Generation of Multi-dimensional Sobol Sequence

March 2013

last update: June 25, 2016

Suppose we want to generate n points of d-dimensional Sobol quasi-random sequence x. Let $x_{i,j}$ denotes the j-th $(1 < j < d)$ dimension of the i-th $(1 < i < n)$ point in the sequence. To generate a sequence for each dimension, say the j-th dimension, we need to choose a primitive polynomial of some degree p for this dimension (i.e. one polynomial for each dimension) [1]

$$P(x) \equiv x^p + a_1 x^{p-1} + a_2 x^{p-2} + \cdots + a_{p-1} x + 1 \tag{1}$$

where the coefficient $a_1, \cdots, a_{p-1} \in \{0,1\}$.

The direction number v_k must be prepared. Each direction number v_k is a binary fraction and can be expressed as

$$v_k = \frac{m_k}{2^k} \ \forall \ k = 1,2,\cdots,h \tag{2}$$

where $h = \lceil \log_2 n \rceil$ and m_k must be odd and $0 < m_k < 2^k$. The first p of $m_k,\ k = 1,\cdots,p$ must be given. The subsequent $m_k,\ k = p+1,\cdots,h$ can be calculated by the following recursive relationship

$$\begin{gathered} m_k = 2a_1 m_{k-1} \oplus 2^2 a_2 m_{k-2} \oplus \cdots \oplus 2^{p-1} a_{p-1} m_{k-p+1} \oplus 2^p m_{k-p} \oplus m_{k-p} \\ \forall\ k = p+1,\cdots,h \end{gathered} \tag{3}$$

where $\oplus$ is the bit-by-bit "*exclusive or*" operator. Dividing (3) by 2^k on both sides, the v_k can written in a recursive form

$$v_k = a_1 v_{k-1} \oplus a_2 v_{k-2} \oplus \cdots \oplus a_{p-1} v_{k-p+1} \oplus v_{k-p} \oplus \frac{v_{k-p}}{2^p}, \qquad \forall\ k = p+1,\cdots,h \tag{4}$$

Then we have the Sobol sequence for the j-th dimension

$$x_i = i_1 v_1 \oplus i_2 v_2 \oplus \cdots \qquad \text{and} \qquad i = (\cdots i_3 i_2 i_1)_2 \tag{5}$$

where $i_k \in \{0,1\}$ denotes the k-th bit from the right when i is written in a binary form.

The above equation can be replaced with a more efficient Gray code implementation. The Gray code of an integer i is defined as

$$g_i = i \oplus \left\lfloor \frac{i}{2} \right\rfloor = (\cdots i_3 i_2 i_1)_2 \oplus (\cdots i_4 i_3 i_2)_2 \tag{6}$$

It has the property that the binary g_{i+1} and g_i differ in only the z-th bit, where i_z is the first zero bit in $i = (\cdots i_z \cdots i_2 i_1)_2$. In fact, Gray code is simply a reordering of the nonnegative integers within every block of 2^m, $m = 0,1,\cdots$. With the Gray code implementation, we simply obtain the sequence in a different order while still preserving their uniformity properties. Hence, instead of using (5), we can generate the Sobol sequence using

$$x_i = g_{i,1} v_1 \oplus g_{i,2} v_2 \oplus \cdots \qquad \text{and} \qquad g_i = \left(\cdots g_{i,3} g_{i,2} g_{i,1}\right)_2 \tag{7}$$

Since g_{i+1} and g_i differ in only the z-th bit, for a more efficient implementation, we can generate the sequence recursively using

$$x_{i+1} = x_i \oplus v_z \qquad \text{where} \qquad z = \log_2(g_{i+1} \oplus g_i) + 1 \tag{8}$$

In real implementation the binary fractions v_k , $k = 1,2,\cdots,h$ are represented by integers, that is

$$\hat{v}_k = 2^B v_k \tag{9}$$

where B is the maximum number of bits used to represent an integer number in a computer language. For example, in Excel/VBA, the $B = 31$ (because VBA does not support unsigned long integers). With this change, the computed x_i is actually an integer, which can then be divided by 2^B to convert to a decimal number.

REFERENCES

1. http://web.maths.unsw.edu.au/~fkuo/sobol/