Random Number Generation

Random numbers are essential in the simulation of almost all discrete systems.

Properties of Random Number

- Two important statistical properties for a sequence of random numbers:
  - Uniformity
  - Independence
  - Each random number is an independent sample drawn from a continuous uniform distribution between 0 and 1.
- Uniform distribution on (a, b); its pdf
  - f(x) = 1 / (b – a), a <= x <= b or
    = 0, otherwise
- Now a = 0, b = 1.
  - What is f(x)?
  - Expected value of each R_i?
  - Variance?

Generation of Pseudo-Random Numbers

- Pseudo is used to imply that the act of generating random numbers by a known method removes the potential for true randomness.
- Why?
  - The set of random numbers can be replicated for debugging or comparisons.
- Important criteria for selection of a random number generator:
  - Fast: millions numbers may be generated.
  - Portable to different machines & languages.
  - Long cycle before numbers begin to repeat themselves.
  - Replicable given a seed or starting point.
  - Closely approximate uniformity and independence.

Techniques for Generating Random Numbers

- Linear Congruential Method
  - Most widely used.
  - Recursive formula: $X_{i+1} = (aX_i + c) \bmod m$, i = 0, 1, 2, …
    - produces a sequence of integers, X_1, X_2, … between 0 and m – 1.
    - X_0: initial value or seed
    - a: constant multiplier
    - c: increment
    - m: modulus
    - If c != 0, it's called mixed congruential method.
    - If c = 0, it's called multiplicative congruential method.
- The selection of the values for a, c, m, and X_0 drastically affects the statistical properties and the cycle length.
- Was a hot research topic.
- A key classical reference is Knuth, D. W, *The Art of Computer Programming, vol. 2: Semi-numerical Algorithms*, Addison-Wesley, Reading, MA, 1981.

More on Techniques

- Ultimate test of any generation scheme:
  - uniformity
  - independence
- Other properties:
  - maximum density: the values assumed by R_i, i = 1, 2, …, leave no large gaps on [0, 1].
  - maximum period: avoid cycling or recurrence of the same sequence of generated numbers.
    - Can be achieved by careful choice of a, c, m, and X_0.
      - pages 259-260.
      - Example 7.2 on period length and gaps in or density of generated numbers.
      - Example 7.4.

Combined Linear Congruential Generators

- Complexity, reliability, and problem size have increased, there is a need to derive generators with substantially longer periods.
- One approach is to combine two or more multiplicative congruential generators.
- Example 7.5 has a period ~ 2×10^{18}.
- Even such a large number may not be adequate for some applications. Longer period can be as long as ~ 3×10^{57}.

Tests for Random Numbers

- Objective is to test uniformity and independence.
- Uniformity
  - Frequency test. Use Kolmogorov-Smirnov or chi-square method to test if the distribution is U[0, 1]
- Independence
  - Runs test. Tests the runs up/down or above/below the mean, using the chi-square method.
  - Autocorrelation test. Tests correlation between numbers in the series and compare to the expected correlation of zero.
  - Gap test. Counts the # of digits that appear between the repetitions of a particular digit and apply K-S test.
  - Poker test. Tests the frequency of certain digits in a series of numbers.

Tests for Random Numbers

- Significance level α must be stated. α is the probability of rejecting the null hypothesis given that the null hypothesis is true.
- Two types of errors can occur
  - Type I error: reject H_0 when in fact it is true.
  - Type II error: accept H_0 when it is false (we fail to reject H_0)
- If several tests are conducted on the same set of numbers, the probability of rejecting the null hypothesis on at least one test, by chance alone, increases.
- Similarly, if one test is conducted on many sets of numbers from a generator, the probability of rejecting the null hypothesis on at least one test by chance alone, (Type I error) increases as more sets of numbers are tested.
- How, then, should we decide to accept or reject? α

Frequency Tests

- Used to test uniformity.
- Two test methods: both measure the degree of agreement between the distribution of a sample of generated random numbers & the theoretical uniform distribution.
  - Kolmogorov-Smirnov test, more powerful, can be used for small sample sizes.
  - chi-square test.

Kolmogorov-Smirnov Test for Frequency

- Based on the largest absolute deviation between F(x) and $S_N(x)$ over the range of the random variable.
  - $D = \max |F(x) - S_N(x)|$
  - F(X) is continuous cdf of the uniform dist.
  - S(x) is the empirical cdf
- Steps:
  - Rank data in increasing order.
  - Compute D^+ and D^-.
  - D = man ($D^{+,}$ $D^{-)}$
  - Determine the critical value, $D\alpha$ for the specified significance level α and the given sample size N.
  - If $D = D\alpha$, Ho is rejected, else no difference has been detected between the sample distribution and the uniform distribution.

Chi-Square Test for Frequency

- Divide the total number of observations (N) into equally numbered classes (n).
- Determine the significance level α.
- $X_0^2 = ? \ (O_i - E_i)2 \ / \ E_i$
  - X_0^2: approximate chi-square distribution with n – 1 degrees of freedom.
  - O_i: observed number in the *i*th class.
  - E_i: expected number in the *i*th class = N/n.
- If X_0^2 = tabulated value, the null hypothesis is not rejected.
- Valid only for large samples, N = 50.

What Frequency Test Can't?

- Both K-S and chi-square methods discussed above are used for frequency test.
- Passing frequency test (uniformity) doesn't mean the numbers generated are random. Need to validate the independence of random numbers.
  - Exmaple:

    0.09 0.08 0.01 0.05 0.07

    0.13 0.19 0.12 0.18 0.16

    0.20 0.26 0.22 0.28 0.25

    …

  Any problem?

Runs Tests

- Three types of runs test
  - Runs up and runs down
  - Runs above and below the mean
  - Length of runs
- A run is defined as a succession of similar events preceded and followed by a different event.
- The length of the run is the number of events that occur in the run.
- Example of tossing a coin 10 times:

  H T T H H T T T H T
  - What's the number of runs?
  - What's the length of each run?
  - These are the two concerns in a runs test.

Runs Up and Runs Down

- An up (down) run is a sequence of numbers each of which is succeeded by a larger (smaller) number.
  - Example on page 270.
  - How many runs in the following sequences?
    - 0.08 0.18 0.23 0.36 0.42 0.55 0.63 0.72 0.89 0.91
    - 0.08 0.93 0.15 0.96 0.26 0.84 0.28 0.79 0.36 0.57
  - The mean and variance of a, the total number of runs in a truly random sequence
    - $\mu_a = (2N - 1) / 3$
    - var(a) = (16N – 29) / 90
    - For N > 20, the distribution of a is approximated by a normal distribution, $N(\mu_a, var(a))$.
    - Z transformation.
  - Example 7.8

Runs Above and Below the Mean

- The test for runs up and down is not completely adequate to evaluate the independence of a sequence of numbers.
- Example on p. 273 is rearranged from example 7.8 such that the first 20 numbers are all above the mean and the last 20 numbers are all below the mean.
- Runs are redefined as being above the mean or below the mean.
- Let n_2 and n_2 be the number of above and below the mean; $N = n_1 + n_2$; and b be the total number of runs.
  - The mean and the variance of b for a truly independent sequence are
    - $\mu_b = 2\, n_1\, n_2 / N + 1/2$
    - $\text{var}(b) = 2\, n_1\, n_2(2\, n_2\, n_2 - N) / N^2(N - 1)$
    - For either n_1 or $n_2 > 20$, b is approximately normally distributed.

Length of Runs

- It is expected that length of runs should not be a constant.
- Let Yi be the number of runs of length i in a sequence of N numbers.
- The expected value of Yi for runs up and down: (7.8, 7.9)
- The expected value of Yi for runs above and below the mean: (7.10)
  - Approximate probability that a run has a length i: (7.11)
  - Approximate expected length of a run: (7.12)
  - Approximate expected total number of runs of all lengths (7.13)
- The appropriate test is the chi-square test.
- Examples 7.10 and 7.11.

Tests for Autocorrelation

- Deals with the dependence between numbers in a sequence: the numbers in the sequence might be related.
- The test requires the computation of the autocorrelation between every m numbers (lag) starting with the *i*th number.
- Autocorrelation $?_{im}$ between R_i, R_{i+m}, R_{i+2m}, …, $R_{i+(M+1)m}$, M: largest integer s.t. i+(M+1)m <=N
- A nonzero autocorrelation implies a lack of independence, therefore
  - Ho: $?_{im} = 0$
  - H1: $?_{im} <> 0$
- For large values of M, the estimator of $?_{im}$ is approximately normal if the values R_i, R_{i+m}, R_{i+2m}, …, are uncorrelated.
- Example 7.12

Gap Test

- Used to determine the significance of the interval between the recurrences of the same digit.
- P(t followed by exactly x non-t digits) = $(0.9)^x(0.1)$, x = 0, 1, 2, …
- To fully analyze a set of numbers for independence using the gap test, every digit 0, 1, 2, .., 9 must be analyzed.
- The observed frequencies of the various gap sizes for all the digits are recorded and compared to the theoretical frequency using the Kolmogorov-Smirnov test.
- Procedure
  1. Specify the cdf theoretical frequency distribution
  2. Arrange the sample of gaps in a cumulative distribution
  3. Find D between F(x) and $S_N(x)$
  4. Determine the critical value, Dα
  5. If D > Dα , Ho is rejected
- Example 7.13

Poker Test

- Used to test independence and based on the frequency with which certain digits are repeated in a series of numbers.
  - 0.255, 0.577, 0.331, 0.414, 0.828, …
  - A pair of like digits appears in each case.
- In three-digit numbers, only three possibilities:
  - The individual numbers can all be different.
    - P = 0.9 * 0.8 = 0.72
  - The individual numbers can all be the same.
    - P = 0.1 * 0.1 = 0.01
  - There can be one pair of like digits.
    - P = 1 – 0.72 – 0.01 = 0.27
- Apply chi-square test to test Ho

Summary

- Generation of random numbers: congruential method
- Testing of generated random numbers for uniformity and independence
- Most computers and simulation languages have routines that generate random numbers.
- User needs to confirm that the generator has been tested thoroughly. Most generators that are widely used have been extensively validated.
- Even if generated numbers pass all the tests, some underlying pattern may go undetected.
- When testing the random numbers, those properties may eventually be detected, perhaps by chance alone, even when those properties do not exist.