

Simulate Random Numbers Using Halton Sequence

Group 7 Project No.1

1 Introduction about Halton sequence

In statistics, Halton sequences are sequences used to generate points in space for numerical methods such as Monte Carlo simulations. Although these sequences are deterministic, they are of low discrepancy, that is, appear to be random for many purposes. They were first introduced in 1960 and are an example of a quasi-random number sequence. They generalise the one-dimensional van der Corput sequences, consult that article for a precise definition.

2 Example About Halton sequence

Example of Halton sequence used to generate points in $(0, 1) \times (0, 1)$ in R^2 . The Halton sequence is constructed according to a deterministic method that uses coprime numbers as its bases. As a simple example, let's take one dimension of the Halton sequence to be based on 2 and the other on 3. To generate the sequence for 2, we start by dividing the interval $(0,1)$ in half, then in fourths, eighths, etc., which generates

$$\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16}, \dots$$

and to generate the sequence for 3, we divide the interval (0, 1) in thirds, then ninths, twenty-sevenths, etc., which generates

$$\frac{1}{3}, \frac{2}{3}, \frac{1}{9}, \frac{4}{9}, \frac{7}{9}, \frac{2}{9}, \frac{1}{27}, \dots$$

When we pair them up, we get a sequence of points in a unit square:

$$(\frac{1}{2}, \frac{1}{3}), (\frac{1}{4}, \frac{2}{3}), (\frac{3}{4}, \frac{1}{9}), (\frac{1}{8}, \frac{4}{9}), (\frac{5}{8}, \frac{7}{9}), (\frac{3}{8}, \frac{2}{9}), (\frac{7}{8}, \frac{1}{27}), \dots$$

Even though standard Halton sequences perform very well in low dimensions, correlation problems have been noted between sequences generated from higher primes. For example, if we started with the primes 17 and 19, the first 16 pairs of points $(\frac{1}{17}, \frac{1}{19}), (\frac{2}{17}, \frac{2}{19}), (\frac{3}{17}, \frac{3}{19}) \dots (\frac{16}{17}, \frac{16}{19})$, have perfect linear correlation. To avoid this, it is common to drop the first 20 entries, or some other predetermined number depending on the primes chosen. In order to deal with this problem, various other methods have been proposed; one of the most prominent solutions is the scrambled Halton sequence, which uses permutations of the coefficients used in the construction of the standard sequence. Another solution is using leaped Halton, just skipping points. If you use e.g. only each 409th point (also other prime numbers not used in the Halton core sequence are possible), then you can achieve significant improvements.

3 Implementation in codes in R and MATLAB

Our group simulate n random numbers with base b between 0 and 1 using Halton sequence , the R code are as follows:

```

1 HaltonSingle<-function(n,b){
2     n0<-n
3     hn<-0
4     f<-1/b
5     while (n0>0) {
6         n1<-floor(n0/b)
7         r<-n0-n1*b
8         hn<-hn+f*r
9         f<-f/b
10        n0<-n1
11    }
12    hn
13 }
14 HaltonSingle(5,2)
15 haltontotal<-function(n,b){
16     hs<-rep(0,n)
17     for(i in 1:n){
18         hs[i]<-HaltonSingle(i,b)
19     }
20 }
21 hs
22 }
23 haltontotal(5,2)

```

The MATLAB code are as follows:

```

1 function hs = haltontotal(n,b)
2 hs =zeros(n,1);
3 for i=1:n
4 hs(i)=HaltonSingle(i,b)
5 end
6
7 function hn = haltonSingle(n,b)
8 n0=n;
9 hn=0;
10 f=1/b;
11 while (n0>0)
12 n1=floor(n0/b);
13 r=n0-n1*b;
14 hn=hn+f*r;
15 f=f/b;
16 n0=n1;
17 end

```

4 Conclusion

From R , We can get the output results

```
1 > HaltonSingle(5,2)
2 [1] 0.625
3 > haltontotal(5,2)
4 [1] 0.500 0.250 0.750 0.125 0.625
```

Also, In MATLAB, we can get the same results. Then we finish our simulation.