4. THE ALIAS METHOD.

4.1. Definition.

Walker (1974, 1977) proposed an ingenious method for generating a random variate X with probability vector $p_0, p_1, \ldots, p_{K-1}$ which requires a table of size $O(K)$ and has a worst-case time that is independent of the probability vector and K. His method is based upon the following property:

Theorem 4.1.

Every probability vector $p_0, p_1, \ldots, p_{K-1}$ can be expressed as an equiprobable mixture of K two-point distributions.

Proof of Theorem 4.1.

We have to show that there are K pairs of integers $(i_0, j_0), \ldots, (i_{K-1}, j_{K-1})$ and K probabilities $q_0, \ldots, q_{K-1}$ such that

$$p_i = \frac{1}{K} \sum_{l=0}^{K-1} (q_l I_{[i_l=i]} + (1-q_l) I_{[j_l=i]}) \qquad (0 \le i < K) .$$

This can be shown by induction. It is obviously true when $K=1$. Assuming that it is true for $K<n$, we can show that it is true for $K=n$ as follows. Choose the minimal p_i. Since it is at most equal to $\frac{1}{K}$, we can take i_0 equal to the index of this minimum, and set q_0 equal to Kp_{i_0}. Then choose the index j_0 which corresponds to the largest p_i. This defines our first pair in the equiprobable mixture. Note that we used the fact that $\frac{(1-q_0)}{K} \le p_{j_0}$ because $\frac{1}{K} \le p_{j_0}$. The other $K-1$ pairs in the equiprobable mixture have to be constructed from the leftover probabilities

$$p_0, \ldots, p_{i_0} - p_{i_0}, \ldots, p_{j_0} - \frac{1}{K}(1-q_0), \ldots, p_{K-1}$$

which, after deletion of the i_0-th entry, is easily seen to be a vector of $K-1$ nonnegative numbers summing to $\frac{K-1}{K}$. But for such a vector, an equiprobable mixture of $K-1$ two-point distributions can be found by our induction hypothesis. ∎

To turn this theorem into profit, we have two tasks ahead of us: first we need to actually construct the equiprobable mixture (this is a set-up problem), and then we need to generate a random variate X. The latter problem is easy to solve. Theorem 4.1 tells us that it suffices to throw a dart at the unit square in the plane and to read off the index of the region in which the dart has landed.

The unit square is of course partitioned into regions by cutting the x-axis up into K equi-spaced intervals which define slabs in the plane. These slabs are then cut into two pieces by the threshold values q_l. If

$$p_i = \frac{1}{K}\sum_{l=0}^{K-1}(q_l I_{[i_l=i]} + (1-q_l)I_{[j_l=i]}) \qquad (0 \le i < K),$$

then we can proceed as follows:

The alias method

Generate a uniform [0,1] random variate U. Set $X \leftarrow \lfloor KU \rfloor$. Generate a uniform [0,1] random variate V.

IF $V < q_X$

THEN RETURN i_X

ELSE RETURN j_X

Here one uniform random variate is used to select one component in the equiprobable mixture, and one uniform random variate is used to decide which part in the two-point distribution should be selected. This unsophisticated version of the alias method thus requires precisely two uniform random variates and two table look-ups per random variate generated. Also, three tables of size K are needed.

We observe that one uniform random variate can be saved by noting that for a uniform [0,1] random variable U, the random variables $X = \lfloor KU \rfloor$ and $V = KU - X$ are independent: X is uniformly distributed on $0, \ldots, K-1$, and the latter is again uniform [0,1]. This trick is not recommended for large K because it relies on the randomness of the lower-order digits of the uniform random number generator. With our idealized model of course, this does not matter.

One of the arrays of size K can be saved too by noting that we can always insure that $i_0, \ldots, i_{K-1}$ is a permutation of $0, \ldots, K-1$. This is one of the duties of the set-up algorithm of course. If a set-up gives us such a permuted table of i-values, then it should be noted that we can in time $O(K)$ reorder the structure such that $i_l = l$, for all l. The set-up algorithm given below will directly compute the tables j and q in time $O(K)$ and is due to Kronmal and Peterson (1979, 1980):

Set-up of tables for alias method

Greater $\leftarrow \emptyset$, Smaller $\leftarrow \emptyset$ (Greater and Smaller are sets of integers.)
FOR l :=0 TO $K-1$ DO
    $q_l \leftarrow K p_l$.
    IF $q_l < 1$
        THEN Smaller $\leftarrow$ Smaller $+\{l\}$.
        ELSE Greater $\leftarrow$ Greater $+\{l\}$.
WHILE NOT EMPTY (Smaller) DO
    Choose $k \in$ Greater ,$l \in$ Smaller [q_l is finalized].
    Set $j_l \leftarrow k$ [j_l is finalized].
    $q_k \leftarrow q_k - (1-q_l)$.
    IF $q_k < 1$ THEN Greater $\leftarrow$ Greater - $\{k\}$, Smaller $\leftarrow$ Smaller $+\{k\}$.
    Smaller $\leftarrow$ Smaller $-\{l\}$.

The sets Greater and Smaller can be implemented in many ways. If we can do it in such a way that the operations "grab one element", "is set empty ?", "delete one element" and "add one element" can be done in constant time, then the algorithm given above takes time $O(K)$. This can always be insured if linked lists are used. But since the cardinalities sum to K at all times, we can organize it by using an ordinary array in which the top part is occupied by Smaller and the bottom part by Greater. The alias algorithm based upon the two tables computed above reads:

Alias method with two tables

Generate a random integer X uniformly distributed on 0, . . . , $K-1$.
Generate a uniform [0,1] random variate V.
IF $V \leq q_X$
    THEN RETURN X
    ELSE RETURN j_X

Thus, per random variate, we have either 1 or 2 table accesses. The expected number of table accesses is

$$1+\frac{1}{K}\sum_{l=0}^{K-1}(1-q_l) .$$