Stochastic Model for the Browning-Bledsoe Pattern Recognition Scheme*

G. P. STECK†

Summary—**A stochastic model is presented which gives the probabilities of successful recognition of the Browning-Bledsoe recognition scheme as a function of scheme parameters and pattern variability parameters. Also, procedures are given for estimating the variability parameters from data so that the model can be used to predict readability. The adequacy of the model is checked by comparing estimated readability with observed readability for two sets of data, one with high variability and one with low variability.**

The Browning-Bledsoe recognition scheme is also treated as a coding and decoding problem in which case the concepts of information theory are useful. Finally, brief mention is made of the connection between pattern recognition problems and classification problems in general, and the Browning-Bledsoe recognition scheme is compared and contrasted with other recognition schemes which make use of measurements on patterns.

I. Introduction

THE BROWNING-BLEDSOE (BB) pattern recognition scheme[1] is a procedure whereby an unknown pattern is scored against a set of learned pattern classes called an *alphabet* and identified as that member of the alphabet which scored highest. Ties are, of course, possible and although this report is concerned with their probability, it is not concerned with means of breaking them.

The success of any recognition scheme may be measured by the probability of correct recognition. This report presents a stochastic recognition model which is used to investigate the functional dependence of successful recognition on the parameters of the recognition scheme itself and on the variability parameters of the patterns presented to it.

The BB recognition scheme can also be considered as a coding and decoding problem, in which case some of the concepts of information theory are useful as well as instructive.

The goal of the work described here has been to understand the recognition scheme sufficiently well so that the parameters of the scheme can be chosen to attain any desired probability of successful recognition.

II. Description of the Browning-Bledsoe Pattern Recognition Scheme

In the BB pattern recognition scheme, a *pattern* x is defined as a rectangular matrix of N zeros and ones. The elements of x are ordered, which means that x can be treated as an N-digit binary number. For large N, say $N \geq 150$, the number of distinguishable patterns is too enormous a number to be dealt with effectively. Consequently, in the BB scheme the N elements of x are divided in some fashion, generally random, into t mutually exclusive ordered sets of n elements each. The pattern is now treated as t n-digit binary numbers requiring $t \cdot 2^n$ memory addresses where $nt = N$.

In the phraseology of Browning and Bledsoe, the ordered sets of n elements into which the pattern is divided are called *n-tuples* and the binary numbers assumed by an n-tuple are called the *states* of the n-tuple. In order to simplify future discussion, it is convenient to describe a pattern x by a column vector $\bar{x}$ of $t \cdot 2^n$ elements which has t elements equal to 1 and the rest of the elements equal to zero; that is,

$$\bar{x} = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_{t \cdot 2^n} \end{bmatrix}$$

where

$$a_{(r-1)\cdot 2^n + k + 1} = \begin{cases} 1 & \text{if the } r\text{th } n\text{-tuple is observed to be in state } k; \\ 0 & \text{otherwise.} \end{cases}$$

Thus $\bar{x}$ is a list of the occurrence or nonoccurrence of each state of each n-tuple. Those rows of $\bar{x}$ which correspond to observed states contain a 1; the rest contain zeros. This means that exactly one of the numbers $a_1, a_2, \cdots, a_{2^n}$ is a 1, exactly one of the numbers $a_{2^n+1}, a_{2^n+2}, \cdots, a_{2 \cdot 2^n}$ is a 1, etc. An example of what is meant by the foregoing is given in Fig. 1.

The pattern presented in Fig 1 is a rather sloppy "A" and if the ordering of the pattern is by rows, this "A" could be treated as the binary number 11100011011000010011O $\cdots$ 1110010001000110100000011, where the leading zeros have been suppressed. One of the many possible 5-tuples is shown in the state 11001.

An unknown pattern is recognized by comparing it with ones that have been learned. This learning process is accomplished in the following way. Suppose the alphabet consists of b pattern classes, $X_1, X_2, \cdots, X_b$, and suppose m representations of each are to be learned. For example,

* Received April 20, 1961; revised manuscript received August 8, 1961. This work was performed under the auspices of the U. S. Atomic Energy Commission. The material in this paper has been presented at a Statistics Seminar, University of California, Berkeley, and has been presented by invitation at the 1961 Annual Meeting of the Institute of Mathematical Statistics, Seattle, Wash.

† Sandia Corp., Albuquerque, N. Mex.

[1] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by Machine," *Proc. EJCC*, pp. 225–232; December, 1959.

$$\begin{array}{cccccccc}
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
\cline{2-6}
1 & \multicolumn{5}{|c|}{1 \quad 1 \quad 0 \quad 0 \quad 1} & 0 & 0 \\
\cline{2-6}
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}$$

Fig. 1—An example of an A shown on an 8×10 mosaic.

pair	state	x_{11}	x_{12}	$x_{13} \cdots$	x_{1m}	X_1
	00	1	1	0	0	1
1	01	0	0	1	1	1
	10	0	0	0	0	0
	11	0	0	0	0	0
	00	0	1	0	0	1
2	01	1	0	1	1	1
	10	0	0	0	0	0
	11	0	0	0	0	0
	00	0	0	0	0	0
3	01	1	0	1	0	1
	10	0	1	0	0	1
	11	0	0	0	1	1

Fig. 2—The result of learning the representations of X_1.

pair	state	X_1	X_2	X_3	X_4
	00	1	1	0	0
1	01	1	0	1	0
	10	0	1	0	0
	11	0	0	1	1
	00	1	1	0	0
2	01	1	1	1	0
	10	0	1	1	1
	11	0	0	0	1
	00	0	0	1	1
3	01	1	1	0	0
	10	1	0	1	0
	11	1	0	0	1

Fig. 3—The memory matrix.

the alphabet might consist of the letters $A, B, C, \cdots, Z$ and the digits $0, 1, 2, \cdots, 9$ in which case $b=36$. Let the kth representation of X_j be denoted by x_{jk}. Corresponding to the first representation of X_1, say x_{11}, are the t observed states of the different n-tuples. (Throughout this paper, pattern classes are described by upper-case letters and a representation of a pattern class is described by lower-case letters.) If the $t \cdot 2^n$ possible states are imagined to be listed vertically, then the learning of x_{11} consists of placing a 1 after each of the t observed states. In other words, this listing for x_{11} is precisely the column vector $\bar{x}_{11}$. The same is done for each of the representations $x_{12}, x_{13}, \cdots, x_{1m}$. For example, if a pattern is divided into three pairs, the list might look like the one given in Fig. 2. The column labeled X_1 in Fig. 2 is obtained by logical addition of the other columns (*i.e.*, $0+0=0$, $0+1=1+0=1+1=1$) and represents all the information available to the BB recognition scheme concerning the pattern class X_1. A similar column is obtained for each of $X_2, X_3, \cdots, X_b$. Note that the elements of the column vector labeled X_1 are random variables; that is, if different random representations of X_1 had been chosen for learning, this column vector would most likely differ from the one given. Note also that the number of 1's in this column vector measures the degree with which the representations of X_1 resemble one another; *i.e.*, it measures the "within-class variability" of the pattern class X_1. If the representations are identical there would be only 2^n ones; if the variation is extreme there could be as many as $t \cdot 2^n$ ones.

The memory matrix M, which is the heart of the recognition scheme, is obtained by setting these columns side by side, as in Fig. 3. Note that the degree to which different pattern classes differ, *i.e.*, the "among class variability," is measured by the distribution of zeros in the rows of M. If there is very little difference between pattern classes the rows of M will be mostly ones or mostly zeros. If there is a good deal of difference between pattern classes, then popular rows, *i.e.*, states which occur often for several pattern classes, should have many zeros; that is, they should not occur for representations of other pattern classes. More precisely, if one considers the probability distribution over n-tuple states which is generated by the random representations of a pattern class and then averages these state probabilities over pattern classes, one obtains a probability distribution over the rows of M. If this distribution is then used to obtain the expected number of zeros, say W, in a row of M, then W is a measure of the among class variability.

The memory matrix can now be formally defined.

$$M = \begin{pmatrix} M_1 \\ M_2 \\ \cdot \\ \cdot \\ M_t \end{pmatrix},$$

where the element in the ith row and jth column of M_r, say $a_r(i, j)$, is given by

$$a_r(i,j) = \begin{cases} 1 \text{ if at least one of the } m \text{ representations of } X_j \\ \qquad \text{puts the } r\text{th } n\text{-tuple in state } i; \\ 0 \text{ otherwise.} \end{cases}$$

Note for the example given in Fig. 3 that

$$M_1 = \begin{Bmatrix} 1100 \\ 1010 \\ 0100 \\ 0011 \end{Bmatrix}, \quad M_2 = \begin{Bmatrix} 1100 \\ 1110 \\ 0111 \\ 0001 \end{Bmatrix}, \quad M_3 = \begin{Bmatrix} 0011 \\ 1100 \\ 1010 \\ 1001 \end{Bmatrix},$$

and

$$M = \begin{Bmatrix} M_1 \\ M_2 \\ M_3 \end{Bmatrix} = \begin{Bmatrix} 1100 \\ 1010 \\ 0100 \\ 0011 \\ 1100 \\ 1110 \\ 0111 \\ 0001 \\ 0011 \\ 1100 \\ 1010 \\ 1001 \end{Bmatrix}.$$

This defines the learning process.

Now suppose an unknown pattern (a representation of X_p), x_p, is presented for recognition. To it corresponds a column vector $\bar{x}_p$. The matrix product,

$$S_p' = (S_{p1}, \cdots, S_{pb}) = \bar{x}_p' \cdot M$$

where $\bar{x}_p'$ denotes the transpose of $\bar{x}_p$, gives a column vector S_p where S_{pj} is the score of x_p against the pattern class X_j. To phrase this another way: if $\bar{x}_p$ is listed vertically alongside M and if all rows of M which correspond to zeros in $\bar{x}_p$ are deleted, then one obtains a submatrix of M. This submatrix, say U, and the rows which comprise it are said to be "selected" from M by $\bar{x}_p$. Since $\bar{x}_p$ is a random representation of X_p it follows that the elements of U are randon variables. The sums obtained by adding within the columns of U form the score matrix, S_p'.

To illustrate what is being accomplished, consider the memory matrix of Fig. 3 and take

$$\bar{x}_p' = (100010000010).$$

The submatrix of M selected by $\bar{x}_p$ consists of the first, fifth, and eleventh rows of M; consequently,

$$U = \begin{Bmatrix} 1100 \\ 1100 \\ 1010 \end{Bmatrix} \quad \text{and} \quad S_p' = (3 \;\; 2 \;\; 1 \;\; 0).$$

Let J denote the set of j's for which

$$\underset{1 \leq j \leq b}{\text{maximum}}\, S_{pj}$$

is attained. If $p \in J$, the identification of x_p is said to be "correct." In the example, the set J consists of the single element $\{1\}$. If $p=1$, *i.e.*, x_p was a representation of X_1, then the identification of x_p is correct. Further, the identification of x_p is said to be "correct without tie" or "correct with tie" according as J contains exactly one or more than one element. This emphasis on whether an identification is correct "with tie" or "without tie" is made for two reasons. First, the mechanics of the model can separate the two cases, and estimation of two probabilities provides a better check on the adequacy of the model than estimation of just one. Second, it is useful to separate the two cases if some kind of tie-breaking procedure is available. This defines the recognition process.

In summary, the learning process consists in forming a memory matrix M of zeros and ones, the rows of which correspond to a particular state of a particular n-tuple and the elements of that row specifying what types of patterns had a representation which put that particular n-tuple into that particular state. The net effect of this is that the density of ones in the columns of M measures the within class variability of the alphabet, and the expected number of zeros in the rows of M measures the among class variability of the alphabet.

The reading process consists in taking one row, as determined by the unknown pattern, from each of the t matrices M_r, adding them together to obtain scores and identifying the pattern(s) with the highest score(s).

The parameters of the BB recognition scheme are: b, which denotes the size of the alphabet considered and is determined by the job to be done; n and t, which are determined by the amount of detail needed to resolve the alphabet into its different patterns; and m, which denotes the number of experiences in the learning process.

The parameter m should be large enough so that a new representation of a pattern should have a high probability of selecting rows of M which have been selected before by that class of pattern. However, the property of patterns which makes the BB recognition scheme work is that certain states are virtually impossible for a particular type of pattern. This and only this property gives incorrect patterns low scores in the score matrix. Consequently, if m is taken too large, many "impossible" states are observed and incorrect patterns get higher scores because the number of ones in M cannot decrease as m increases. This phenomenon of disappearance of zeros in M as m increases is called *saturation.* This phenomenon of saturation is unfortunate. Usually in sampling problems the more observations one has, the more information one has about the statistical population in question. Here the opposite is true: beyond a certain point the more observations one has, the *less* information one has about the population of patterns. Fortunately there are ways out of this difficulty.

Instead of forming the memory matrix M of ones and zeros to denote the occurrence or non-occurrence of events, one can form M from the relative frequencies with which the n-tuples are in their various states. Recognition schemes based on a memory matrix formed from frequencies are also considered by Browning and Bledsoe.[1]

While the recognition scheme using frequencies will generally perform better than the scheme using zeros and ones in the sense of recognition ability, the scheme using frequencies will generally perform less well in the sense of speed and storage requirements. If $b=36$ (which corresponds to an alphabet of English capital letters and the ten digits), then a single IBM-704 computer word of 36 bits is sufficient to store one row of M. However, if relative frequencies from 0.00 to 0.99 are expected, then approximately seven computer words are required to specify a row of M. Thus, using a frequency scheme with n-tuples of length n requires roughly the same computer memory capacity as using a "zero and one" scheme with n-tuples of length $n+3$. Consequently the question of which scheme is best in some over-all sense is still open. Some remarks concerning the choice of an optimum memory matrix and the relation of the BB scheme to other pattern recognition schemes are given in Section VIII.

In Section III the notions of within pattern and among pattern variability are imposed on the recognition scheme to obtain a stochastic recognition model.

III. Stochastic Recognition Model

As has been observed in Section II, the probability of correct recognition is influenced adversely by within pattern variability and influenced favorably by among pattern variability. To fix the ideas, imagine a typical n-tuple and a new representation of X_1.

This new pattern selects t rows of M, one of which could look like the following

$$1001001100 \cdots 01101.$$

For successful recognition of X_1, a zero in the first position of the selected row is not desired; the other zeros are desired. This motivates the definition of two parameters p and q (p and q are independent parameters and do not necessarily sum to one) which denote, respectively, the probability of a zero where it is not wanted and the probability of a zero where it is wanted. Obviously it is desirable that p be small and q be large.

With m, n, t, and b fixed, p and q are functions of pattern variability, p being small if the within-class variability is small and q being large if the among-class variability is large. The random element in the recognition problem is then the random variation of the patterns about typical patterns. While no means have as yet been found to relate the parameters p and q directly to pattern variability, they are in an intuitive sense measures of this variability and are sufficient to describe the *effect* of pattern variability on recognition ability.

The stochastic recognition model is described as follows. When a random representation of X_j is presented for recognition it selects t rows of M as described above. Thus to the random representation of X_j corresponds the random matrix U with t rows and b columns. Let the elements of U be

$$\begin{array}{c} U_{11}, U_{12}, \cdots, U_{1j}, \cdots, U_{1b} \\ \\ U_{21}, U_{22}, \cdots, U_{2j}, \cdots, U_{2b} \\ \vdots \quad \vdots \qquad \vdots \qquad \vdots \\ U_{t1}, U_{t2}, \cdots, U_{tj}, \cdots, U_{tb}. \end{array}$$

First, it is assumed that these variables are completely independent. This means that the responses of different n-tuples are assumed independent, which is a reasonable assumption if no two n-tuples are looking at the same part of the pattern. Additional remarks justifying this assumption will be made below. The independence assumption also means that the elements of a row are independent, which is certainly tenable if the patterns are thought of as varying independently. Second, it is assumed that U_{1j}, $U_{2j}, \cdots, U_{tj}$ are identically distributed $B(1, 1-p)$ random variables. (The symbol $B(n, p)$ denotes a binomially distributed random variable with parameters n and p.) This assumption is unrealistic since some n-tuples will generally be considerably better for recognition purposes than others; however, this nonuniformity of n-tuple response is averaged out so that the assumption is at least reasonable. (Remember that the pattern to be recognized is a representation of X_j.)

Finally, it is assumed that the other variables, U_{mk}, $m=1, 2, \cdots, t$, $k=1, 2, \cdots, b$, $k\neq j$, are identically distributed $B(1, 1-q)$ random variables. This final assumption is the only one which could be considered unreasonable as well as unrealistic for, obviously, some patterns will be more alike than others (for example, an "O" is much more like a "Q" than it is like an "X"). This feature of patterns is not averaged out and would seem to affect the recognition model in such a way that in reality $P_1=P$(correct without tie) is smaller than predicted by the model, and in reality $P_2=P$(correct with tie) is larger than predicted by the model. In other words, the effect of "look alikes" is to increase P_2 at the expense of P_1 and probably even to reduce P_1+P_2.

Let us now return to the assumption that the responses of different n-tuples are independent. This assumption is not realistic but the fact is that the correlations involved are small as soon as the n-tuples have some separation. To illustrate this, consider a situation in which the assumption of independence is strictly true for any pair of nonoverlapping n-tuples. Consider some pattern, say an A, blocked in on a mosaic as in Fig. 6, and generate a random representation of this pattern in the following way. Let each black square in the original pattern have a probability of 0.2 of becoming white and let each of the boundary white squares in the original pattern have a probability of 0.2 of becoming black—all these choices being made independently. The result of repeating this construction is a collection of random representations of A, perhaps blotchy but recognizable, for which nonoverlapping n-tuples respond independently.

In this extreme example all the "noise" in the pattern is introduced independently at each point of the pattern. For other patterns, like the typed and handwritten characters considered in this paper, part of the noise is of this kind because of the photocell mosaic interpretation of the pattern; the rest arises because of scale changes, shifting, and other distortions. Distortions, too, generally lead to fairly localized dependences and separated mosaic points will not be highly correlated. (It is granted, however, that separated mosaic points can be highly correlated—for example, if the only distortion is one of shift.)

Furthermore the n-tuple independence assumption was checked for the handwritten data used by comparisons of the following kind. Let F and G denote two 2-tuples and let FG denote the 4-tuple obtained by appending G to the end of F. If F and G respond independently then, and only then, $P(F \text{ is in state } i)\cdot P(G \text{ is in state } j) - P(FG \text{ is in state } ij) = 0$, where if i is the state 01 and j is the state 11, then ij is the state 0111. Many such differences were checked and almost all of them found to be suitably small.

Under the above independence and distributional assumptions, the score matrix

$$S_j' = (S_{j1}, S_{j2}, \cdots, S_{jj}, \cdots, S_{jb})$$

consists of b random variables where S_{jj} is a $B(t, 1-p)$ random variable and S_{jk}, $k \neq j$, is a $B(t, 1-q)$ random variable. The unknown pattern is then identified as that member of the alphabet which scores highest, assuming no tie.

It follows then that the unknown pattern X_j is recognized "correct without tie" or "correct with tie" according as

$$S_{jj} > \max_{k \neq j} S_{jk} \quad \text{or} \quad S_{jj} = \max_{k \neq j} S_{jk}.$$

The probabilities of these events are found to be

$$\begin{aligned} P_1 &= P(\text{correct without tie}) \\ &= \sum_{h=1}^{t} P\left(S_{jj} = h \text{ and } \max_{k \neq j} S_{jk} < h\right) \\ &= \sum_{h=1}^{t} P(S_{jj} = h)\cdot[P(S_{jk} < h \mid k \neq j)]^{b-1} \\ &= \sum_{h=1}^{t} \binom{t}{h} p^{t-h}(1-p)^h \left[\sum_{s=0}^{h-1} \binom{t}{s} q^{t-s}(1-q)^s\right]^{b-1}, \end{aligned}$$

and

$$\begin{aligned} P_2 &= P(\text{correct with tie}) \\ &= \sum_{h=1}^{t} \binom{t}{h} p^{t-h}(1-p)^h \left\{\left[\sum_{s=0}^{h} \binom{t}{s} q^{t-s}(1-q)^s\right]^{b-1} \right. \\ &\quad \left. - \left[\sum_{s=0}^{h-1} \binom{t}{s} q^{t-s}(1-q)^s\right]^{b-1}\right\}. \end{aligned}$$

An IBM-704 computer program has been written which computes these probabilities for $p = 0.01(0.01)0.50$ and $q = 0.10(0.01)0.99$ as a function of t and b. (The notation $n = a(b)c$ means that n ranges from a to c in increments of b.) Typical graphs of P_1 and P_2 appear in Fig. 4.

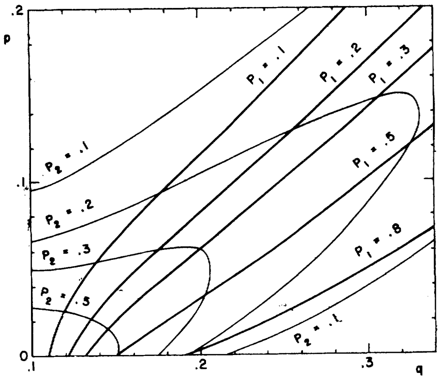

Fig. 4—Graphs of P_1 and P_2 for $b = 36$ and $t = 24$ (6-tuples on a 144 point mosaic).

If the recognition scheme embodies a tie-breaking procedure which breaks a tie correctly with probability α, then the probability of successful recognition is $P = P_1 + \alpha P_2$.

IV. Estimation of Variability Parameters

Since no effort will be made to determine p and q from the patterns themselves, it is necessary to describe them in terms of the memory matrix M. Intuitively, as m increases, both p and q should decrease as M becomes more saturated, and as n increases both p and q should increase as M is essentially desaturated. Since varying m and n in either direction produces both good and bad effects (p and q increasing or decreasing together) it is necessary that any description of p and q in terms of M should include a description of how they behave with m and n.

Consider first the variability parameter p. Let X_j represent a particular type of pattern, let π_{ijr}, $(\Sigma\pi_{ijr} = 1)$, denote the probability that a representation of X_j puts the rth n-tuple in the ith state, and let Z_{jr} denote the number of zeros in the jth column of M_r after m alphabets have been learned. Z_{jr} is then the number of states of the rth n-tuple not selected by the m random representations of X_j, and can be expressed as

$$Z_{jr} = \sum_{i=1}^{2^n} Y_{ij},$$

where Y_{ij} is 0 or 1 according as at least one of the m representations of X_j selects the ith state of the rth n-tuple or not. Also,

$$P(Y_{ij} = 1) = (1 - \pi_{ijr})^m.$$

Therefore,

$$\begin{aligned} E[Z_j] &= \sum_i EY_{ij} \\ &= \sum_i P(Y_{ij} = 1) \\ &= \sum_i (1 - \pi_{ijr})^m \\ &= G_{jr}(m), \text{ say.} \end{aligned}$$

The probability that another, independent, representation of X_j selects a state of the rth n-tuple not previously selected is

$$p_{jr}(m) = \sum_i \pi_{ijr}(1 - \pi_{ijr})^m$$
$$= G_{jr}(m) - G_{jr}(m + 1).$$

Note that $p_{jr}(m)$ is the variability parameter p specialized to a particular type of pattern and a particular n-tuple. This fact together with the form of the above equations suggests defining a function $G(m)$, where $bt\,G(m)$ denotes the expected number of zeros in M. Then

$$G(m) = \frac{1}{bt}\sum_j \sum_r G_{jr}(m) = \frac{1}{bt}\sum_j \sum_r \sum_i (1 - \pi_{ijr})^m.$$

Suppose G can be written in the following form,

$$G(m) = \sum_i (1 - \rho_i)^m, \quad \text{where} \quad \sum_i \rho_i = 1,$$

where the ρ_i are parameters depending on the variability of the patterns and the choice of n-tuples; that is, ρ_i represents some kind of average of the π_{jr}, $j=1, 2, \cdots, b$, $r=1, 2, \cdots, t$. $G(m)$ therefore represents an average expected number of zeros in a column of a typical M_r, the average being taken over all types of patterns and n-tuples. In this formulation it makes sense to define the variability parameter p by

$$p_n(m) = \sum \rho_i(1 - \rho_i)^m$$
$$= G_n(m) - G_n(m + 1).$$

In this last expression the dependence of p and G on m and n is indicated. The dependence on the particular n-tuples used is assumed though not indicated.

Now let $Z(m)$ denote the observed number of zeros in M after m representations of the alphabet have been learned. Then

$$\hat{G}(m) = \frac{1}{bt} Z(m)$$

is an empirically obtained graph of the function $G(m)$, and its first difference, $\hat{\Delta}(m) = \hat{G}(m) - \hat{G}(m+1)$, is an estimate of $p(m)$, though a poor one since it ignores neighboring p's. However, without a model for $p(m)$ as a function of m, no standard "best" estimates exist and some arbitrariness is inevitable. In order to effect some averaging on m, the estimates $\hat{p}(m)$ given in Table I were obtained from the parabola fitted by least squares to the 21 numbers, $\hat{a}(m \pm k)$, $k=0, 1, 2, \cdots, 10$, where

$$\hat{a}(k) = (1/3)[\hat{\Delta}(k - 1) + \hat{\Delta}(k) + \hat{\Delta}(k + 1)].$$

Now consider the variability parameter q. Again let π_{ijr} denote the probability that a representation of X_j puts the rth n-tuple in state i. The probability that a random pattern selects the ith state of the rth n-tuple is then

$$\frac{1}{b}\sum_j \pi_{ijr} = \beta_{ir}, \text{ say.}$$

If W_{ir} denotes the expected number of zeros in the ith row of M_r after m alphabets have been learned, then the expected number of zeros in the row of M_r selected by a random pattern is

$$W_r = \sum_i \beta_{ir} W_{ir}.$$

Let

$$W = \frac{1}{t}\sum_r W_r;$$

then W can be thought of as the expected number of zeros in a typical row of M selected by the unknown pattern. With probability p, one of these zeros will be in the position corresponding to what the unknown pattern really is; consequently the probability of a zero in some position other than that, which is q, is defined as

$$q = p\left(\frac{W - 1}{b - 1}\right) + (1 - p)\frac{W}{b - 1} = \frac{W - p}{b - 1}\cdot$$

The variability parameter q is then estimated from the equation

$$\hat{W} = \frac{1}{t}\sum_r \hat{W}_r = \frac{1}{tb}\sum_i \sum_j \sum_r \hat{\pi}_{ijr}\hat{W}_{ir},$$

where $\hat{\pi}_{ijr}$ and $\hat{W}_{ir}$ are the observed values of π_{ijr} and W_{ir}, respectively, by using

$$\hat{q} = \frac{\hat{W} - \hat{p}}{b - 1}\cdot$$

A very similar expression for q arises from considering the matrix M as a transmission channel. This idea is developed in Section V.

V. Information Theoretic Aspects of the BB Recognition Scheme

In information theory a *channel* consists of a pair of abstract spaces X and Y and a probability distribution over Y for each x. The elements of X are the inputs to the channel and the elements of Y are the outputs of the channel. In the context of the BB pattern recognition scheme X is the space of patterns and Y is the space of states of an n-tuple.

As before, let W_{ir} denote the expected number of zeros in the ith row of M_r. If an unknown pattern selects row i then the conditional distribution over the space of patterns, X, is uniform over the patterns represented by 1's in the ith row of M_r. Consequently the amount of information required to specify the unknown pattern when the ith row of M_r is received at the output of the channel (*i.e.*, selected) is given by

$$-\sum_{j=1}^{b-W_{ir}}\left(\frac{1}{b - W_{ir}}\right)\log\frac{1}{b - W_{ir}} = \log(b - W_{ir}).$$

(All logarithms are to the base 2.)[2]

[2] See, for example, Amiel Feinstein, "Foundations of Information Theory," McGraw-Hill Book Co., Inc., New York, N. Y., ch. 3; 1958.

The *equivocation* of the channel is obtained by averaging the above expression over i. From the previous section the probability of an unknown pattern selecting the ith row of M_r was β_{ir}, ($\sum_i \beta_{ir} = 1$); therefore the equivocation associated with the rth n-tuple is

$$E_r = \sum_i \beta_{ir}(b - W_{ir}).$$

If the types of patterns are assumed to be equally likely, *i.e.*, the probability distribution over X is uniform, then the amount of information required to specify a pattern before transmission is log b, and the transmission rate, R_r, is defined by

$$\begin{aligned} R_r &= \log b - E_r \\ &= \sum_i \beta_{ir}[\log b - \log (b - W_{ir})], \left(\text{since } \sum_i \beta_{ir} = 1\right), \\ &= \log \prod_i \left\{ \frac{1}{1 - \dfrac{W_{ir}}{b}} \right\}^{\beta_{ir}}. \end{aligned}$$

That is, the transmission rate is the amount of information needed before transmission less what is needed afterwards.

The relationship that exists between R and q is developed as follows. If $R_r = 1$, that is, the rth n-tuple transmits one bit, then all that is known about the unknown pattern is which half of the alphabet it is in. In other words, on the average the row of M_r selected by the pattern will be half ones and half zeros. Similarly, if $R_r = k$, then the row of M_r will have on the average $b(1 - 1/2^k)$ zeros. The probability is p that a zero occurs where it is not wanted, and if it is assumed that the other zeros are uniformly distributed among the other $b - 1$ positions, then the variability parameter could be defined by (call it q^* to distinguish it from the q previously defined)

$$\begin{aligned} q_r^* &\cong p\left(\frac{b(1 - 2^{-R_r}) - 1}{b - 1}\right) + (1 - p)\left(\frac{b(1 - 2^{-R_r})}{b - 1}\right) \\ &= 1 - \frac{b2^{-R_r} - (1 - p)}{b - 1}. \end{aligned}$$

But

$$\begin{aligned} 2^{-R_r} &= \prod_i \left(1 - \frac{W_{ir}}{b}\right)^{\beta_{ir}} \\ &\cong \prod_i \left(1 - \frac{\beta_{ir} W_{ir}}{b}\right) \\ &\cong 1 - \frac{1}{b} \sum_i \beta_{ir} W_{ir}. \end{aligned}$$

Therefore,

$$\begin{aligned} q_r^* &\cong \frac{\sum_i \beta_{ir} W_{ir} - p}{b - 1} \\ &= \frac{W_r - p}{b - 1}. \end{aligned}$$

For a particular n-tuple, this was the expression derived earlier, and $q^* = (1/t)\Sigma q_r^* = q$.

VI. Models for the Behavior of $p_n(m)$

In Section IV, $p_n(m)$ was defined by $p_n(m) = G_n(m) - G_n(m+1)$, where $G_n(m)$ denoted the average expected number of zeros in a column of a typical M_r, the average being taken over pattern classes and n-tuples. The function $G_n(m)$ was assumed to be of the form

$$G_n(m) = \sum_{i=1}^{2^n} (1 - \rho_i)^m, \quad \text{where} \quad \sum \rho_i = 1,$$

and ρ_i is the average probability of a random pattern putting a typical n-tuple into the ith state.

Two one-parameter classes of functions to approximate $p_n(m)$ were examined to see whether they could provide a rationale for estimation of $p_n(m)$. Both of these classes of functions fit the data reasonably well.

First, following Kamentsky[3] who dealt with a similar function, the ρ_i (in some order) were assumed to be a geometric series; that is, $\rho_i = \alpha \exp\left[-\alpha(i - \frac{1}{2})\right]$. Then approximating the sum by an integral yields the following functional form for $p_n(m)$,

$$\begin{aligned} p_1(m \mid n, \alpha) &= \sum \rho_i(1 - \rho_i)^m \cong \alpha \int_0^{2^n} (1 - \alpha e^{-\alpha x})^m e^{-\alpha x} dx \\ &\cong \left[1 - (1 - \alpha)^{m+1}\right]/\alpha(m + 1). \end{aligned}$$

Second, the assumption that different 1-tuples respond independently with the same probability of being in their least frequent state implies that exactly $\binom{n}{k}$ of the $\rho_i = \alpha^k(1 - \alpha)^{n-k}$, $k = 0, 1, \cdots, n$, where α denotes the average probability that a 1-tuple is in its least frequent state. Consequently,

$$p_2(m \mid n, \alpha) = \sum_{k=0}^{n} \binom{n}{k} \alpha^k (1 - \alpha)^{n-k} \left[1 - \alpha^k(1 - \alpha)^{n-k}\right]^m.$$

The second function provides a slightly better fit to the data than the first; however, neither fits well enough for the purpose of estimating p. There are two reasons for this. First, both p_1 and p_2 underestimate p for small m and then overestimate p when m is larger. Second, P_1 is very sensitive to variation of p—changing p by 0.02 can change P_1 by as much as 0.1 in the range of p's encountered in Table

[3] L. A. Kamentsky, "Simulation of three machines which read rows of handwritten arabic numbers," IRE Trans. on Electronic Computers, vol. EC-10, pp. 489–501; September, 1961.

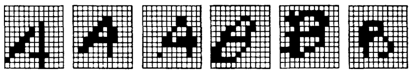

Fig. 5—Examples of *A* and *B* from the BTLHW data.

TABLE I

Predicted and Observed Recognition Rates

Data	Parameters			Estimates		Predicted		Observed	
	n	m	CR	$\hat{q}$	$\hat{p}$	$\hat{p}_1$	$\hat{p}_2$	P_1	P_2
BTLHW uncentered	6	25	900	0.236	0.119	0.18	0.20	0.13	0.23
	6	40	360	0.189	0.089	0.14	0.23	0.10	0.29
	8	25	900	0.371	0.226	0.16	0.17	0.18	0.19
BTLHW centered	6	25	900	0.228	0.079	0.30	0.26	0.18	0.30
	8	25	900	0.352	0.155	0.31	0.23	0.23	0.27
TYPE	3	10	477	0.625	0.041	1.00	0	0.84	0.02
	4	10	477	0.590	0.055	1.00	0	0.84	0.02

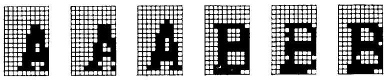

Fig. 6—Examples of *A* and *B* from typewritten data.

I (see Fig. 4)—and this means that even small systematic errors in estimating p must be avoided.

If these systematic errors are to be eliminated, it will be necessary to postulate ρ_i which decrease more slowly than a geometric series. This is currently being explored.

VII. Experimental Results

The stochastic recognition model described above has been tried on two sets of data in several experimental situations. One set of data was obtained from the Bell Telephone Laboratories and will be referred to as the "BTL Handwritten" data (BTLHW).[4] These data consist of 50 handwritten representations of an alphabet containing the capital English letters and the digits 0, 1, $\cdots$, 9 (obtained from 50 different people) which have been digitized by a 12×12 photocell mosaic. Some samples of "A" and "B" are shown in Fig. 5 to indicate the extreme within-character variation. The BTLHW data were used in two ways: 1) "as is"; and 2) with preprocessing to center the center of gravity of the pattern. The experimental results are shown in Table I.

The second set of data (TYPE) was obtained from a typewriter by photographing (the camera appears to be off-center) the typed characters and digitizing with a flying spot scanner to a 10×15 mosaic. (It was necessary to drop 6 points from the 150-bit mosaic to convert patterns to 144 bits so that programs written for the BTLHW data could also be used for these data. The points dropped were the four corners and the points adjacent to the corners in the bottom row.) The TYPE data consist of about 20 representations of an alphabet containing the capital English letters, the digits 2 thru 9, the solidus (/), and the pound sign (#). The digits 0 and 1 were arbitrarily eliminated as being too similar to the letters *O* and *I*. Examples of *A* and *B* from this set of data are shown in Fig. 6.

The degree of fit between model and experiment is shown in Table I. As before, n denotes the size of the n-tuple and m denotes the number of alphabets learned. In each case $b=36$ and $t=144/n$. The quantity CR denotes the number of characters read. (The characters read were different from those learned.) The quantities $\hat{q}$, $\hat{p}$, $\hat{P}_1$, and $\hat{P}_2$ are estimates of q, p, $P_1=P$(correct without tie), and $P_2=P$(correct with tie), respectively. The estimate $\hat{q}^*$ was 10 percent to 40 percent larger than $\hat{q}$ and was not used. Finally, P_1 and P_2 are the observed values of P_1 and P_2.

The results in Table I clearly show the effect of "look-alikes" in producing the inequalities $\hat{P}_1 > P_1$ and $\hat{P}_2 < P_2$. In spite of this deficiency the model furnishes useful qualitative and quantitative information regarding the recognition ability of the BB scheme.

VIII. Final Remarks

The BB recognition scheme consists basically of two aspects. One of these is novel and the other is essentially classical. The classical aspect is contained in the score equation

$$S' = \bar{x}' \cdot M$$

[4] For comments concerning this BTL data and the BB recognition scheme, see the following correspondence:

W. H. Highleyman and L. A. Kamentsky, "Comments on a character recognition method of Bledsoe and Browning," IRE Trans. on Electronic Computers, vol. EC-9, p. 263; June, 1960.

W. W. Bledsoe, "Further results on the *N*-tuple pattern recognition method," L. Uhr, "A possibly misleading conclusion as to the inferiority of one method for pattern recognition to a second method to which it is guaranteed to be superior," and W. H. Highleyman, "Further comments on the *N*-tuple pattern recognition method," IRE Trans. on Electronic Computers, vol. EC-10, pp. 96–97; March, 1961.

where $\bar{x}$ is a vector of measurements on the pattern to be recognized and the matrix M is determined from making the same measurements on known representations of the patterns in the alphabet considered. The above equation firmly embeds the BB recognition scheme in the body of statistical literature associated with classification problems in general.[5] The novel aspect of the BB scheme consists in what it measures. In many other recognition schemes based on measurements, well defined and specific measurements are made—be they connectivity, number of branch points, number of intersections with horizontal and vertical grids, curvature, or whatever. In the BB scheme measurement is made of random properties of the pattern, and the numerical values of the measurements are 1 or 0; 1 if the pattern possesses the property and 0 if it doesn't. If special attention is paid to properties represented by n-typles with high transmission rate, it is possible that measurements will be made which are more useful in classifying patterns than topologically oriented ones.

Toward the end of Section II the suggestion was made that a memory matrix M should be sought which would be in some sense optimum. The statistical theory of classification provides a rationale for determining such an M. Work in this area will be reported at another time.

[5] See, for example, C. R. Rao, "Advanced Statistical Methods in Biometric Research," John Wiley and Sons, Inc., New York, N. Y.; 1952.

Correspondence

On the Number of Types of Self-Dual Logical Functions*

When a switching network for a certain logical function f is known, functions, obtained from f by variable transformations (permuting and/or complementing one or more variables), can easily be realized in the same network by relabeling and/or changing the polarity of input leads. Two logical functions are defined to belong to the same *type*, when one of the two can be transformed into another by a variable transformation. It is, then, an interesting problem to enumerate the number of the types in a given set of logical functions.

The number of symmetry types of *Boolean* functions of n variables was obtained by D. Slepian[1] in 1953. Recently B. Elspas[2] has enumerated the number of self-complementary symmetry types of *Boolean* functions.

The logical function, expressed by a network consisting purely of self-dual logical elements, such as parametrons and magnetic cores without bias input, is also self-dual. Conversely any self-dual function can be realized in a network consisting of self-dual majority decision elements.[3–5]

In this letter, the number of self-dual logical functions and the number of their symmetry types are enumerated with a modified Slepian's method.

The *dual function* $\bar{f}\,(x_1, \cdots, x_n)$ of a logical function $f\,(x_1, \cdots, x_n)$ is a function defined by the formula obtained from the definition of $f(x_1, \cdots, x_n)$ through exchanging the operations of logical product $\cdot$ and logical sum $+$, namely,

$$\bar{f}(x_1, \cdots, x_n; +, \cdot) = f(x_1, \cdots, x_n; \cdot, t). \quad (1)$$

By DeMorgan's theorem, (1) can be rewritten as

$$\bar{f}(x_1, \cdots, x_n) = f'(x_1', \cdots, x_n'), \quad (2)$$

where y' denotes the complement of y.

Let an assignment of the states of input variables $x_1=\xi_1$, $x_2=\xi_2, \cdots, x_n=\xi_n$ be denoted by $\Xi=(\xi_1, \xi_2, \cdots, \xi_n)$ and be called an *input vector*. The *complementary vector* Ξ' of an input vector Ξ is defined as a vector which has the complemented components of Ξ. For example, the complementary vector of the vector (0, 0 1) is (1, 1, 0).

Eq. (2) means that the value of a function f for an input vector Ξ and the value of the dual function $\bar{f}$ for the complementary vector Ξ' are *complementary* to each other, where *complementary* means that if one of the two takes the value of 1 (or 0), the other has the value of 0 (or 1).

A *self-dual function* f is a function for which

$$f = \bar{f}. \quad (3)$$

For a self-dual function, the values of f for mutually complementary input vectors are complementary to each other, as indicated by (2) and (3).

The value of f for Ξ', therefore, is automatically determined by the assignment of the value for Ξ. Hence, there exist $2^{2^{n-1}}$ of self-dual logical functions of n variables.

Herewith, it is to be noted that the self-duality of a logical function is a type property, namely, that either all or none of a symmetry type is self-dual. Thus the self-dual functions of n variables can be classified with the aid of variable transformations.

The set of possible variable transformations will hereafter be denoted by O_n, which forms a finite group. The number P_n of symmetry types of the self-dual functions of n variables can be enumerated by applying Pólya and Slepian's formula;[1]

$$P_n = \frac{1}{2^n n!} \sum_C n_C x_C, \quad (4)$$

were C stands for a conjugate class in the group O_n, n_C is the number of elements belonging to the class C, x_C is the number of the self-dual functions which are invariant under the

* Received June 20, 1961; revised manuscript received, September 11, 1961. This paper is based on an article which was published in Japan by the same author—"On the number of types of self-dual logical functions," *J. Information Processing Soc. Japan*, vol. 2, pp. 17–21; February, 1961. (In Japanese.)

[1] D. Slepian, "On the number of symmetry types of Boolean functions of n variables," *Can. J. Math.*, vol. 5, no. 2, pp. 185–193; 1953.

[2] B. Elspas, "Self-complementary symmetry types of Boolean functions," IRE Trans. on Electronic Computers, vol. EC-9, pp. 264–266; June, 1960.

[3] H. Takahashi, "Computing Machines," Iwanami Book Co., Tokyo; 1958. (In Japanese).

[4] Z. Kiyasu, "Mathematics for Digital Circuits," Kyoritsushuppan Book Co., Tokyo; 1960. (In Japanese).

[5] S. Muroga, I. Toda, and S. Takasu, "Theory of majority decision elements," *J. Franklin Inst.*, vol. 271, pp. 376–418; May, 1961.