Pseudorandomness

Debdeep Mukhopadhyay

Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
INDIA -721302

Objectives

- **Random Bit Generation**
- **Pseudorandom Bit Generation**
- **Statistical Tests**
- **Crypto-Pseudorandom bit Generation**

Usefulness in Cryptography

- **Enormous**
- **Key stream in One Time Pads**
- **Secret key in block ciphers**
- **primes p, q in the RSA algorithm**
- **private key in Digital Signature Algorithms**
  - **all these quantities must be chosen from a large space**
  - **probability of a particular value being selected should be small to avoid optimized search**

Random Bit Generator

- **It is a device which outputs a sequence of statistically independent and unbiased bits.**
- **A random integer in the range [0,n] can be obtained by generating a random bit sequence of length ceil(log n)+1, and converting into an integer**
- **Ideally true random number generators should be used.**
- **But they are costly and inefficient**
- **The problem can be solved by substituting random bit generators with pseudorandom generators.**

Pseudorandom bit generators

- **It is a deterministic algorithm which given a truly random binary sequence of length k, outputs a binary sequence of length l>>k, which appears to be random.**
  - **input to the PRBG is called seed**
  - **output is called the PRB sequence.**

Random Tests

- **A linear congruential generator produces a PR sequence of numbers x_1, x_2, … according to the linear recurrence:**

  $x_n = ax_{n-1} + b \bmod m,\ n \geq 1$

  This generator passes statistical tests (tests built on the properties of random sequences)

  But given a partial sequence, they are predictable, even if a, b and m are unknown: like the LFSR

Polynomial Statistical Tests

- **A PRBG is said to pass all polynomial time statistical tests if:**
  - **no polynomial time algorithm can correctly distinguish between**
    - **an output sequence of the generator**
    - **a truly random sequence of the same length**

  with probability significant greater than ½.

Next Bit Test

- **A PRBG is said to pass the next bit test if there is no polynomial time algorithm which on input of the first l bits of the sequence s can predict the $(l+1)^{st}$ bit of s with probability significantly greater than ½.**

Universality of the next bit test

- **A PRBG passes the next bit test if and only if it passes all polynomial time statistical tests.**
  - **A PRBG that passes the next bit test, possibly under some possibly unproven but well known mathematical assumptions is called Cryptographically Secure PRBG.**

Random Bit Generators

- **Hardware:**
  - **elapsed time between emission of particles during radioactive decay**
  - **thermal noise from a resistor**
  - **sounds from a microphone**
  - **gate delays in circuits**

Random Bit Generators

- **Software:**
  - system clock
  - elapsed time between keystrokes or mouse movements
  - user input
  - system load in computers
  - network statistics

De-skewing

- **A natural source of random bits is often defective**
  - output bits are biased (probability of a 1 or 0 is not ½)
  - correlated (the probability of a source emitting 1 depends on the previous bit)
- **De-skewing techniques are employed to generate a truly random sequence.**

Example

- **Suppose a generator produces uncorrelated but biased bits**
  - **probability of 1 is p**
  - **probability of 0 is 1-p**
    - **p is unknown but fixed**
  - **Group the output sequence into pairs of bits**
  - **Replace output pairs 01 with 0**
  - **Replace output pairs 10 with 1**
  - **Discard the remaining possible pairs**
- **This makes the sequence unbiased and also uncorrelated.**

A FIPS Pseudorandom bit generation

- **Input: a random, secret 64 bit seed, s, integer m, 3-DES key k**
- **Output: m pseudorandom 64 bit strings, $x_1,\ldots, x_m$**
- **Compute the intermediate value $I=E_k(D)$, where D is the date/time**
- **For i from 1 to m,**
  - **$x_i=E_k(s \wedge I)$**
  - **$s=E_k(x_i \wedge I)$**
- **Return $(x_1,\ldots,x_m)$**

Five Basic Tests

- Let $s = s_0, s_1, \ldots, s_m$ be a binary sequence
- Statistical tests to determine whether the binary sequence possesses specific characteristics that a truly random sequence is likely to have.

Frequency Test

- Also called monobit test
- Determines whether the number of 0's and 1's are approximately same.

Serial Tests

- To determine whether the number of occurrences of 00, 01, 10, 11 as subsequences of s are approximately the same as that in a random sequence.

Poker Test

- Let m be a positive integer.
- Divide the sequence s into k non-overlapping parts each of length m.
- The Poker test determines whether the number of times of occurrence of each possible 2^m subsequence is the same as that in a random sequence.

Runs Test

- **A run of s is a subsequence of s consisting of consecutive 0s or 1s, which is neither preceded nor succeeded by the same symbol.**
- **A run of 0 is called a gap.**
- **A run of 1 is called a block.**
- **A runs test determines whether the number of runs of various lengths in the sequence s is as expected for a random sequence.**

Autocorrelation Test

- **The test checks for correlation between the sequence s and (non-cyclic) shifted versions of it.**

The Normal Distribution

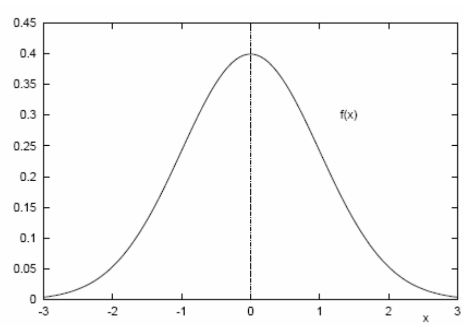
A random variable X has a normal distribution with mean μ and variance σ^2 if its probability density function is defined by:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \ -\infty<x<\infty$$

$Notation: N(\mu,\sigma^2)$

Standard Normal Distribution: N(0,1)

The N(0,1) Distribution



α	0.1	0.05	0.025	0.01	0.005	0.0025	0.001	0.0005
x	1.2816	1.6449	1.9600	2.3263	2.5758	2.8070	3.0902	3.2905

The Chi Square Distribution

Let $v \geq 1$. A random variable X has a χ^2 distribution if the probability density function is defined by:

$$f(x) = \begin{cases} \dfrac{1}{\Gamma(v)2^{v/2}} x^{(v/2)-1} e^{-x/2}, & 0 \leq x < \infty \\ 0, & x < 0 \end{cases}$$

where Γ is the gamma function defined by:

$$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx, \text{ for } t > 0.$$

The mean and variance are v and 2v respectively.

Selected Percentiles

υ	α = 0.100	0.050	0.025	0.010	0.005	0.001
1	2.7055	3.8415	5.0239	6.6349	7.8794	10.8276
2	4.6052	5.9915	7.3778	9.2103	10.5966	13.8155
3	6.2514	7.8147	9.3484	11.3449	12.8382	16.2662
4	7.7794	9.4877	11.1433	13.2767	14.8603	18.4668
5	9.2364	11.0705	12.8325	15.0863	16.7496	20.5150
6	10.6446	12.5916	14.4494	16.8119	18.5476	22.4577
7	12.0170	14.0671	16.0128	18.4753	20.2777	24.3219
8	13.3616	15.5073	17.5345	20.0902	21.9550	26.1245
9	14.6837	16.9190	19.0228	21.6660	23.5894	27.8772
10	15.9872	18.3070	20.4832	23.2093	25.1882	29.5883
11	17.2750	19.6751	21.9200	24.7250	26.7568	31.2641
12	18.5493	21.0261	23.3367	26.2170	28.2995	32.9095
13	19.8119	22.3620	24.7356	27.6882	29.8195	34.5282
14	21.0641	23.6848	26.1189	29.1412	31.3193	36.1233
15	22.3071	24.9958	27.4884	30.5779	32.8013	37.6973
16	23.5418	26.2962	28.8454	31.9999	34.2672	39.2524
17	24.7690	27.5871	30.1910	33.4087	35.7185	40.7902
18	25.9894	28.8693	31.5264	34.8053	37.1565	42.3124
19	27.2036	30.1435	32.8523	36.1909	38.5823	43.8202
20	28.4120	31.4104	34.1696	37.5662	39.9968	45.3147
21	29.6151	32.6706	35.4789	38.9322	41.4011	46.7970
22	30.8133	33.9244	36.7807	40.2894	42.7957	48.2679
23	32.0069	35.1725	38.0756	41.6384	44.1813	49.7282
24	33.1962	36.4150	39.3641	42.9798	45.5585	51.1786
25	34.3816	37.6525	40.6465	44.3141	46.9279	52.6197
26	35.5632	38.8851	41.9232	45.6417	48.2899	54.0520
27	36.7412	40.1133	43.1945	46.9629	49.6449	55.4760
28	37.9159	41.3371	44.4608	48.2782	50.9934	56.8923
29	39.0875	42.5570	45.7223	49.5879	52.3356	58.3012
30	40.2560	43.7730	46.9792	50.8922	53.6720	59.7031
31	41.4217	44.9853	48.2319	52.1914	55.0027	61.0983
63	77.7454	82.5287	86.8296	92.0100	95.6493	103.4424
127	147.8048	154.3015	160.0858	166.9874	171.7961	181.9930
255	284.3359	293.2478	301.1250	310.4574	316.9194	330.5197
511	552.3739	564.6961	575.5298	588.2978	597.0978	615.5149
1023	1081.3794	1098.5208	1113.5334	1131.1587	1143.2653	1168.4972

v=5, α=0.025
x_α=12.8325
=>Pr[x> x_α]= α

Hypothesis Testing

- Hypothesis: It is an assertion about a distribution of one or more random variables.
- Testing of hypothesis is involved with probability.
  - Type I error: good samples are rejected.
  - Type II error: bad samples are accepted.
- The significance level α is thus very important.
  - it is the probability of rejecting a hypothesis when it is good.
  - when it is high we have more Type I error
  - when it is low we have more Type II error

Randomness Testing

- Statistic: A function of the elements of a random sample, for example the number of 0's in a sequence.
- It is assumed that a random distribution is either a normal or chi-square for a value of v.
- A significance level α is chosen, and a value of x_α is fixed.
- The statistic is computed.

Randomness Testing

- **Statistic expected to take on smaller values for random sequences:**
  - **If the statistic $X_S > X_\alpha$ reject.**
  - **one sided test**

- **Statistic expected to take intermediate values for random sequences:**
  - **If the statistic $X_S > X_\alpha$ or $X_S < -X_\alpha$ reject.**
  - **two sided test**

Tests and Statistic

- **All the 5 tests have a corresponding statistic**
  - **example for Frequency Test:**

    $X=(n_0-n_1)^2/n$, where n_0 and n_1 are respectively the number of 0's and 1's in a sequence of size n.

    Expected value of the statistic is low for a random sequence, so we engage an one-sided test.

The RSA bit PRBG

- **Setup: Generate two large primes p, q**
- **Compute N=pq and Φ=(p-1)(q-1)**
- **Select a random integer e, 1<e< Φ, such that gcd(e, Φ)=1**
- **Select a random integer x_0 in the interval [1,n-1]**
- **For i=1 to l do**
  - $x_i=x_{i-1}^e \bmod N$
  - $z_i=LSB(x_i)$
- **The output sequence is z_1, z_2,…**

Blum Blum Shub Generator

- **Generate two large secret random and distinct primes p and q each congruent to 3 mod 4. Compute N=pq.**
- **Select a random integer in [1,N-1] st. gcd(s,N)=1. Compute $x_0=s^2 \bmod N$.**
- **For i from 1 to l, do:**
  - $x_i=x_{i-1}^2 \bmod N$
  - $z_i=LSB(x_i)$
- **The output sequence is $z_1,\ldots,z_l$.**

Points to Ponder!

- **1 round of Feistel Structure is not Pseudorandom.**
- **2 rounds of Feistel Structure is not pseudorandom.**

Further Reading

- **A. Menezes, P. Van Oorschot, Scott Vanstone, "Handbook of Applied Cryptography" (Available online)**

Next Days Topic

- **Cryptographic Hash Functions**