A Characterization of Binary Codes that Correspond to a Class of Group-Testing Procedures[1]

by Milton Sobel

Technical Report No. 148

November 1970

University of Minnesota
Minneapolis, Minnesota

[1]This research was supported by NSF Grant GP-11021.

A Characterization of Binary Codes that Correspond to a Class of Group-Testing Procedures[1]

by Milton Sobel
University of Minnesota

In group-testing we wish to classify each of N units as good or defective. We have available a mechanism of testing any subset of size x $(1 \leq x \leq N)$ but a test on x units has only two possible disjoint outcomes: either

i) all x are good or

ii) at least one of the x is defective. (We don't know which of the x are defective and for $x > 1$ we don't know how many of the x are defective.)

Our goal is find a rational method (or strategy) of testing that keeps down in some sense the number of tests required to classify all the N units. The basic binomial formulation with known p (and $q = 1 - p$) assumes that the condition of each unit is a Bernoulli random variable with a common probability p of being defective and $q = 1 - p$ of being good. Moreover, these random variables are assumed to be mutually independent. The basic goal for this formulation is to find a strategy that will minimize the expected number of tests $E\{T\}$ required to classify all N units; this strategy can vary as a function of p (or q).

Several procedures for this problem are considered in [6] and related problems (e.g., the case of unknown p) are considered in [1], [2], [3] [4], [5], and [7]. In [6] it was pointed out that a certain procedure R_1 (based on a finite initial number N of units) always deals with at most

[1]This research was supported by NSF Grant GP-11021.

2 sets of unclassified units and never mixes these to form the next test group. One set is the defective set of size m, which is known to contain at least 1 defective unit, and the other is a binomial set. The procedure R_1 is therefore said to be non-mixing (NM) and, in fact, is the optimal procedure in this subclass. The procedure R_1 also has the DB property, namely that when $m \geq 2$ the defective set is searched before testing units in the binomial set.

In this paper we consider (not just procedure R_1 alone but) the class of all non-mixing group-testing procedures with the DB property. We refer to these as NMDBGT (or simply as H) procedures and the notation GT-procedure will be used for any group-testing procedure. It should be noted that inference is used whenever possible in all group-testing procedures. Procedures which allow mixing are considered in [3] and [4]. The improvement due to mixing is i) generally small, ii) available only for q close to one and iii) is accomplished at the cost of complications in the instructions required to carry out the procedure.

For convenience, all GT-codes are written in alphabetic (or dictionary) ordering using the digits 0 and 1 as the letters of the alphabet. Since the nature of group-testing procedures with the DB property is to search for the first defective and pass all good units found in the process, it would be quite unnatural to write the GT-codes in any ordering other than alphabetic.

It is easily seen (and was explicitly pointed out in [4]) that every GT-procedure can be written as a binary code; we arbitrarily associate 0 with the result i) above and 1 with the result ii). In fact it is easy to show (see Appendix of [4]) that the resulting code is an exhaustive

code, i.e., a code with the so-called prefix property (i) no code word is the prefix of any other code word and, if N_j ($j = 1, 2, \ldots, J$) are the lengths of the code words, with the property (ii)

$$(1.1) \qquad \sum_{j=1}^{J} 2^{-N_j} = 1.$$

It was also seen in [4] that an exhaustive code does not necessarily correspond to any group-testing procedure. For example, the exhaustive code with four words

(1.2) 0, 100, 101, 11

does not correspond to any group-test. In fact, four words indicates two units to classify and the only 2 group-tests possible are (A)'1-at-a-time'

(1.3) 00, 01, 10, 11

and (B) 'start with 2 units'

(1.4) 0, 10, 110, 111.

Our goal in this paper is to find a necessary and sufficient condition that any given exhaustive code C (which we can assume is written alphabetically) is an H-code, i.e., that it corresponds to an NMDBGT (or an H-) procedure.

In [4], several necessary properties of a GT-code were found but the problem of finding a sufficient condition for an exhaustive code to be a GT-code (or an H-code) is more difficult.

2. Structural Analysis of the H-code.

Any group-testing procedure for classifying N units has exactly 2^N endpoints (or terminal vertices) corresponding to the 2^N possible

states of nature, if each unit can only be either good or defective. Hence the corresponding code has exactly 2^N code words, each word consisting of zeros and ones. Moreover, the expected number of tests coincides with the expected length of the code words, or as it is usually called, the cost of the code.

It should be noted that the H-procedure assumes that the units are randomized and ordered at the outset only; subsequent randomization within the defective set or within the binomial set (without mixing the two sets) do not affect the procedure or its analysis and hence we can disregard them.

Consider any exhaustive code with 2^N code words and assume it is written alphabetically. Break up the code words into one word W_0 with only zeros (which we call the root word) and the remaining set W of words that contain the digit 1 at least once. Using the alphabetic ordering, we break up the $2^N - 1$ code words in W into ordered subsets S of sizes $2^0, 2^1, \ldots, 2^{N-1}$. Let r (with $1 \leq r \leq N$) denote the number of zeros in the root word W_0 and let x_1 denote the number of units put in the first test. If the first digit is changed to 1 then it means that the H-procedure searched for a defective in the positions 1 through x_1 and, by its nature, if the first defective is in position y_1 $(1 \leq y_1 \leq x_1)$, then we return to a binomial (or so-called H) situation with $y - 1$ units classified as good and one unit classified as defective. Hence the code words that start with the digit 1 consist exactly of certain subsets in S of total size

$$(2.1) \qquad 2^{N-1} + 2^{N-2} + \ldots + 2^{N-x_1} = 2^N - 2^{N-x_1}.$$

All these words are in a terminal collection of the original subsets S

mentioned above; namely, of the original (ordered) N subsets, we are now dealing with the last x_1 subsets. Consider any one of these x_1 subsets. The words in this subset have in common a certain number of digits (after the initial digit 1), which indicate how the procedure found out that the first defective was in that subset. We extract these common digits for each subset (keeping the order of the subsets) and use it to write down (and define) the prefix code corresponding to the first zero in W_0.

We also have to consider the second and subsequent zeros in the root word W_0. For the j^{th} zero in W_0 $(1 \leq j \leq r)$, we consider all the code words of the form $(0^{j-1}1\ldots)$ and these again consist exactly of a successive collection of subsets in the original collection S. In fact, the number of these code words must be

$$(2.2) \qquad 2^{N-1-S_{j-1}} + 2^{N-2-S_{j-1}} + \ldots + 2^{N-S_j} = 2^{N-S_{j-1}} - 2^{N-S_j},$$

where $S_j = x_1 + \ldots + x_j$, $S_0 = 0$ and x_j is the number of units used in the j^{th} test, if the first $j - 1$ tests are all successful. Each of the subsets on the left side of (2.2) has common digits (after the initial $(0^{j-1}1,)$) that indicate how the procedure found the first defective, if the first defective was in the y^{th} position $(S_{j-1} < y \leq S_j)$. We use these to form the prefix code corresponding to the j^{th} zero in W_0. The set of these r prefix codes describes the search for the first defective, if any was present.

Each word in any of the r prefix codes is repeated in the original code C; the number of times it is repreated is a power of 2 since we get back to an H-situation after a defective is found. In fact, the

remaining (or suffix) code, after suppressing the j^{th} prefix code and its prefix $(0^{j-1}1\cdot)$, must again be an H-code. It should be noted that, for the latter to be correct in all generality, we must accept the empty code $\{\emptyset\}$ as an H-code. This leads to no inconsistency since the empty code is an exhaustive code, satisfying the prefix property and (1.1); we take the empty code to consist of one code word $\{\emptyset\}$ of length 0.

To illustrate the above suppose we start with 4 units and take $x_1 = 3$ and $x_2 = 1$. We use the notation H(n) and G(m, n) from [6]. For G(3, n) and G(2, n) we will take $x' = 1$ (from the defective set) for any total n. For H(n) we will take $x'' = n$ for $n = 2, 3$. Then the tree takes the form

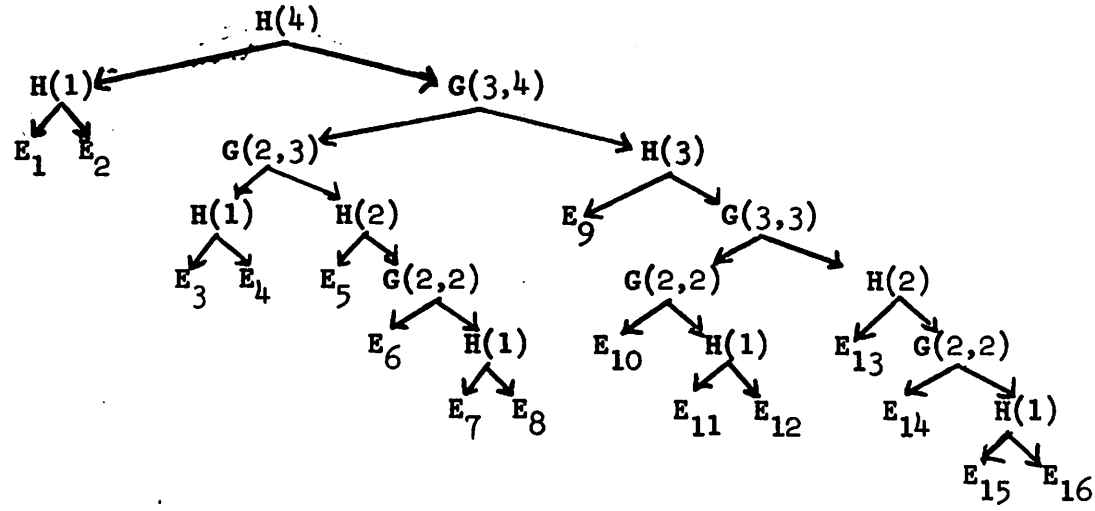

Figure 1: A group-testing procedure (without mixing) for N = 4 units.

The complete H-code for the procedure in Figure 1 is

(2.3)

	W_0
00	110
01	11100
1000	111010
1001	111011
1010	11110
10110	111110
101110	1111110
101111	1111111.

The E_j $(1 \leq j \leq 16)$ in Figure 1 indicate the 16 possible end points corresponding to 16 states of nature for 4 units. The lines in (2.3) indicate the break up into W_0 and subsets of size 2^0, 2^1, 2^2 and 2^3 code words. Corresponding to the first 0 in W_0 we have the prefix code

(2.4) 00, 01, 1.

Corresponding to the second 0 in W_0, we have the empty prefix code, indicating that it wasn't necessary to search for that defective unit. We note that the prefix code in (2.4) is exhaustive and alphabetic, but it need not be an H-code. For each code word in (2.4) the associated suffix code is an H-code. For example, the suffix code associated with 01 is the 4-word code {0, 10, 110, 111}, which already appeared in (1.4).

We now state the necessary and sufficient condition for a given exhaustive code C (written alphabetically) to be an H-code. The condition is stated inductively and we assume that (0, 1) and the empty set {∅} are H-codes.

Theorem:

An exhaustive, alphabetic code C is an H-code if and only if

1) it contains 2^N code words for some N,

2) the prefix code corresponding to the $j^{\underline{th}}$ zero in W_0 is an exhaustive code with x_j code words $(j = 1, 2, \ldots, r)$,

3) the suffix code associated with the $i^{\underline{th}}$ word of the $j^{\underline{th}}$ prefix code $(1 \leq i \leq x_j ; j = 1, 2, \ldots, r)$ must be an H-code of size $2^{N-S_{j-1}-i}$,

4) $x_1 + \ldots + x_r = N$.

Proof: It is clear from the single illustration above that any H-procedure will satisfy properties 1), 2), 3) and 4) above. We need only show that any

exhaustive (alphabetic) code can serve as a prefix code, i.e., can tell us how the procedure searched for a defective, once the existence of a defective unit was established. This is shown in a lemma below. Then the structure given in 3) and 4) enables us to associate the given code C with the H-procedure that returns to the binomial (or H-) situation as soon as a defective unit is found.

For this purpose we define a G(m) (or G) procedure as a group-testing procedure for finding a single defective unit among m units, if we know there is at least one defective present. The G-procedure has the property that if a test on x_1 out of m units indicates that a defective unit is present then for $x_1 = 1$ we are through and for $x_1 > 1$ we disregard the $m - x_1$ units and search for a defective unit in the new defective set of size x_1. There are exactly m possible positions for the first defective unit and the corresponding code (which we call a G(m)-code) has exactly m code words. We need the following

Lemma:

The number of exhaustive, alphabetic codes with m code words (m arbitrary) is exactly the number of G(m)-codes.

Proof: It is clear that a G(m)-procedure gives rise to an exhaustive, alphabetic procedure with exactly m code words corresponding to the m possible positions of the first defective unit.

Consider any arbitrary exhaustive alphabetic code C with exactly m code words. We use induction on m. Let x_1 denote the number of code words starting with the digit 1. Associate these with the event that the first defective is among the first x_1 units and, by suppressing the first digit 1, we then have to find which one of these x_1 code words represent

the true state of nature. By the induction hypothesis the total number of possible continuations is the number of $G(x_1)$-codes.

Similarly we associate all the words starting with zero with the event that the first defective is among the last $m - x_1$ units. By suppressing the first digit 0, we then have to find which of these $m - x_1$ code words represent the true state of nature. By the induction hypothesis the total number of possible continuations is the number of $G(m-x_1)$-codes. Note that x_1 can take on all possible values $(1 \leq x_1 \leq m-1)$ in the given code C, that both x_1 and $m - x_1$ are less than m, and that any change in x_1 changes the code as well as the G-procedure.

The lemma clearly holds for $m = 2$ since the only exhaustive, alphabetic code with two code words is {0, 1} and this is also the only G(2)-procedure.

Putting these together, the lemma follows. In fact, we have shown a one-to-one correspondence between the exhaustive, alphabetic codes with m code words and the G(m)-procedures.

3. An Example Illustrating the Use of the Result .

Given the 32 word code C,

(3.1)

00	1100	11100	1111100
010	110100	111010	11111010
0110	1101010	1110110	111110110
0111	1101011	1110111	111110111
1000	110110	1111000	11111100
1001	1101110	1111001	11111101
1010	11011110	1111010	11111110
1011	11011111	1111011	11111111,

check to see if it is an H code. The major partition into W_0 and powers of 2 is already shown in (3.1) by horizontal lines. The prefix code for the first 0 in W_0 is easily seen to be {0, 10, 11} so that $x_1 = 3$,

and for the second 0 in W_0 to be {0, 1} so that $x_2 = 2$. These add to 5 and there are $2^5 = 32$ code words, so that properties 1, 2 and 4 hold. To check property 3 we first note that the suffix code corresponding to 0 in the first prefix code is (1.3); we consider the other 2 suffix codes later. Corresponding to 0 in the second prefix code we have the empty code {∅} and corresponding to 1 we have {0, 1}, both of which are H-codes.

The suffix code corresponding to 10 in the first prefix code has $8 = 2^3$ code words and can be analyzed as an H-code by starting afresh with all 4 properties. This time there is only one prefix code given by {00, 01, 1} and $x_1' = 3$ and we need only look at the three suffix codes. One is the empty code {∅}, one is {0, 1} and one is the code in (1.4).

Returning to the suffix code corresponding to the word 11 in the first prefix code, we now have a 16 word code to check; it starts with 00 in the $17^{\underline{th}}$ word in (3.1). The two prefix codes are both {0, 1}, so that $x_1'' = x_2'' = 2$ and we have only to check the four suffix codes. Corresponding to the first prefix code we see (1.3) for the 0 and we have to check the last 8 code words of C starting with 00 in the $25^{\underline{th}}$ code word. Corresponding to the second prefix code we obtain the empty code {∅} and {0, 1}.

To check the last 8 words of C, starting with column 6, we find that the two prefix codes are the empty set {∅} and {0, 1}. The three suffix codes are (1.3) for {∅}, the empty code {∅} for the 0 and the code {0, 1} for the 1. Since these are all H-codes the checking is complete. Hence the given code C is an H-code.

REFERENCES

[1] Kumar, S. (1970). Multinomial group-testing. SIAM Jour. App. Math. 19 340-350.

[2] Kumar, S. and Sobel, M. (1970). Group-testing with at most c tests for finite c and $c \to \infty$. Technical Report No. 146, Dept. of Statistics, Univ. of Minnesota.

[3] Sobel, M. (1960). Group-testing to classify all defectives in a binomial sample. Information and Decision Processes, ed. R. E. Machol. McGraw-Hill, 127-161.

[4] Sobel, M. (1968). Optimal group-testing. Proceedings of the Colloquium on Information Theory Organized by the Bolyai Mathematical Society. Debrecen, Hungary, 411-488.

[5] Sobel, M. (1968). Binomial and Hypergeometric group-testing. Studia Sci. Math. Hungar. 3 19-42.

[6] Sobel, M. and Groll, P. A. (1959). Group-testing to eliminate efficiently all defectives in a binomial sample. Bell System Tech. Jour. 38 1179-1252.

[7] Sobel, M. and Groll, P. A. (1966). Binomial group-testing with an Unknown Proportion of Defectives. Technometrics 8 631-656.