1 Introduction

This notes are meant to provide a conceptual background for the numerical construction of random variables. They can be regarded as a mathematical complement to the article [1] (please follow the hyperlink given in the bibliography). You are strongly recommended to read [1]. The recommended source for numerical implementation of random variables is [3] (or its FORTRAN equivalent), [2] provides and discusses analogous routines in PASCAL.

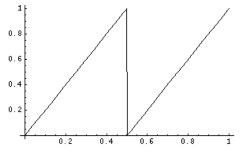
2 Binary shift map

The binary shift is the map

$$\sigma : [0,1] \to [0,1]$$

such that $x_{n+1} = \sigma(x_n)$ is

$$x_{n+1} = 2\,x_n \mod 1 \equiv \begin{cases} 2\,x_n & 0 \leq x_n < \frac{1}{2} \\ 2\,x_n - 1 & \frac{1}{2} \leq x_n \leq 1 \end{cases} \tag{2.1}$$



Adopting the binary representation for any $x \in [0,1]$

$$x = \sum_{i=1}^{\infty} a_i 2^{-i} \qquad \Rightarrow \qquad \boldsymbol{a} = (a_1, a_2, \dots)$$

where

$$a_i \in \{0,1\} \qquad \forall\, i \in \mathbb{N}$$

we see that

- if $0 \leq x < 1/2$ then $a_1 = 0$ and

$$2\sum_{i=2}^{\infty} a_i 2^{-i} = \sum_{i=1}^{\infty} a_{i+1} 2^{-i} \qquad \Rightarrow \qquad \sigma \circ (0, a_2, a_3 \dots) = (a_2, a_3, \dots)$$

- if $1/2 \leq x < 1$ then $a_1 = 1$ and

$$2\sum_{i=1}^{\infty} a_i 2^{-i} - 1 = a_1 - 1 + \sum_{i=1}^{\infty} a_{i+1} 2^{-i} = \sum_{i=1}^{\infty} a_{i+1} 2^{-i} \qquad \Rightarrow \qquad \sigma \circ (a_1, a_2, a_3 \dots) = (a_2, a_3, \dots)$$

Thus, (2.1) acts on any initial condition $x \in [0,1]$ $x \sim (a_1, a_2, \dots)$ by removing the first entry and shifting to the left the ensuing ones. There are relevant consequences:

- The sensitive dependence of the iterates of σ on the initial conditions. If two points x and x' differ only after their n-th digit a_n , i.e. $x = (a_1, \dots, a_n, a_{n+1}, \dots)$ and $x' = (a_1, \dots, a_n, a'_{n+1}, \dots)$ this difference becomes amplified under the action of σ:

$$\sigma^n(x) = (a_{n+1}, \dots) \qquad \& \qquad \sigma^n(x') = (a'_{n+1}, \dots)$$

  where $\sigma^2(x) = \sigma(\sigma(x))$, etc.

- The sequence of iterates $\sigma^n(x)$ has the same random properties as successive tosses of a coin. Namely, $\sigma^n(x)$ is smaller or larger than $1/2$ depending on whether a_{n+1} is zero or one. If we associate to coin tossing a Bernoulli variable

$$\xi : \Omega \to \{0\,,1\} \qquad \xi(H) = 1$$

  we can always associate to any realization of the sequence of i.i.d. $\{\xi_i\}_{i=1}^n$ ($\xi_i \overset{d}{=} \xi$) an binary sequence specifying an $x \in [0,1]$. In other words we have for any $x \in [0,1]$ an isomorphism of the type

$$x \sim \begin{cases} (0 & 1 & 0 & 1 & 1 & 0 & \dots) \\ (C & H & C & H & H & C & \dots) \end{cases}$$

- All dyadic rational numbers i.e. rational numbers of the form

$$\frac{p}{2^a} \qquad p\,,a\,,\in \mathbb{N}$$

  have a terminating binary numeral. This means that the binary representation has a finite number of terms after the radix point e.g.:

$$\frac{3}{2^5} = 0.00011 \tag{2.2}$$

  This means that the set of dyadic rational numbers is the basin of attraction of the fixed point in zero.

- Other rational numbers have binary representation, but instead of terminating, they recur, with a finite sequence of digits repeating indefinitely (i.e. they comprise a *periodic part* which may be preceded by a pre-periodic part):

$$\frac{13}{36} = 0.01\overline{011100}_2$$

  where $\bar{\bullet}$ denotes the periodic part: they correspond to the set of *periodic orbits* together with their basin of attraction of the shift map.

- Binary numerals which neither terminate nor recur represent irrational numbers. Since rational are *dense* on real for any $x \in [0,1]$ and any ε there is at least one point on a periodic orbit (and actually an infinite number of such points) in $[x-\varepsilon, x+\varepsilon]$. This fact has important consequences for numerics. Rational numbers form (and therefore initial conditions for periodic orbits of the binary shift) a *countable infinite set* with zero Lebesgue measure. Generic initial conditions (i.e. real number on $[0,1]$) are *uncountable* and have full Lebesgue measure. Non-periodic orbits are hence in principle *generic*. Not in practice, though, if by that we mean a numerical implementation of the shist map. Computer can work only with finite accuracy numbers: at most they can work with recurring sequences of large period.

Definition 2.1 (Perron-Frobenius operator)**.** *Given a one dimensional map*

$$f : [0,1] \to [0,1]$$

and a probability density ρ over $[0,1]$, the one step-evolution ρ' of ρ with respect to f is governed by the Perron-Frobenius operator defined by

$$\rho'(x) = \mathcal{F}[\rho](x) := \int_0^1 dy\, \delta(x-y)\, \rho(y)$$

The definition of the Perron-Frobenius operator, allows us to associate to any map

$$x_{n+1} = f(x_n)$$

an evolution law for densities

$$\rho_{n+1}(x) = \int_0^1 dy\, \delta(x-y)\, \rho_n(y)$$

In particular we have

Definition 2.2 (*Stationary density*)**.** *A density is stationary with respect to f if*

$$\rho(x) = \int_0^1 dy\, \delta(x - \sigma(y))\rho(y)$$

For the shift map we have

Proposition 2.1 (*Invariant density*)**.** *The uniform distribution $\rho(x) = 1$ is the unque invariant density of the shift map.*

Proof. Using the definition of stationary density and the expression of shift map we have

$$\rho(x) = \int_0^{\frac{1}{2}} dy\, \delta(x - 2\,y)\, \rho(y) + \int_{\frac{1}{2}}^1 dy\, \delta(x - 2\,y + 1)\, \rho(y)$$

For any $x \in [0, 1]$ the integral gives

$$\rho(x) = \frac{1}{2}\left\{\rho\left(\frac{x}{2}\right) + \rho\left(\frac{x+1}{2}\right)\right\}$$

The equality is readily satisfied by setting $\rho(x) = 1$. The solution is also unique. We can establish using the the expression of a generic intial density after n-iterations. In order to determine such an expression we can proceed by induction

- After two steps

$$\rho^{(2)}(x) = \frac{1}{2}\left(\frac{1}{2}\rho\left(\frac{x}{4}\right) + \frac{1}{2}\rho\left(\frac{x+1}{4}\right)\right) + \frac{1}{2}\left(\frac{1}{2}\rho\left(\frac{\frac{x}{2}+1}{2}\right) + \frac{1}{2}\rho\left(\frac{\frac{x+1}{2}+1}{2}\right)\right) = \frac{1}{4}\sum_{i=0}^{3}\rho\left(\frac{x+i}{4}\right) \quad (2.3)$$

- We may infer that after n steps

$$\rho^{(n)}(x) = \frac{1}{2^n}\sum_{i=0}^{2^n-1}\rho\left(\frac{x+i}{2^n}\right) \quad (2.4)$$

- the inference implies that

$$\rho^{(n+1)}(x) = \frac{1}{2^{n+1}}\sum_{i=0}^{2^n-1}\rho\left(\frac{x+i}{2^{n+1}}\right) + \frac{1}{2^{n+1}}\sum_{i=0}^{2^n-1}\rho\left(\frac{\frac{x+i}{2^n}+1}{2}\right) \quad (2.5)$$

  The first sum ranges from $x/2^{n+1}$ to $(x+2^n-1)/2^{n+1}$. The second from $(x+2^n)/2^{n+1}$ to $(x+2^{n+1}-1)/2^{n+1}$. Therefore we can re-write the (2.5) as

$$\rho^{(n+1)}(x) = \frac{1}{2^{n+1}}\sum_{i=0}^{2^{n+1}-1}\rho\left(\frac{x+i}{2^{n+1}}\right) \quad (2.6)$$

  which proves the inference.

In the limit $n \uparrow \infty$ the latter converges to

$$\lim_{n\uparrow\infty} \rho_n(x) = \lim_{n\uparrow\infty} \frac{1}{2^n} \sum_{i=0}^{2^n-1} \rho\left(\frac{x+i}{2^n}\right) = \int_0^1 dy\, \rho(y) = 1$$

□

3 Excursus on Kolmogorov complexity

Definition 3.1 (*Kolmogorov complexity*)**.** *The Kolmogorov complexity $K_{\boldsymbol{U}}(x)$ of a string x with respect to a universal computer $\boldsymbol{U}$ is defined as*

$$K_{\boldsymbol{U}}(x) := \min_{p:\ \boldsymbol{U}(p)=x} \ell(p)$$

the minimum lenght ℓ over all programs p that print x and halt.

Note that $p\colon \boldsymbol{U}(p) = x$ should be read program p which $\boldsymbol{U}$ executes to first print x and then to stop.
Thus, $K_{\boldsymbol{U}}(x)$ is the shortest description length of x over all descriptions interpreted by the universal computer $\boldsymbol{U}$.

Definition 3.2 (*Conditional Kolmogorov complexity*)**.** *The conditional Kolmogorov complexity $K_{\boldsymbol{U}}(x|l(x))$ of a string x with respect to a universal computer $\boldsymbol{U}$*

$$K_{\boldsymbol{U}}(x|l(x)) := \min_{p:\boldsymbol{U}(p,l(x))=x} \ell(p)$$

is the shortest description length if the computer $\boldsymbol{U}$ has the length of x made available to it.

The Kolmogorov complexity of an integer is the Kolmogorov complexity of its binary string representation:

Definition 3.3 (*Kolmogorov complexity of an integer*)**.** *we call*

$$K_{\boldsymbol{U}}(n) := \min_{p:\mathcal{U}(p)=n} \ell(p)$$

If the computer $\boldsymbol{U}$ knows the number of bits in the binary representation of the integer then we only need to provide the values of these bits. The program will have the length

$$\ell(p) = c + \ln n$$

Algorithm 1 Compressible sequence

```
for $k = 1, n$ do
    print 1
end for
stop
```

The Kolmogorov complexity of is

$$K_{\boldsymbol{U}}(\underbrace{1111111\ldots1}_{\text{n times}}) = c + \ln n$$

whilst its conditional complexity implying that the computer knows 1 is

$$K_{\boldsymbol{U}}(\underbrace{1111111\ldots1}_{\text{n times}}|1) = c$$

On the other hand, the complexity of a sequence of n-digits $\{G_k\}_{k=1}^n$ must be $O(n)$: In other words the shortest code is something close to the simplest: *just copy the sequence*.

Algorithm 2 Non-compressible sequence

```
print $G_1$
print $G_2$
print $G_3$
⋮
print $G_n$
stop
```

Definition 3.4 (*Incompressible string*)**.** *We call an infinite string incompressible if*

$$\lim_{n\uparrow\infty} \frac{K_{\boldsymbol{U}}(x_1x_2\dots x_n|n)}{n} = 1$$

4 Basic algorithm for a uniformely distributed random variable

This section floows closely section 7.1 of [3].

The study of the shift map, suggests us that general maps of the form

$$x_{j+1} = a\,x_j + c(\mathrm{mod}\ m) \tag{4.1}$$

make up good starting points for random number generators. The main problem is that since existing computers can only deal with finite arithmetics, these maps will generate recurrent sequences. The recurrence will eventually repeat itself, with a period that is obviously no greater than m. If m, m, and c are properly chosen, then the period will be of maximal length, i.e., of length m. In that case, all possible integers between 0 and $m-1$ occur at some point, so any initial seed choice of x_o is as good as any other: the sequence just takes off from that point. The simplest algorithm proposed by [3] generates a random variable by iterating a map of the form (4.1) with $c = 0$:

$$x_{j+1} = a\,x_j(\mathrm{mod}\ m)$$

with

$$a = 7^5 = 16807 \qquad \& \qquad m = 2^{31} - 1 = 2147483647$$

One sets

$$m = aq + r \qquad \begin{cases} q = 127773 \\ r = 2836 \end{cases}$$

so that if r is small, specifically $r < q$, and $0 < z < m1$, it can be shown that both $a\,(z\,\mathrm{mod}\,q)$ and $r\,\lfloor z\,/\,q\rfloor$ lie in the range $0, ..., m1$, and that

$$a\,z\,\mathrm{mod}\,m = \begin{cases} a(z\,\mathrm{mod}\,q) - r\,\lfloor z/\,q\rfloor & \text{if } \geq 0, \\ a(z\,\mathrm{mod}\,q) - r\,\lfloor z\,/\,q\rfloor + m & \text{otherwise} \end{cases}$$

with $\lfloor\ \rfloor$ denoting the floor integer part.

The period of (4.2) is $2^{31} - 2 \approx 2.110^9$.

Algorithm 3 Ran0: Minimal random number generator of Park and Miller. Returns a uniform random deviate between 0.0 and 1.0.

```
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define MASK 123459876

float ran0(long *idum)

long k;
float ans;

*idum ˆ= MASK;
k=(*idum)/IQ;
*idum=IA*(*idum-k*IQ)-IR*k;
if (*idum ¡ 0) *idum += IM;
ans=AM*(*idum);
*idum ˆ= MASK;
return ans;
```

Algorithm 4 XOR

```
temp = *idum;
*idum= MASK;
MASK=temp
```

5 Gaussian random variable

Suppose $\boldsymbol{\xi} = (\xi_1\,, \xi_2)$ are independent random variable uniformely distributed on the unit interval $[0, 1]$:

$$p_{\boldsymbol{\xi}}(x_1\,, x_2) = 1$$

We can define two new random variables

$$\begin{aligned}\eta_1 &= \sqrt{-2\,\ln \xi_1}\cos(2\,\pi\,\xi_2)\\ \eta_2 &= \sqrt{-2\,\ln \xi_1}\sin(2\,\pi\,\xi_2)\end{aligned}$$

which take now values on the entire real axis. Clearly

$$\begin{aligned}\xi_1 &= e^{-\frac{\eta_1^2+\eta_2^2}{2}}\\ \xi_2 &= \frac{1}{2\,\pi}\tan^{-1}\left(\frac{\eta_2}{\eta_1}\right)\end{aligned} \tag{5.1}$$

The joint probability distribution of $\boldsymbol{\eta} = (\eta_1\,, \eta_2)$ is

$$p_{\boldsymbol{\eta}}(y_1, y_2) = p_{\boldsymbol{\xi}}(x_1, x_2)\left|\frac{\partial(x_1, x_2)}{\partial(y_1, y_2)}\right| = \frac{e^{-\frac{y_1^2+y_1^2}{2}}}{2\,\pi}$$

whence we conclude that each of the η_i $i = 1, 2$ have Gaussian distribution with zero average and unit variance. The numerical implication is that one we have a reliable code for the uniform distribution we can generate using the change of variable formula a code to generate Gaussian random variables.

References

[1] J. Ford, *How random is a coin toss*, Physics Today **36**: 4047
`https://wiki.helsinki.fi/download/attachments/48862734/ford.pdf`.

[2] P. E. Kloeden, E. Platen, H. Schurz, *"Numerical Solution Of Sde Through Computer Experiments"*, Springer (Universitext) (2003) preview from `http://books.google.com/`.

[3] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery,
Numerical recipes in C: the art of scientific computing
Cambridge University Press (1992) and
`http://www.fizyka.umk.pl/nrbook/bookcpdf.html`,
see also `http://www.nr.com`.