HARVEST REPORT #12

Subject: Use of Table Lookup in Statistical Evaluations of Frequency Counts

By: G. F. Cramer

Date: January 14, 1957

January 14, 1957

HARVEST REPORT #12

Subject: Use of Table Lookup in Statistical Evaluations of Frequency Counts

Making a frequency count of N characters consists of finding the number of times each of the possible characters, C_1, C_2, C_3, ..., C_n occurs among the N characters. It is customary to denote the frequency of occurrence of the character C_i by the symbol f_i.

We are interested in statistics of the form

(1) $S = F(f_1, f_2, f_3, \ldots, f_n, N)$

where the function F may also involve fixed constants in addition to the quantities f_i and N which depend upon the size of the sample being counted. It seems that most of the commonly used statistics can be written in the form

(2) $F = E(N) \sum G(f_i) + H(N)$.

Some of these can be reduced to the still simpler form

(3) $F = \sum G(f_i)$.

The summations are to be taken from $i = 1$ to $i = n$. This suggests the possibility of combining the processes of "counting in memory", table lookup, and accumulation of summands in the accumulator to evaluate F while a stream of characters is passing through a continuous stream register. To do this, it would be necessary to take each character C_i whenever it occurs and use it in a bit assembly unit to form the address of the memory location where f_i is being stored. The f_i would need to be brought out to a bit assembly and used to build up the address of the memory location where $G(f_i + 1) - G(f_i)$ is stored. Finally, $G(f_i + 1) - G(f_i)$ would be fed into the accumulator while f_i would be increased by one and stored in the same memory location from which it came. This assumes that n memory locations are used to store the f_i's while nm memory locations are used for the $[G(f_i + 1) - G(f_i)]$'s, m being the maximum value any f_i is allowed to assume. This might allow the calculation of F and a comparison of F with a threshold value to occur everytime a character C_i is counted if this is desired, but it seems unlikely that it would be necessary to do this comparison so often in most applications.

If F is of the form (2), we could merely form $\sum G(f_i)$ as each character presents itself but only evaluate E(N), H(N), and F from time to time, stopping the streaming while these calculations and the comparison of F with the threshold value are carried out.

There are useful statistics whose evaluation is fundamentally simpler than that of (3). For example, if F has the form

(4) $F = \sum f_i W_i$ weighted sum

where W_i is a constant, we can express it as

(4.1) $F = W_1 + W_1 + \ldots + W_1 + W_2 + W_2 + \ldots + W_2 + \ldots + W_n + W_n + \ldots + W_n$

where W_i occurs f_i times. Thus it is possible to evaluate F very simply by sending W_i to the accumulator each time the character C_i occurs. This does not require that f_i be available in the memory and so involves only a single table lookup instead of both a counting-in-memory process and a table lookup. The somewhat more general statistic

$$F = \sum f_i \mathrm{Log}(nf_i/N)$$

reduces to the form (4) whenever N is large enough so that nf_i/N "stabilizes" or becomes essentially constant with increasing N.

Another statistic which is simple to evaluate is

(5) $F = 1/2 \sum f_i(f_i - 1) = 1/2 \sum (f_i^2 - f_i)$

This can be handled by counting in memory combined with the sending of f_i to the accumulator each time f_i is called out of the memory by the appearance of character C_i in the stream. The reason this works is that

$(f_i + 1)^2 = f_i^2 + 2f_i + 1 = f_i^2 + f_i + (f_i + 1)$ so that

$1/2 \left[(f_i + 1)^2 - (f_i + 1)\right] = 1/2 (f_i^2 - f_i) + f_i$. Finally $f_i + 1$

is sent back to the memory location from which f_i was obtained.

Another simple statistic is

(6) $F = \sum f_i^2$

This can be evaluated each time a character is counted by sending both f_i and $f_i + 1$ to the accumulator while replacing f_i by $f_i + 1$ in the memory.

One should bear in mind that the more general statistic (3) requires both counting and table lookup if it is to be evaluated on a character by character basis while (4), (5), and (6) can be handled more simply only because of the special form of the $G(f_i)$ in each of these cases.

GFC/jh

G. F. Cramer