Scanning Algorithms and some of their properties*

Tucker McElroy
U.S. Bureau of the Census

Dimitris N. Politis
University of California, San Diego

Definition 0.1 *A* scan *is a collection of n block-subsamples of the sequence $X_1, \ldots, X_n$ with the following two properties: (a) within each scan there is a single block of each size $k = 1, \ldots, n$; and (b) those n blocks are nested, i.e., the block of size k_1 can be found as a stretch within the block of size k_2 when $k_1 \leq k_2$.*

A randomly selected scan can be picked by the following inductive algorithm:

Algorithm A: Forward Random Scan

1. (Starting point) Let the first block be (X_J) with J chosen randomly (and equiprobably) from the set $\{1, \ldots, n\}$.

2. (Getting the block of size $k+1$ from the block of size k for $k < n$.) Suppose that $(X_i, \ldots, X_{i+k-1})$ has been chosen as the block of size k; then, there are only two possible candidates for the block of size $k+1$, namely $(X_i, \ldots, X_{i+k})$ and $(X_{i-1}, \ldots, X_{i+k-1})$.

*For use in connection with the papers: 'Computer-intensive rate estimation, diverging statistics, and scanning' by T.McElroy and D.N.Politis (2007a), and 'A fine-tuned estimator of a general convergence rate' by T.McElroy and D.N.Politis (2007b).

If $i = 1$, then the block of size $k+1$ must necessarily be $(X_i, \ldots, X_{i+k})$; i.e., growth must be from the right.

If $i + k - 1 = n$, then the block of size $k + 1$ must necessarily be $(X_{i-1}, \ldots, X_{i+k-1})$; i.e., growth must be from the left.

Otherwise, toss a fair coin to choose between the two candidates for the block of size $k + 1$.

Alternatively, one can use a backward recursion:

ALGORITHM B: BACKWARD RANDOM SCAN

1. (STARTING POINT) The algorithm begins with the last block which is $(X_1, \ldots, X_n)$ always.

2. (GETTING THE BLOCK OF SIZE $k - 1$ FROM THE BLOCK OF SIZE k FOR $k > 1$.) Suppose that $(X_i, \ldots, X_{i+k-1})$ has been chosen as the block of size k; then, there are only two possible candidates for the block of size $k - 1$, namely $(X_i, \ldots, X_{i+k-2})$ and $(X_{i+1}, \ldots, X_{i+k-1})$. Toss a fair coin to choose between those two candidates for the block of size $k - 1$.

The probabilistic properties of the two algorithms are quite different. In particular, Algorithm A seems to favor moving towards the edges (i.e., X_1 or X_n). In other words, although the beginning steps of Algorithm A are quite random, the last steps will (with high probability) look like the last steps of either the direct or the reverse scan that were previously mentioned. By contrast, Algorithm B favors the center of the data set, i.e., points with time index close to $[n/2]$; as a matter of fact, under Algorithm B the probabilities associated with blocks of size one are binomial and not equiprobable as they are under Algorithm A. We formalize the above discussion in the

following proposition. To define it, let $B_i^k = (X_i, \ldots, X_{i+k-1})$, i.e., B_i^k for $i = 1, \ldots, n-k+1$ are all the possible blocks of size k.

Proposition 0.1 *Let P_A and P_B denote the probability mechanisms induced by Algorithms A and B respectively. Also let $scan_A^k$ denote the block of size k that is found in a scan generated via Algorithm A; similarly, define $scan_B^k$. Then, for $1 \leq k < n$ we have:*

$$P_A\{scan_A^k = B_i^k\} = \frac{1}{n} \quad for \;\; 1 < i < n-k+1,$$

$$and \quad P_A\{scan_A^k = B_1^k\} = P_A\{scan_A^k = B_{n-k+1}^k\} = \frac{k+1}{2n}.$$

In addition,

$$P_B\{scan_B^k = B_i^k\} = \tbinom{n-k}{i-1} \left(\frac{1}{2}\right)^{n-k} \quad for \;\; 1 \leq i \leq n-k+1.$$

Proof of Proposition 0.1. The proof is immediate by a (recursive) probability tree diagram going forward for Algorithm A and backward for Algorithm B. The binomial coefficients in $P_B\{scan_B^k = B_i^k\}$ manifest themselves via Pascal's triangle. □

Observe that the probability of selecting an end block (i.e., either the first or the last) of size k at the kth step of Algorithm A is $(k+1)/(2n) + (k+1)/(2n) = (k+1)/n$ that tends to 1 as k increases. After selecting one of the two end blocks, Algorithm A proceeds deterministically, growing the blocks in a one-sided way; thus, the last steps will likely look like the last steps of either the direct or the reverse scan. By contrast, Algorithm B favors the blocks of size 1 that are near the middle point $[n/2]$ (within $\pm\sqrt{n}$) according to the binomial distribution given in Proposition 0.1 (with $k = 1$).

Ideally, our objective is to be able to generate scans in a very 'random' manner; ideally, we would like to have an algorithm that implies a uniform (or close to uniform) distribution on the set of all scans. It seems that neither Algorithm A nor B are sufficiently 'random' but, since they represent opposing trends, a compromise between the two is given below.

Algorithm A(f): Forward Weighted Random Scan

1. (Starting point) Let the first block be (X_j) with probability $f(j)$ where f is a discrete probability distribution on the set $\{1, \ldots, n\}$.

2. (Getting the block of size $k+1$ from the block of size k for $k < n$.) Same as for Algorithm A.

If $f(j) = 1/n$, for all j, then the above reduces to Algorithm A. To achieve a high degree of 'randomness', e.g., a degree of equiprobability among blocks of size k, it is suggested that f is taken as a symmetric distribution with mode at the center, i.e., around the point $[n/2]$. It is unclear, however, how peaked the f distribution should be. The two extremes are given by the uniform $f(j) = 1/n$ of Algorithm A, and the ultra-peaked distribution that assigns probability 1 to the middle point $X_{(n+1)/2}$ (say, n is odd); but the latter should be avoided as it seriously downweights (or even outright excludes) many of the blocks having size $k < n/2$.

As a matter of fact, even the binomial $f(j) = \binom{n-1}{j-1}\left(\frac{1}{2}\right)^{n-1}$ of Algorithm B is too concentrated around the middle of the data having an effective range of $\pm\sqrt{n}$ around the center; it is also to be avoided. Therefore, it is suggested that f should be a compromise somewhere in between the uniform $f(j) = 1/n$ and the binomial $f(j) = \binom{n-1}{j-1}\left(\frac{1}{2}\right)^{n-1}$.

Algorithm B—although computationally efficient—favors the 'center' of the data, and thus does not give the desired variety. Algorithm A(f)—with

a carefully chosen f— is preferrable but is quite more computer-intensive. Nevertheless, in the case of weakly dependent, stationary data, Algorithm B can be modified to yield very 'random' scans while maintaining its computational efficiency. In what follows, we present this modified (circular) version of Algorithm B stressing that it can only be employed in the case of weakly dependent, stationary data $X_1, \ldots, X_n$.

Recall that in the case of weakly dependent, stationary data $X_1, \ldots, X_n$, the technique of circular extension of the data has been considered as early on as Grenander and Rosenblatt (1957). The circular extension allows us to consider time points bigger than n by a $Modulo(n)$ operation. We can use this notion to define a more 'random' version of Algorithm B that is still computationally efficient.

ALGORITHM B′: CIRCULAR EXTENSION FOR WEAKLY DEPENDENT & STATIONARY TIME SERIES

1. Let U be drawn by a Uniform distribution on $\{0, 1, \ldots, n-1\}$, and construct the new series $W_1, \ldots, W_n$ by letting $W_i = X_{(i+U) \ Modulo(n)}$, for $i = 1, \ldots, n$.

2. Apply Algorithm B to the new series $W_1, \ldots, W_n$.

Intuitively, the new series $W_1, \ldots, W_n$ is a *bona fide* version of the data except for time points in the neighborhood of $n + U$ where a discontinuity in the dependence structure is observed. For weakly dependent data, this neighborhood is small and its effect is negligible. However, Algorithm B′ is *not* suitable for 'long-memory' series.

References

[1] Grenander, U. and Rosenblatt, M. (1957). *Statistical Analysis of Stationary Time Series*. Wiley, New York.

[2] T.McElroy and D.N.Politis (2007a). Computer-intensive rate estimation, diverging statistics, and scanning, *Annals of Statistics*.

[3] T.McElroy and D.N.Politis (2007b). A fine-tuned estimator of a general convergence rate, *Australian/New Zealand Journal of Statistics.*