CDM

(Pseudo-) Randomness

Klaus Sutner
Carnegie Mellon University
`www.cs.cmu.edu/~sutner`
Fall 2004

Battleplan

- Randomness in Physics
- Formalizing Randomness
- Practical RNGs

The Magic Device

Quantis RNG

Features

- True quantum randomness *(passes all randomness tests)*
- High bit rate of 4Mbits/sec (up to 16Mbits/sec for PCI card)
- Low-cost device
- Compact and reliable
- PCI card comes with drivers for Windows (2000/XP), Linux (2.4.x, 2.6.x), FreeBSD (4.x, 5.x) and Solaris (sparc & x86, 8, 9).

Applications

- Cryptography
- Numerical Simulations
- Statistical Research
- Lotteries and gambling

Big Step Forward . . .

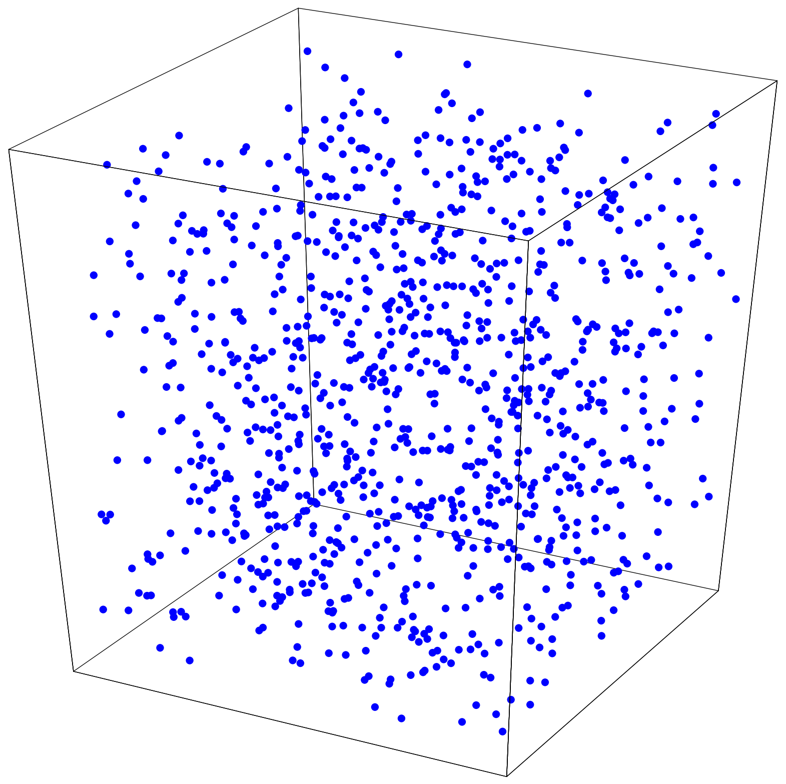
This is a whole lot better than having to keep a bunch of lava lamps around to get random bits.

Never mind a lump of radioactive material and a Geiger counter.

Krypton-85

But if you can keep the radioactive stuff someplace else and obtain the random bits over the web it's not so bad.

Here are 1024 true random bits from `www.fourmilab.ch/hotbits`.



Fiat Lux

Incidentally, Noll and Cooper at Silicon Graphics discovered one day that the pretty lava lamps were completely irrelevant: they could get even better random bits with the lens cap on (there is enough noise in the circuits to get good randomness).

Quantis also uses light, but unlike the original lava lamp system, it exploits an elementary quantum optical process: a photon hitting a semi-transparent mirror either passes or is reflected.

Developed at the University of Geneva, first practical model in 1998.

Note that quantum physics is the only part of physics that claims that the outcome of certain processes is random (which is why Einstein always disliked quantum physics rather strongly).

Pre-History

In the olden days, the RAND Corporation used a kind of electronic roulette wheel to generate a million random digits (rate: one per second).

In 1955 the data were published under the title:

A Million Random Digits With 100,000 Normal Deviates

"Normal deviates" simply means that the distribution of the random numbers is bell-shaped rather than uniform.

But the New York Public Library to shelved the book in the psychology section.

The RAND guys were surprised to find that their original sequence had several defects and required quite a bit of post-processing before it could pass muster as a random sequence. This took years to do.

Available at `http://www.rand.org/publications/classics/randomdigits`.

De-Skewing

Suppose the probability of a 0 is $1/2 + \varepsilon$ and the bits are all independent. To eleminate this bias, John von Neumann suggested to following algorithm:

- Read the bits two at a time.
- Skip 00 and 11.
- For 01 and 10 output the first bit.

The probabilities are easily computed:

$$\begin{array}{lll} 00 & (1/2+\varepsilon)^2 & = 1/4 + \varepsilon + \varepsilon^2 \\ 01 & (1/2+\varepsilon)(1/2-\varepsilon) & = 1/4 - \varepsilon^2 \\ 10 & (1/2-\varepsilon)(1/2+\varepsilon) & = 1/4 - \varepsilon^2 \\ 11 & (1/2-\varepsilon)^2 & = 1/4 - \varepsilon + \varepsilon^2 \end{array}$$

So the probability of a 0 in the de-skewed sequence is the same as for a 1!

Exercise 1. *Explain why von Neumann's de-skewing algorithm is not so great after all.*

Defining Randomness

What does it mean for a sequence to be random? The point here is that we need a purely mathematical definition without and recourse to physics.

First off, it is better to consider infinite sequences $\boldsymbol{x} \in \mathbf{2}^\omega$.

Definition 1. *Density*

Define the density up to n to be $D(\boldsymbol{x}, n) = 1/n \sum_{i<n} x_i$.

The limiting density of $\boldsymbol{x}$ is $\lim_n D(\boldsymbol{x}, n)$ if that limit exists.

It is “clear” that a random sequence should have limiting density 1/2 (though one might wonder why the limit should actually exist in the strict sense of convergence in analysis).

In fact, one would want a certain degree of convergence: the $D(\boldsymbol{x}, n)$ should tend to 1/2, though not too rapidly.

Decimation

In 1919 von Mises suggested a notion of randomness based on the limiting density of the sequence itself and certain derived subsequences.

Definition 2. *An infinite sequence* $\boldsymbol{x} \in \mathbf{2}^\omega$ *is* Mises random *if the limiting density of any subsequence* (x_{i_j}) *is 1/2 where the subsequence is selected by a Auswahlregel.*

So what on earth is a Auswahlregel, a selection rule?

For example, the following all should have limiting density 1/2:

$$x_0, x_1, x_2, \ldots, x_n, \ldots$$

$$x_0, x_2, x_4, \ldots, x_{2n}, \ldots$$

$$x_1, x_4, x_7, \ldots, x_{3n+1}, \ldots$$

$$x_0, x_1, x_4, \ldots, x_{n^2}, \ldots$$

Auswahlregeln

However, there is one big caveat:

The rule must be defined as a function $\mathbb{N} \longrightarrow \mathbb{N}$ without any knowledge of $\boldsymbol{x}$: otherwise we can simply pick a subsequence of all 0's.

Now suppose we have countable system of Auswahlregeln and our sequence passes all these tests. In other words, we have countably many increasing functions $f : \mathbb{N} \longrightarrow \mathbb{N}$ and for each of these:

$$\lim D((x_{f(n)}), n) = 1/2.$$

Then we can think of $\boldsymbol{x}$ as random.

One can show that for any countable collection of Auswahlregeln there are always uncountably many sequences that are random in this sense.

Sounds all eminently reasonable.

Ville

Unfortunately, in 1939 Ville showed that for any countable system of Auswahlregeln there is always a sequence $\boldsymbol{x}$ that passes all the tests (i.e., the limiting density is $1/2$ for all these subsequences) but that is nonetheless biased towards 1.

More precisely, it was known that a random sequence should have

$$\limsup_n \sqrt{\frac{2n}{\log\log n}}(D(\boldsymbol{x}, n) - 1/2) = 1$$

$$\liminf_n \sqrt{\frac{2n}{\log\log n}}(D(\boldsymbol{x}, n) - 1/2) = -1$$

and Ville's example violated the second condition.

Counting Blocks

But there are still several important ideas here.

First, we can look at the density of not just a single bit but of an arbitrary finite blocks $w \in \mathbf{2}^m$: the number of occurences of a particular block w in $x_1 x_2 \ldots x_n$ should approach $n/2^m$.

Example 1. *Suppose we have a de Bruijn sequence X of order k and let $\boldsymbol{x} = X^\omega$. Then $\boldsymbol{x}$ has limiting density $1/2^m$ for all blocks of length at most k but not for larger blocks.*

In practice, counting blocks (up to a certain size) is a very good test for randomness (χ^2 test).

Second, there is the idea that a sequence is random if it passes a number of tests that are designed to reveal non-randomness should it be present. While Mises' original method does not quite work, we will see a better notion of a test in a moment.

Kolmogorov-Randomness

Kolmogorov suggested to use incompressibility as a measure of randomness.

Definition 3. *An infinite sequence $\boldsymbol{x} \in \mathbf{2}^\omega$ is* Kolmogorov-random *if for some constant c: $C(x_1 \dots x_n) \geq n - c$ for all n.*

Again, this sounds eminently reasonable, but unless one is more careful with the definition of C (recall the discussion of prefix complexity and Chaitin's Ω), this won't work.

Theorem 1. *Martin-Löf*

Let f be computable such that $\sum 1/2^{f(i)}$ diverges. Then for any $\boldsymbol{x} \in \mathbf{2}^\omega$ there are infinitely many n such that $C(x_1 \dots x_n) < n - f(n)$

The proof is quite involved. Note that $f(n) = \log n$ satisfies the hypothesis.

Martin-Löf Randomness

So how do we describe a randomness test in general?

Conider a descending chain of sets of binary words

$$K_0 \supseteq K_1 \supseteq K_2 \supseteq \ldots \supseteq K_n \supseteq \ldots$$

where each K_n is closed under suffixes.

Furthermore, let $\mu(K_n) \leq 2^{-n}$ so that these sets are not too fat.

Think of the K_n as subtrees of the full infinite binary tree. Then the infinite branches that lie in all these trees are considered to be failures as far as randomness goes: they can be described by the sequence of test sets.

The choice of the K_n is obviously important here, and we have to make sure they are not too complicated (otherwise we could refute the existence of any random sequence).

Sequential Test

The key lies in imposing a computability constraint.

Definition 4. *A* seqential test *has the additional property that*

$$K = \{\, (n,x) \bigm| x \in K_n \subseteq \mathbf{2}^\star \,\}$$

is semi-decidable.

Here is an amazing result that shows that in essence we only need to deal with a single test.

Theorem 2. *There is a universal sequential test U such that for any sequential test K we have $K_{n+c} \subseteq U_n$ for some constant c.*

The proof uses the existence of a universal Turing machine and is not particularly difficult.

The Definition

Definition 5. *An infinite sequence $\boldsymbol{x}$ is Martin-Löf random if it passes a universal sequential test.*

This definition is justified by the fact that any test of randomness in ordinary probability theory can be translated into a sequential test.

For example, to check for limiting density we can use

$$K(\varepsilon, n) = \bigcup_{k \geq n} \{ \boldsymbol{x} \mid |D(\boldsymbol{x}, k) - 1/2| > \varepsilon \}$$

Exercise 2. *Figure out the details behind this construction. In particular, make sure that semi-decidability holds.*

Exercise 3. *Show how to check for the frequencies of blocks of arbitrary finite length in a sequential test.*

Alas . . .

Martin-Löf randomness provides a good conceptual framework and can be used to prove real theorems about randomness.

But, the definition does not yield any methods to construct a random sequence. Contrariwise:

Theorem 3. *Any Martin-Löf random sequence fails to be computable.*

Exercise 4. *Prove this theorem. Assume a random sequence is computable and show how to construct a test that rejects it.*

Pseudo-Randomness

In the real world, make do with a pseudo-random number generator (PRNG) of the form

$$\begin{aligned} x_0 &\qquad \text{chosen somehow (at random :-))} \\ x_{n+1} &= f(x_n) \end{aligned}$$

where f is easily computable, typically using arithmetic and some bit-plumbing. Of course, we are taking a huge step away from real randomness here.

Anyone attempting to produce random numbers by purely arithmetic means is, of course, in a state of sin.

John von Neumann

Feedback Shift Registers

As we have seen, a simple FSR can be used to produce bit sequences with very long periods.

These sequences have weak random properties, but sometimes they are good enough.

Their most striking feature, though, is their computational cheapness.

One has to be a bit careful with the feedback polynomial: trinomials $x^d + x^k + 1$ are very popular (there are only linearly many for a given span, and there is a lot of literature for them), but have bad properties with respect to RNG.

Linear Congruential Generator

A typical example: a simple affine map modulo m.

$$x_{n+1} = a x_n + b \bmod m$$

The trick here is to choose the proper values for the coefficients. Can be found in papers and on the web.

A choice that works reasonably well is

$$a = 1664525, b = 1013904223, m = 2^{32}$$

Note that a modulus of 2^{32} amount to unsigned integer arithmetic on a 32-bit architecture.

Multiplicative Congruential Generator

Omit the additive offset and use multiplicative constants only.

If need be, use a higher order recurrence.

$$x_n = a_1 x_{n-1} + a_2 x_{n-2} + \ldots a_k x_{n-k} \bmod m$$

For prime moduli one can achieve period length $m^k - 1$.

Almost as fast and easy to implement as LCG.

Inverse Congruential Generator

Choose the modulus m to be a prime number and write

$$\overline{x} = \begin{cases} 0 & \text{if } x = 0, \\ x^{-1} & \text{otherwise.} \end{cases}$$

Then we can define a pseudo-random sequence by

$$x_{n+1} = a\overline{x_n} + b \bmod m$$

Computing the inverse can be handled by the extended Euclidean algorithm.

Again, it is crucial to choose the proper values for the coefficients.

Mersenne Twister

Very recent (1998) method by Matsumoto and Nishimura, seems to be the tool of choice at this point.

- Has huge period of $2^{19937} - 1$, a Mersenne prime.
- Has negligible serial correlation between successive values.
- Only statistically unsound generators are much faster.
- Is statistically random in all the bits of its output (after a bit of post-processing).

The method is very clever and not exactly obvious.

The algorithm works on bit-vectors of length w (typically 32 or 64).

Let k be the degree of the recursion, and choose $1 \leq m < k$ and $0 \leq r < w$.

The MT Recurrence

Recall that $x_i \in \mathbf{2}^w$.

Define the join $\mathrm{join}(x, y)$ of $x, y \in \mathbf{2}^w$ to be the the first $w - r$ bits of x followed by the last r bits of y.

$$x_{n+k} = x_{n+m} + \mathrm{join}(x_n, x_{n+1})A$$

where A is a sparse companion-type matrix that makes it easy to perform the vector-matrix multiplication. E.g.

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & \ddots & & \\ 0 & 0 & 0 & \dots & 1 \\ a_{w-1} & a_{w-2} & a_{w-3} & \dots & a_0 \end{pmatrix}$$

Good Parameters

Here is an excellent choice for the parameters:

$$w = 32, k = 624, m = 397, r = 31$$

and the A matrix is given by $\boldsymbol{a} = \mathrm{0x9908B0DF}$.

Note that this requires a bit of storage for the state: we are dealing with a recurrence of order 624 (need an array of 624 words).

This choice achieves the theoretical upper bound for the period:

$$2^{wk-r} = 2^{19937} \approx 4.32 \times 10^{6001}.$$

After a little bit of post-processing this methods produces very hight quality pseudo-random numbers, and is not overly costly.

Summary

- Attempts to define randomness
  - Van Mises: decimation
  - Kolmogorov: incompressibility
  - Martin-Löf: effective tests
- Practical RNGs
  - Linear congruential generators
  - Multiplicative congruential generators
  - Inverse congruential generators
  - Mersenne twister