Acta Informatica 3, 347—355 (1974)

Performance Bounds on the Splitting Algorithm for Binary Testing

M. R. Garey and R. L. Graham

Received December 5, 1973

Summary. In machine fault-location, medical diagnosis, species identification, and computer decisionmaking, one is often required to identify some unknown object or condition, belonging to a known set of M possibilities, by applying a sequence of binary-valued tests, which are selected from a given set of available tests. One would usually prefer such a testing procedure which minimizes or nearly minimizes the expected testing cost for identification. Existing methods for determining a minimal expected cost testing procedure, however, require a number of operations which increases exponentially with M and become infeasible for solving problems of even moderate size. Thus, in practice, one instead uses fast, heuristic methods which hopefully obtain low cost testing procedures, but which do not guarantee a minimal cost solution. Examining the important case in which all M possibilities are equally likely, we derive a number of cost-bounding results for the most common heuristic procedure, which always applies next that test yielding maximum information gain per unit cost. In particular, we show that solutions obtained using this method can have expected cost greater than an arbitrary multiple of the optimal expected cost.

Introduction

In many situations[1], one is often required to identify some unknown object or condition belonging to a known set of M possibilities by applying a sequence of binary (i.e., two-valued) tests. Ordinarily, one would prefer such a testing procedure which minimizes the expected cost for identification. However, at present the only known methods for obtaining general binary testing procedures with minimum expected cost require an exponential (in M) number of operations and, hence, are not practical for problems of even moderate size. In practice one tries to use fast, heuristic techniques which, though not guaranteeing a minimal cost solution, yield procedures of relatively low expected cost. In this paper we investigate the most common heuristic of this type, the so-called *splitting algorithm*. In this procedure, the next test is always chosen to be a test which maximizes the information gain per unit cost. Our investigation will be restricted to the important case in which all the M possibilities are equally likely[2]. However, even with this restriction of uniformity, it will be shown that it is possible for a solution obtained using the splitting algorithm to have an expected cost which differs from the optimal expected cost by an arbitrarily large factor.

1 e.g., machine fault location, medical diagnosis, computer decisionmaking and species identification.

2 The general case will be treated in a future paper [7].

Notation and Definitions

A *binary identification problem* consists of the following:

(a) A set $\Theta = \{\theta_1, \ldots, \theta_M\}$ of M *objects* which are the possibilities for the single unknown object θ;

(b) A corresponding set $\{p_1, \ldots, p_M\}$ of M *object probabilities* satisfying $p_i > 0$ and $\sum_{i=1}^{M} p_i = 1$ (since we are assuming just one unidentified object) where p_i is the probability that $\theta = \theta_i$;

(c) A set $\mathcal{T} = \{T_1, \ldots, T_N\}$ of N distinct subsets of Θ known as *tests*. The test T_i is applied by asking: "Is the unknown object θ an element of T_i?" The answer is either yes (1) or no (0). We shall always assume that the set of tests $\mathcal{T}$ is sufficient to distinguish each of the $\theta_i \in \Theta$, i.e., for any θ_i, θ_j, $i \neq j$, there exists a test T_k such that either $\theta_i \in T_k$, $\theta_j \notin T_k$ or $\theta_i \notin T_k$, $\theta_j \in T_k$.

(d) A corresponding set $\{C_1, \ldots, C_N\}$ of *test costs* where $C_i \geqq 0$ is the cost incurred whenever test T_i is applied. For the remainder of the paper *we shall assume that* $C_i = 1$ *for all* i.

A *testing procedure* for a binary identification problem is a set of rules for deciding (on the basis of previous test results) which test in $\mathcal{T}$ to perform next in the testing process. The unknown object is said to be *identified* if there remains just a single one of the M possibilities consistent with the outcomes of the applied tests. The *cost of a testing procedure*, under the assumption of unit test costs, is defined to be the expected number of tests required for identification of the unknown object, i.e., $\sum_{i=1}^{M} p_i N_i$, where N_i is the number of tests required by the testing procedure when θ_i is the unknown object. Finally, an *optimal* testing procedure is one which has the minimum cost over all testing procedures which use tests from $\mathcal{T}$.

In [5, 6], a dynamic programming approach is described for constructing optimal testing procedures. However, this method can require as many as $N \cdot 2^M$ operations and is therefore practical only for small M.

A heuristic method, which we shall call the "splitting algorithm", frequently occurs in the literature (cf. [2–4, 9, 10, 12]). Although the details of the various descriptions differ slightly depending upon their intended generality, when applied to binary identification problems with unit test costs they all reduce to the following algorithm. Suppose that after the first k tests have been applied, the unknown object θ is known to lie in the subset $S_k \subseteq \Theta$. The $(k+1)^{\text{st}}$ test is now chosen to be that test T_j for which the sums

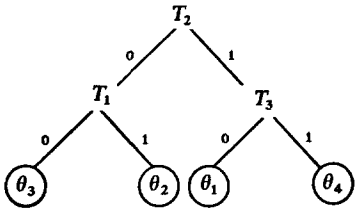
$$\sum_{\theta_i \in S_k \cap T_j} p_i \quad \text{and} \quad \sum_{\theta_i \in S_k - T_j} p_i$$

are most nearly equal[3].

Example. Suppose $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$ with $p_1 = 0.1$, $p_2 = 0.2$, $p_3 = 0.3$, $p_4 = 0.4$, $\mathcal{T} = \{T_1, T_2, T_3\}$ with $T_1 = \{1, 2\}$, $T_2 = \{1, 4\}$, $T_3 = \{4\}$, and all $C_i = 1$.

3 Ties are broken arbitrarily.

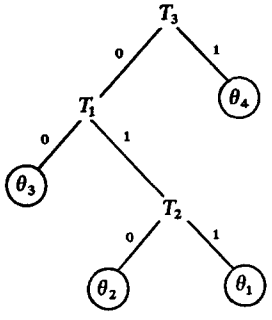
The splitting algorithm for this example is illustrated by the following diagram.



The expected cost of this procedure is

$$\sum_{i=1}^{4} p_i N_i = (0.1)2 + (0.2)2 + (0.3)2 + (0.4)2 = 2.$$

On the other hand, the optimal testing procedure using $\mathscr{T}$ is shown below.



The expected cost for this procedure is

$$\sum_{i=1}^{4} p_i N_i = (0.1)3 + (0.2)3 + (0.3)2 + (0.4)1 = 1.9.$$

The rationale underlying the splitting algorithm is that in the case of unit cost tests, each test is chosen to maximize the information gain per unit cost (and so, is an example of a so-called "greedy" algorithm). However, as the example shows, the splitting algorithm does not necessarily produce an optimal procedure. This is because the use of certain high information gain tests initially may split Θ into subsets which subsequently can only be split with tests giving a very low information gain, while choosing tests with somewhat less information gain initially may split Θ into subsets which can then be further split with high information gain. Thus, the relationships between the available tests may have an important effect upon the performance of the splitting algorithm. In fact, we shall show that even for the case in which all p_i are equal, sets of tests $\mathscr{T}$ exist for which the ratio of the expected cost of the splitting algorithm to the expected cost of an optimal testing procedure is arbitrarily large.

Known Results for Complete Test Sets

The set of tests $\mathcal{T}$ for $\Theta=\{\theta_1, \ldots, \theta_M\}$ is called *complete* if for any $S \subseteq \Theta$, there is a $T \in \mathcal{T}$ such that either $T=S$ or $\Theta - T = S$. The following result shows that the splitting algorithm performs quite well when $\mathcal{T}$ is complete.

Theorem (Shannon [13]; see also [1]). Assume $\mathcal{T}$ is a complete set of unit cost binary tests for Θ. Let K_0 denote the expected cost of an optimal testing procedure and let K^* denote the expected cost of any testing procedure constructed using the splitting algorithm. Then

$$K^* < K_0 + 1.$$

Sandelius [11] has noted that in the special case in which $\mathcal{T}$ is complete and all the p_i are equal, the splitting algorithm is, in fact, an optimal testing procedure. Huffman [8] has derived a simple algorithm which efficiently constructs an optimal testing procedure whenever $\mathcal{T}$ is complete. Although Huffman originally produced his algorithm in the context of optimal variable-length binary codes, Zimmerman [14] later rediscovered the same algorithm in connection with binary testing procedures.

A Lower Bound. We now show that, in contrast to the preceding result, the splitting algorithm may perform quite poorly when the set $\mathcal{T}$ of available tests is not complete. Given a set of equiprobable objects and a set of binary tests $\mathcal{T}$ for those objects, let K_0 be the expected number of tests required by an optimal testing procedure using $\mathcal{T}$. Let K^* be the *minimum* expected number of tests taken over all testing procedures which can be obtained using the *splitting algorithm*. Similarly, let K' be the *maximum* expected number of tests taken over all testing procedures which can be obtained using the splitting algorithm.

Define $R^*(M)$ to be the maximum value achieved by the ratio K^*/K_0 taken over all sets $\mathcal{T}$ and all Θ with at most M equiprobable objects.

Our first result concerns the behavior of $R^*(M)$.

Theorem 1. For $M > 2$,

$$R^*(M) > \frac{1}{10}\left[\frac{\log_2 M}{\log_2 \log_2 M}\right].$$

Proof. For each pair of positive integers $m < n$, we describe a binary testing problem $P(m, n)$. The set of objects Θ will have $M' = 2^n$ objects. Each $\theta_i \in \Theta$ will be specified by a binary n-tuple $(b_1, \ldots, b_n)$ with $b_i = 0$ or 1 for $1 \leqq i \leqq n$. The parameter m will determine the structure of the set of tests $\mathcal{T}$, which will consist of two types of tests, the type A tests and the type B tests.

The type A tests, which will be selected by the splitting algorithm, are denoted by $T_A(d_1, \ldots, d_j)$ where j assumes all the values $0, m, 2m, \ldots, [n/m]m$ and n, and, for each j, the d_i range over all 2^j choices of 0's and 1's. The test $T_A(d_1, \ldots, d_j)$ consists of the 2^{n-j} objects $(b_1, \ldots, b_n)$ for which $b_i = d_i$, $1 \leqq i \leqq j$.

The type B tests are denoted by $T_B(d_k, d_{k+1}, \ldots, d_n)$ where k assumes all the values $1, 2, \ldots, n-m$ and, for each k, the d_i range over all 2^{n-k+1} choices of 0's and 1's. The test $T_B(d_k, \ldots, d_n)$ consists of the 2^{k-1} objects $(b_1, \ldots, b_n)$ for which $b_i = d_i$, $k \leqq i \leqq n$.

We first describe a testing procedure which uses only type B tests. The procedure first applies the various tests $T_B(d_{n-m}, \ldots, d_n)$ until the last $m+1$ digits of the unknown object θ have been determined. Once this has been done, the remaining $n-m-1$ digits can be determined using one test apiece by an appropriate choice of tests of the form $T_B(0, d_{n-m}, \ldots, d_n)$, $T(0, d_{n-m-1}, \ldots, d_n)$, $\ldots, T_B(0, d_2, \ldots, d_n)$. This testing procedure is seen to require an expected number of tests equal to

$$K_B = 2^m + n - m - \frac{1}{2} - \frac{1}{2^{m+1}}.$$

On the other hand, the splitting algorithm will use only tests of type A, since at each point in the procedure there will be a type A test available which splits off a fraction of at least 2^{-m} of the remaining possibilities for θ, whereas no type B test splits off a fraction of more than $2^{-(m+1)}$ of the remaining possibilities.

Thus, the splitting algorithm constructs a testing procedure which identifies θ by first applying the various tests $T_A(d_1, \ldots, d_m)$ to determine the first m digits, then applying the tests $T_A(d_1, \ldots, d_m, d_{m+1}, \ldots, d_{2m})$ to determine the second block of m digits, etc., finally applying the tests $T_A(d_1, \ldots, d_{sm}, d_{sm+1}, \ldots, d_n)$ to complete the identification where $s = [n/m]$. The expected number of tests required by this procedure is easily calculated to be

$$K_A = s \cdot \frac{(2^m - 1)(2^{m-1} + 1)}{2^m} + \frac{(2^u - 1)(2^{u-1} + 1)}{2^u}$$

where

$$n = sm + u \quad \text{so that} \quad 0 \leqq u < m.$$

Now, let M be an arbitrary integer exceeding 32. Define integers n and m by

$$n = [\log_2 M], \qquad m = [\log_2 n] - 1$$

where $[x]$ denotes the greatest integer $\leqq x$. Thus, $n > m \geqq 1$. By the definition of R^*,

$$\begin{aligned} R^*(M) \geqq R^*(2^n) \geqq \frac{K_A}{K_B} &\geqq \frac{s(2^m - 1)(2^{m-1} + 1)}{2^m(2^m + n)} \\ &\geqq \frac{s \cdot 2^m}{2(2^m + n)} = \frac{s}{2(1 + n \cdot 2^{-m})} \\ &> \frac{s}{10} = \frac{1}{10}\left[\frac{n}{m}\right] \quad \text{since } n < 2^{m+2}, \\ &\geqq \frac{1}{10}\left[\frac{\log_2 M}{\log_2 \log_2 M}\right]. \end{aligned}$$

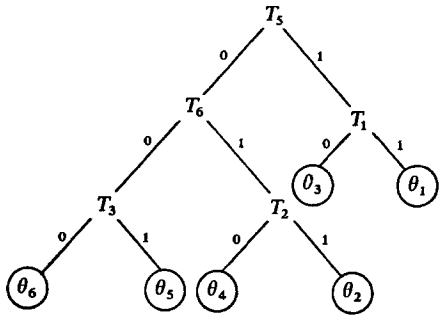
Since $R^*(M)$ is always at least 1, the theorem obviously holds for $2 < M \leqq 32$. This completes the proof. ∎

An Upper Bound. When the splitting algorithm is applied to a particular binary testing problem, a number of different testing procedures can result because of the possibility of ties (i.e., equally good tests). For example, consider the following problem:

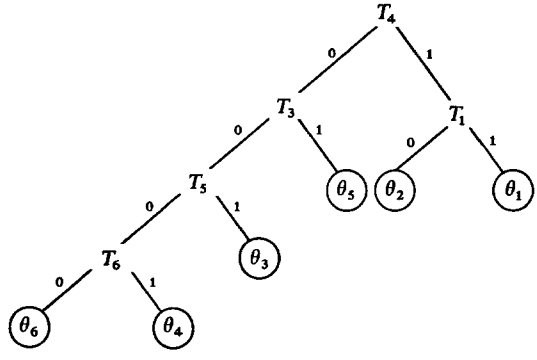
$$\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}, \quad p_i = 1/6, \quad 1 \leqq i \leqq 6,$$

$$\mathcal{T} = \{T_1, \ldots, T_6\} \quad \text{with} \quad T_1 = \{1\}, \quad T_2 = \{2\}, \quad T_3 = \{5\}, \quad T_4 = \{1, 2\},$$
$$T_5 = \{1, 3\}, \quad T_6 = \{2, 4\}.$$

One possible testing procedure which can result from the splitting algorithm is:



The expected cost of this procedure is $1/6(3+3+3+3+2+2)=8/3$ which is optimal for this problem. However, another procedure which can result from the splitting algorithm is the following:



The expected cost of this procedure is $1/6(4+4+3+2+2+2)=17/6$.

In fact, for infinitely many M it is not difficult to construct examples with M equiprobable objects for which the *best* testing procedure derived from the splitting algorithm has expected cost $K^*=K_0=\log_2 M$ (and is therefore optimal) while the *worst* testing procedure derived from the splitting algorithm has expected cost K' satisfying

$$\frac{K'}{K_0} > \frac{1}{4}\,\frac{\log_2 M}{\log_2 \log_2 M}.$$

The following result shows that this is essentially the worst behavior of the splitting algorithm in this case.

Theorem 2. Suppose at most $c \log_2 M$ tests are ever required to identify θ in the optimal testing procedure. Then

$$\frac{K'}{K_0} \leqq \frac{2c \log_2 M}{1+\log_2 c+\log_2 \log_2 M} + 2c.$$

The proof of Theorem 2 will depend on the following result.

Lemma. Suppose $\Theta=\{\theta_1, \ldots, \theta_M\}$, $p_i=1/M$ for all i, and for some p, $0<p\leqq 1/2$, $\mathcal{T}$ satisfies the following condition:

For all $S \subseteq \Theta$ with $|S| \geqq 2$, there exists $T \in \mathcal{T}$ such that

$$p|S| \leqq |S \cap T| \leqq (1-p)|S|.$$

Then

$$K' \leqq \frac{\log_2 M}{p \log_2 (1/p)} + \frac{1-p}{p},$$

Proof. The proof will proceed by induction on M. The Lemma certainly holds for $M = 1$ and 2. Assume for some $M_0 \geqq 3$ that it holds for all $M < M_0$. Consider a testing problem satisfying the hypotheses of the Lemma with $M = M_0$ and suppose we generate a testing procedure using the splitting algorithm. Thus, the first test in the procedure must split Θ into two subsets S and $\overline{S}$ of sizes $p' M_0$ and $(1-p') M_0$, respectively, where $p \leqq p' \leqq 1/2$. Let K be the expected number of tests required by S after the first test and let $\overline{K}$ be the expected number required by $\overline{S}$. Then

$$K' \leqq p' K + (1-p') \overline{K} + 1.$$

By the induction hypothesis, however, we have

$$K \leqq \frac{\log_2 (p' M_0)}{p \log_2 (1/p)} + \frac{1-p}{p}$$

and

$$\overline{K} \leqq \frac{\log_2 ((1-p') M_0)}{p \log_2 (1/p)} + \frac{1-p}{p}.$$

Therefore

$$K' \leqq \frac{\log_2 M_0}{p \log_2 (1/p)} + \frac{1}{p} - \frac{H(p')}{p \log_2 (1/p)}$$

where $H(x)$ is the familiar entropy function (see [1]) given by

$$H(x) = -x \log_2 x - (1-x) \log_2 (1-x).$$

Since $H(p) \leqq H(p')$ then

$$\begin{aligned} K' &\leqq \frac{\log_2 M_0}{p \log_2 (1/p)} + \frac{1}{p} - \frac{H(p)}{p \log_2 (1/p)} \\ &\leqq \frac{\log_2 M_0}{p \log_2 (1/p)} + \frac{1}{p} - 1. \end{aligned}$$

This proves the Lemma. ∎

Proof of Theorem 2. Consider a binary testing problem with $|\Theta| = M$ which satisfies the hypothesis of Theorem 2. We show that the hypotheses of the Lemma hold with $p = \frac{1}{2c \log_2 M}$. Let S be any subset of Θ with $|S| \geqq 2$. By considering the sequential splitting of S induced by the optimal testing procedure, we see that if $T \in \mathcal{T}$ minimizes $\left| \frac{1}{2} - \frac{|S \cap T|}{|S|} \right|$ then we must have

$$\frac{|S \cap T|}{|S|} \geqq \frac{1}{2c \log_2 M}.$$

For otherwise, no test could split off as many as $|S|/2c \log_2 M$ elements from S so that after $c \log_2 M$ tests, there would still remain an unresolved subset of size greater than $|S|/2$. Since this is at least one then this would contradict the

hypothesis. Hence, we may take $p = \frac{1}{2c \log_2 M}$ in the Lemma. By the conclusion of the Lemma, we have

$$K' \leqq \frac{2c \log_2^2 M}{1 + \log_2 c + \log_2 \log_2 M} + 2c \log_2 M - 1$$

and so, since $K_0 \geqq \log_2 M$,

$$\frac{K'}{K_0} \leqq \frac{K'}{\log_2 M} \leqq \frac{2c \log_2 M}{1 + \log_2 c + \log_2 \log_2 M} + 2c$$

and the theorem is proved. ∎

Concluding Remarks

We have seen that although the splitting algorithm performs quite well when the set of binary tests is complete, its behavior can deteriorate considerably when the test set is not complete. However, the example used in Theorem 1 for obtaining a large value of $R^*(M)$ was itself rather large. To what extent this is necessary is not yet known. In the more general case in which the p_i are allowed to be unequal, it can be shown [7] that large values of $R^*(M)$ can be obtained with relatively small values of M.

A number of interesting open questions remain. For example, how large can K'/K_0 be (as a function of M)? What are the best possible constants in Theorems 1 and 2? What is the smallest value of M for which $R^*(M)$ exceeds a fixed value t? In Theorem 2, by how much can the hypothesis be weakened? For example, is it enough to assume $K_0 \leqq c' \log_2 M$? Of course, similar questions can be asked in the more general case of unequal p_i and C_i. Some partial results in this direction can be found in [5] and [7].

The authors are indebted to R. L. Rivest for simplifying the original proof of Theorem 1 and to P. J. Burke for his many valuable comments.

References

1. Ash, R.: Information theory. New York: Interscience 1965, Sec. 2.5
2. Barnett, J.A., Gower, J. C.: Selection of tests for identifying yeasts (To appear)
3. Chang, H. Y.: An algorithm for selecting an optimum set of diagnostic tests. IEEE Trans. Electronic Computers EC-**14**, 706–711 (1965)
4. Chang, H. Y.: A distinguishability criterion for selecting efficient diagnostic tests. Proc. AFIPS 1968 SJCC **32**, Montvale (N.J.): AFIPS Press 1968, p. 529–534
5. Garey, M. R.: Optimal binary decision trees for diagnostic identification problems. University of Wisconsin, Ph.D. Thesis, June 1970
6. Garey, M. R.: Optimal binary identification procedures. SIAM J. Appl. Math. **23**, 173–186 (1972)
7. Garey, M. R., Graham, R. L.: To appear
8. Huffman, D. A.: A method for the construction of minimum redundancy codes. Proc. I.R.E. **40**, 1098–1101 (1952)
9. LaMacchia, S. E.: Diagnosis in automatic checkout. I.R.E. Trans. Military Electronics MIL-**6**, 302–309 (1962)
10. Pankhurst, R. J.: A computer program for generating diagnostic keys. Computer J. **13**, 145–151 (1970)

11. Sandelius, M.: On an optimal search procedure. Am. Math. Monthly **68**, 133–134 (1961)
12. Seshu, S.: On an improved diagnosis program. IEE Trans. Electronic Computers EC-**14**, 76–79 (1965)
13. Shannon, C. E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423, 623–656 (1948)
14. Zimmerman, S.: An optimal search procedure. Am. Math. Monthly **66**, 690–693 (1959)

Dr. Ronald L. Graham
Bell Telephone Lab., Incorporated
Murray Hill, N., J. USA

Dr. Michael R. Garey
Bell Telephone Lab., Incorporated
Murray Hill, N.J., USA