frequency of the ith cell, respectively. Compute the following:

$$X_0^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

- X_0^2 has a Chi-Square distribution with (k-1) degrees of freedom

Chi-Square Test (continued ...)

- Define a null hypothesis, $H(0)$, that observations come from a specified distribution

- The null hypothesis cannot be rejected at a significance level of α if

$$X_0^2 < X_{[1-\alpha,k-s-1]}^2$$

meaning of significance level $\alpha = P(reject\ H(0) \mid H(0)\ is\ true)$

- s is number parameters in the distribution $s = 1$ poisson $s = 2$ normal
- There is a Chi-Square table that comparison can be made to

Chi-Square Test Example

Interval	Oi	Ei	Chi-Sq
1	50	50	0
2	48	50	0.08
3	49	50	0.02
4	42	50	1.28
5	52	50	0.08
6	45	50	0.5
7	63	50	3.38
8	54	50	0.32
9	50	50	0
10	47	50	0.18
	500		5.84

- Example: 500 random numbers generated using a random number generator; observations categorized into cells at $k = 10$ intervals of 0.1, between 0 and 1. At level of significance of 0.1, are these numbers IID $U(0,1)$?
- $X_0^2 = 5.84$
- Chi-Sq table $X_{[0.9,9]}^2 = 14.68$
- Hypothesis accepted at significance level of 0.10.

More on Chi-Square Test

- Errors in cells with small E_i's affect the test statistics more than cells with large E_i's.
- Minimum size of E_i debated
  - recommends a value of 3 or more; if not combine adjacent cells.
- Test designed for discrete distributions and large sample sizes only. For continuous distributions, Chi-Square test is only an approximation
  - (i.e., level of significance holds only for $n \to \infty$).

Kolmogorov-Smirnov (KS) Test

- Difference between observed CDF $F_0(x)$ and expected CDF $F_e(x)$ should be small; formalizes the idea behind the Q-Q plot.
- Step 1: Rank observations from smallest to largest:

$$Y_1 \leq Y_2 \leq Y_3 \leq \ldots \leq Y_n$$

- Step 2: Define $F_o(x) = (\#i\colon Y_i \leq x)/n$
  - Number of samples <= x / n
- Step 3: Compute K as follows:
- $K = \max_x |F_e(x) - F_0(x)|$
- $K = \max_{1 \leq j \leq n} \{\frac{j}{n} - F_e(Y_j), F_e(Y_j) - \frac{j-1}{n}\}$

Kolmogorov-Smirnov (KS) Test

Yj	j	$\frac{j}{n} - F_e(Y_j)$	$F_e(Y_j) - \frac{j-1}{n}$
5	1	0.017896	0.048771
6	2	0.075098	-0.00843
6	3	0.141765	-0.0751
17	4	0.110331	-0.04366
25	5	0.112134	-0.04547
39	6	0.077057	-0.01039
60	7	0.015478	0.051188
61	8	0.076684	-0.01002
72	9	0.086752	-0.02009
74	10	0.143781	-0.07711
104	11	0.086788	-0.02012
150	12	0.02313	0.043537
170	13	0.04935	0.017316
195	14	0.075607	-0.00894
229	15	0.101266	-0.0346
	MAX	0.143781	0.051188

- Example: Test if given population is exponential with parameter $\beta = 0.01$; that is $F_e(x) = 1 - e^{-\beta x}$;
- $K_{[0.9,15]} = 1.0298.$
- Max is less so observations pass test.

Vs.

K-S Test	Chi-Square Test
• Small Samples • Continuous Distributions • Differences between observed and expected cumulative probabilities • Uses each observation in the sample without any grouping • Cell size is not a problem • Exact	• Large Samples • Discrete Distributions • Differences between observed and hypothesized probabilities • Groups observations into small number of cells • Cells sizes affect the conclusion but no firm guidelines • Approximate