fore been carried out with signals simulating the worst-case SNR conditions[1] in large-capacity matrices, in order to assess the limits in the capability of the circuit. It has been found that, with fixed-level clipping and with the buffer-register loading as in normal operation, the circuit should be capable of processing reliably (with considerable margin) a matrix with 16,384 or more cores.[9] With noise-matched clipping, it should be able to sense a matrix with 32,768 or more cores.[10] (Intermediate matrix sizes were not tested because of their obvious lack of practical significance.) With a simulated matrix size of 65,536 cores, the circuit fails to read reliably, and it has been found that here the worst-case SNR has practically reached the theoretical absolute lower-limit value of 1:1. Measurements have indicated that the circuit requires a minimum usable signal area of about 5 mv·μsec for the reliable setting of the buffer register,[11] and that the worst-case SNR's in the tested cases of simulation for 16,384 and 32,768 cores are about 2.2:1 and 1.8:1 respectively.

[9] This is simulated with 1 fully-excited and 254 (in two groups of 127) partially-excited cores. The most critical discrimination is between the signal of a "1" superposed by opposing noises from 127 partially-excited cores containing "1" and supporting noises from 127 others containing "0", and that of an "0" superposed by supporting noises from 127 cores containing "1" and opposing noises from 127 others containing "0".

[10] This is simulated with 1 fully-excited and 382 (in two groups of 191) partially-excited cores, in a similar manner as above.

[11] This requirement could probably be still less if the loading should be reduced. The buffer-register flip flop connected to the circuit is a simple two-transistor arrangement, with the set input being directly at one transistor base without a buffering emitter-follower stage.

The performance of the circuit as described above is limited primarily by the frequency characteristics of the transistors used. Employing transistors with higher cutoff frequencies and adequate current-amplification factors, the circuit should be able to exhibit the same advantages at higher repetition rates and shorter cycle times.

Conclusion

The paper has discussed in detail the practicability of improving the performance of the sense-amplifier circuit for conventional ferrite-core memories through the principles of pre-amplification strobing and noise-matched clipping. A circuit incorporating these principles and achieving notable reliability and economy has been described. It has shown that the circuit is suitable for working with short cycle times and low SNR values, and can be used to process matrices of much larger sizes than the currently-accepted apparent upper limit of about 4096 cores. It is believed that the application of these principles, as illustrated by the sense-amplifier circuit described here, can also be extended with advantage to other types of circuits which have the similar task of retrieving information from signals containing nonsporadic disturbances with critical SNR values.

Acknowledgment

The author wishes to express his thanks to Prof. H. Piloty for his kind encouragement and support of this work and for his permission to publish this paper.

A Recognition Method Using Neighbor Dependence*

C. K. CHOW†, MEMBER, IRE

Summary—**Within the framework of an early paper[1] which considers character recognition as a statistical decision problem, the detailed structure of a recognition system can be systematically derived from the functional form of probability distributions. A binary matrix representation of signal is used in this paper. A nearest-neighbor dependence method is obtained by going beyond the usual assumption of statistical independence. The recognition network consists of three levels—a layer of AND gates, a set of linear summing networks in parallel, and a maximum selection circuit. Formulas for weights or recognition parameters are also derived, as logarithms of ratios of conditional probabilities. These formulas lead to a straightforward procedure of estimating weights from sample characters, which are then used in subsequent recognition.**

Simulation of the recognition method is performed on a digital computer. The program consists of two main operations—estimation of parameters from sample characters, and recognition using these estimated values. The experimental results indicate that the effect of neighbor dependence upon recognition performance is significant. On the basis of a rather small sample of 50 sets of hand-printed alphanumeric characters, the recognition performance of the nearest-neighbor method compares favorably with other recognition schemes.

* Received April 21, 1962.

† Burroughs Corporation, Paoli, Pa.

[1] C. K. Chow, "An optimum character recognition system using decision functions," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 247–254; December, 1957.

Introduction

IN THE DESIGN of recognition systems, there are two principal areas of investigation: the extraction of characteristic features from patterns and the synthesis of recognition networks. For any given recognition task, system performance depends upon effective solution of both areas.

The first problem, that of deriving efficient sets of features, not only has not been solved, but has not as yet been properly formulated with sufficient clarity and completeness. A good general discussion of this subject has been given, among other topics of artificial intelligence, by Minsky.[2] (A large number of pattern recognition references is included in a bibliography also compiled by Minsky.[3]) In many studies on character recognition, the design of features is based primarily upon the engineer's ingenuity and intuition. Recently, Lewis[4] considered the problem of selecting features from a set of features supplied by the designer; one major restriction on the application of his work is that "the selection and the decision process both assume the characteristics to be statistically independent."

The present paper considers the problem of synthesizing recognition networks; the principal concern is the derivation of the network structure by going beyond the usual assumption of statistical independence among the characteristic features. Not only is the problem itself of interest, but, in addition, the results are useful in the selection of features, in that the virtue of features for reliable recognition must be ultimately evaluated in conjunction with some recognition network.

Since the underlying principle and mathematical derivation do not intrinsically depend upon the nature of the given features, a most primitive representation of pattern is used in this paper. It is also believed that the use of primitive features provides a more stringent test of the recognition method. A pattern is represented here by a two-dimensional array of elements, each element denoting the presence or absence of an ink mark at a particular location. The adoption of binary features is essential here to achieve relatively simple networks.

The recognition problem is considered as a statistical decision problem. The functional structure of optimum systems has been previously derived,[1] and the detailed structure of the recognition network depends upon the *a priori* distribution of characters and conditional probability distributions of patterns. (A character is considered here as a class of patterns such that all patterns in that class are identified as that character.) For example, if the noise is additive and Gaussian, then the correlation system with proper bias[5] has the maximum rate of correct recognition. This paper derives from the functional form of probability distributions a recognition method which utilizes the nonlinear relations among signals. Specifically, the detailed structure of a recognition network using neighbor dependence is obtained. Some experimental results are reported.

Assumptions and Notations

The recognition problem is considered here to be a problem of testing multiple hypotheses in statistical inference. Common to all decision problems, the essential elements are: 1) *a priori* information, 2) a decision space or set of admissible decisions, 3) observation, or signal derived from the input pattern, 4) a decision rule, and 5) a measure of performance, or criterion of optimality.

The basic problems in recognition are proper choices of a signal space and its coordinate system (namely, characteristic measurement of characters), and of a decision rule. Generally speaking, a decision rule is a map from the signal space to the decision space; the decision rule associates a unique decision with each signal. Equivalently, the rule partitions the signal space into disjoint regions, and recognition is achieved by ascertaining in which region the signal representing the unknown pattern lies. The structure of recognition networks is of principal concern in this paper.

For convenience, the problem of rejection is not elaborated upon here; the only admissible decisions are those identifying an unknown character as one of the given alphabet. Signal preprocessings, which are uniform with respect to all characters, or are independent of the class to which the pattern belongs, are not considered here. In effect, the assumption is made that such preprocessings as size normalization and registration have already been performed.

Consider an alphabet of c characters, $a_1, a_2, \cdots, a_c$. A character is represented in this paper by a two-dimensional array of elements, with each of which a binary random variable v_{ij} is associated. Arbitrarily, let ONE and ZERO denote the presence and absence, respectively, of an ink mark at a particular location. The signal space, therefore, consists of all vertices of an n-dimensional cube, each pattern being represented by a vertex of the cube. Each character a_i $(i=1, 2, \cdots, c)$ is a subset of the vertices, or a class of patterns.

Let $r \times s$ be the size of the array. The signal corresponding to a spatial pattern is represented by a binary matrix v; $v=[v_{ij}]$, $1 \leq i \leq r$ and $1 \leq j \leq s$. The joint probability distribution of v_{ij}'s depends upon which character the pattern is derived from. Let $P(v|a_i)$ denote the (discrete) conditional probability of pattern v,

[2] M. Minsky, "Steps toward artificial intelligence," PROC. IRE, vol. 49, pp. 8–30; January, 1961.

[3] M. Minsky, "A selected descriptor-indexed bibliography to the literature on artificial intelligence," IRE TRANS. ON HUMAN FACTORS IN ELECTRONICS, vol. HFE-2, pp. 39–55; March, 1961.

[4] P. M. Lewis, "The characteristic selection problem in recognition systems," IRE TRANS. ON INFORMATION THEORY, vol. IT-8, pp. 171–178; February, 1962.

[5] C. K. Chow, "Comments on optimum character recognition systems," IRE TRANS. ON ELECTRONIC COMPUTERS (*Correspondence*), vol. EC-8, p. 230; June, 1959.

given that the character is a_i. Let $p=(p_1, p_2, \cdots, p_c)$ be the distribution of characters; p_i is the *a priori* probability that character a_i occurs. Evidently, $p_1+p_2 \cdots +p_c=1$, and $p_i>0$. Information concerning the distribution p and the conditional probabilities $P(v|a_i)$'s constitutes the *a priori* information of the recognition system. The designer's knowledge of these must be built into the optimum system. The criterion of minimum error rate is used. The functional diagram of the optimum system, which has been derived previously, is depicted in Fig. 1. The system first computes, based upon *a priori* information, the set of conditional probabilities $P(v|a_i)$'s for the input signal v, weights the results by the corresponding *a priori* probabilities, p_i's, selects the largest one of the $p_iP(v|a_i)$'s, and, finally, identifies the pattern as the character a_k, if $p_kP(v|a_k)$ is the largest.

The detailed structure of the network depends upon the functional form of the conditional distributions, $P(v|a_i)$'s. For example, if the v_{ij}'s (the elements of matrix v), given the character, are statistically independent, then $P(v|a_k)$ is simply the product of $P(v_{ij}|a_k)$, and, consequently, after a logarithmic transformation, the recognition network consists of a set of linear summing networks.

In general, the point signals, v_{ij}'s, are not independent, but depend upon each other as well as upon their locations in the matrix and the character class. The resultant structure, therefore, is more complicated. To illustrate how recognition networks and formulas for weights may be derived from the functional form of probability distributions, it is assumed that each point signal may depend upon its nearest neighboring points as well as upon the character class and the location of the point within that character. Fig. 2 illustrates graphically the location (i, j) and its four nearest neighbors.

To be more precise, it is assumed that the conditional distribution is of the following form:

$$P(v|a_k) = \prod_{\substack{1\le i\le r \\ 1\le j\le s}} P(v_{ij}|v_{i,j-1}; v_{i-1,j}; a_k), \qquad (1)$$

with the definition that

$$v_{0j} = v_{i0} = 0, \qquad \text{for all } i \text{ and } j,$$

and

$$P(v_{i,j}|v_{i,j-1}; v_{i-1,j}; a_k) = \begin{cases} P(v_{11}|a_k) & \text{if } i=j=1 \\ P(v_{1j}|v_{1,j-1}; a_k) & \text{if } i=1 \text{ and } j>1 \\ P(v_{i1}|v_{i-1,1}; a_k) & \text{if } i>1 \text{ and } j=1 \end{cases}. \qquad (2)$$

The general term in (1) includes only the north and west neighbors (above and to the left); the other two neighbors are not explicitly needed. The dependence propagates through the neighbors in this fashion. Eq. (2) is simply a convenient notation to describe the

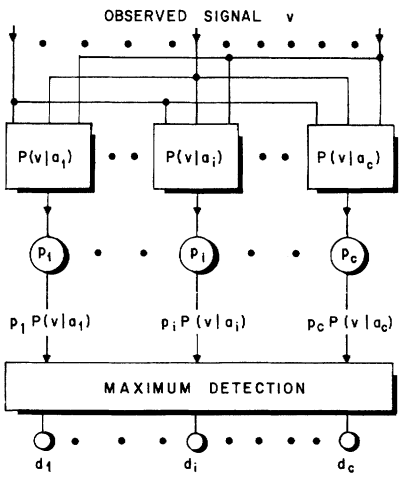


Fig. 1—Minimum error-rate system.

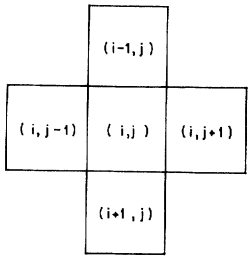


Fig. 2—Nearest neighbors.

(north and west) boundary points of the character.

The assumption of nearest-neighbor dependence is prompted by an intuition that this type of dependence is dominant in characters. Of course, a larger neighborhood could be used, thus extending the range of dependence. The derivation of recognition networks for a larger range of neighbor dependence is the same as that for the nearest-neighbor model. Both networks have the same structure, which consists of a layer of AND gates, a set of weighting and summing networks, and a maximum selection network. The size of the networks increases exponentially with the size of the neighborhood (or the range of dependence); however, for a given range of dependence, the size of the networks increases linearly with the size of the signal matrix. In practice, the exact probability distributions are generally unknown; one of the designer's tasks is to collect statistical information on these distributions. The use of a neighbor-dependence model should be considered as an approximation to the unknown distribution. By varying the range of dependence, a sequence of successive approximating structures can be obtained. Using the available statistical data, the designer can then select a particular structure from this sequence to achieve a reasonable compromise between the size or cost of the system and the recognition performance required.

Derivation of Nearest-Neighbor Model

For a given character (k) and location (i, j), the general term $P(v_{ij}|v_{i-1,j}; v_{i,j-1}; a_k)$ assumes one of

eight possible values, depending upon the states of $v_{i,j}$ and the neighbors $v_{i-1,j}$ and $v_{i,j-1}$. These eight values, summarized in Table I, are denoted as $\beta_m(i, j, k)$ and $\gamma_m(i, j, k)$, with $m = 0, 1, 2$, or 3. Subscript m is used to designate the state of the two neighbors, and is arbitrarily chosen as the decimal equivalent of the binary code formed by the states of the two neighbors. The values $\beta_m(i, j, k)$ and $\gamma_m(i, j, k)$ are, respectively, the conditional probabilities that v_{ij} is ZERO (or white) and ONE (or black), given that the character is a_k, and given that the states of two neighbors are m.

TABLE I
DEFINITION OF PARAMETERS

Point v_{ij}	Nearest Neighbor		m	$P(v_{ij} \mid v_{i-1}, v_{i,j-1}, a_k)$
	$v_{i,j-1}$	$v_{i-1,j}$		
0	0	0	0	$\beta_0(i, j, k)$
0	0	1	1	$\beta_1(i, j, k)$
0	1	0	2	$\beta_2(i, j, k)$
0	1	1	3	$\beta_3(i, j, k)$
1	0	0	0	$\gamma_0(i, j, k)$
1	0	1	1	$\gamma_1(i, j, k)$
1	1	0	2	$\gamma_2(i, j, k)$
1	1	1	3	$\gamma_3(i, j, k)$

Parameters β's and γ's are probabilities, and therefore non-negative, and are related as

$$\beta_m(i, j, k) + \gamma_m(i, j, k) = 1, \tag{3}$$

for all m, i, j, and k. In general, the values of β_m and γ_m vary from character to character and from point to point.

Since all v_{ij}'s are either 0 or 1, the probability function may be expressed as a product of four factors:

$$\left.\begin{aligned} &P(v_{ij} \mid v_{i,j-1};\ v_{i-1,j};\ a_k) \\ &= \left\{\beta_0(i, j, k)\left[\frac{\gamma_0(i, j, k)}{\beta_0(i, j, k)}\right]^{v_{ij}}\right\}^{(1-v_{i,j-1})(1-v_{i-1,j})} \\ &\cdot \left\{\beta_1(i, j, k)\left[\frac{\gamma_1(i, j, k)}{\beta_1(i, j, k)}\right]^{v_{ij}}\right\}^{(1-v_{i,j-1})v_{i-1,j}} \\ &\cdot \left\{\beta_2(i, j, k)\left[\frac{\gamma_2(i, j, k)}{\beta_2(i, j, k)}\right]^{v_{ij}}\right\}^{v_{i,j-1}(1-v)_{i-1,}} \\ &\cdot \left\{\beta_3(i, j, k)\left[\frac{\gamma_3(i, j, k)}{\beta_3(i, j, k)}\right]^{v_{ij}}\right\}^{v_{i,j-1}\,v_{i-1,j}} \end{aligned}\right\}. \tag{4}$$

The recognition system is to compute, for the unknown pattern v, the conditional probabilities $p_kP(v \mid a_k)$, $k = 1, 2, 3, \cdots, c$, and then select the largest probability. Eqs. (1) and (4) (or Table I) may be used directly, or equivalently, any monotonic function of $p_kP(v \mid a_k)$ may be computed, and the (algebraically) largest probability selected. An inspection of (4) suggests the use of a logarithmic transformation to facilitate network mechanization. Since $\ln x$ is a monotonically increasing function of x, the system remains optimum, if it computes $\ln p_kP(v \mid a_k)$'s denoted as $T(v \mid a_k)$'s, and selects the algebraically largest. Using (1) and (4), the following expression, after some algebraic manipulations, is obtained:

$$\left.\begin{aligned} T(v \mid a_k) &= \ln p_kP(v \mid a_k) \\ &= b(k) + \sum_{i,j} w_1(i, j, k)v_{ij} \\ &\quad + \sum_{i,j} w_2(i, j, k)v_{i,j}v_{i,j-1} \\ &\quad + \sum_{i,j} w_3(i, j, k)v_{i,j}v_{i-1,j} \\ &\quad + \sum_{i,j} w_4(i, j, k)v_{i,j-1}v_{i-1,j} \\ &\quad + \sum_{i,j} w_5(i, j, k)v_{i,j}v_{i,j-1}v_{i-1,j} \end{aligned}\right\}, \tag{5}$$

where summation indices i and j run through the entire character field from 1 to r and s, respectively. The bias $b(k)$ and weights w's are given by the following equation:

$$\left.\begin{aligned} b(k) &= \ln p_k + \sum_{i,j} \ln \beta_0(i, j, k) \\ w_1(i, j, k) &= \ln \frac{\gamma_0(i, j, k)\beta_2(i, j+1, k)\beta_1(i+1, j, k)}{\beta_0(i, j, k)\beta_0(i, j+1, k)\beta_0(i+1, j, k)} \\ w_2(i, j, k) &= \ln \frac{\beta_0(i, j, k)\gamma_2(i, j, k)}{\gamma_0(i, j, k)\beta_2(i, j, k)} \\ w_3(i, j, k) &= \ln \frac{\beta_0(i, j, k)\gamma_1(i, j, k)}{\gamma_0(i, j, k)\beta_1(i, j, k)} \\ w_4(i, j, k) &= \ln \frac{\beta_0(i, j, k)\beta_3(i, j, k)}{\beta_1(i, j, k)\beta_2(i, j, k)} \\ w_5(i, j, k) &= \ln \frac{\gamma_0(i, j, k)\beta_1(i, j, k)\beta_2(i, j, k)\gamma_3(i, j, k)}{\beta_0(i, j, k)\gamma_1(i, j, k)\gamma_2(i, j, k)\beta_3(i, j, k)} \end{aligned}\right\}. \tag{6}$$

To accommodate the boundary points, the following definitions are used in (6):

$$\left.\begin{aligned} &\frac{\beta_2(i, s+1, k)}{\beta_0(i, s+1, k)} = 1, \\ &\text{and} \\ &\frac{\beta_1(r+1, j, k)}{\beta_0(r+1, j, k)} = 1, \end{aligned}\right\} \tag{7}$$

for all i, j, and k.

Because of neighbor dependence, $T(v \mid a_k)$ is not a linear function of v_{ij}'s, but, as indicated in (5), is a weighted sum of v_{ij}'s and the double and triple products of v_{ij}'s. The weights are logarithms of ratios of conditional probabilities β's and γ's. The first term on the right-hand side of (5) represents a constant bias. If the assumption of nearest-neighbor dependence is valid, then the decision rule given in (5) with the weights given in (6) is optimum. On the other hand, if the nearest-neighbor model serves merely as an approximation to the unknown distribution, then the formulas (6)

do not necessarily yield the best possible values. It is to be noted that, if the range of dependence were to be increased, products of higher orders would appear in the expression for T.

A Nearest-Neighbor Network

A mechanization of the nearest-neighbor system, based upon (5), is shown in Fig. 3. (The mechanization, as shown in the figure, is sufficiently general to be representative of the larger class of neighbor dependence systems.) The mechanization consists of three layers, as follows.

The first layer receives the binary signal matrix $[v_{ij}]$ as input, and forms products of neighboring point signals. Since the signal is binary, only AND gates are required. Each signal point, except those at the north and west boundaries, requires three two-input gates and one three-input gate. The configuration of these gates is shown in the two diagrams in Fig. 4. Each circle in Fig. 4 denotes a signal point. Each line segment (Fig. 4(a)) represents a two-input gate fed by the point signals which the line segment connects. Similarly, each triangle (Fig. 4(b)) represents a three-input gate.

The outputs of the first layer are the input signals, v_{ij}'s, and the double and triple products of neighboring signals, as indicated in (5). These outputs are binary, and feed the second layer.

The second layer consists of a set of weighting and summing networks, one for each character of the alphabet. The outputs of the first layer feed these networks in parallel. In addition, each summing network has a constant bias, to realize term $b(k)$ of (5). The weights, given in (6), may be negative as well as positive. The weighted sum is, therefore, the $T(v|a_k)$ of (5). The set of $T(v|a_k)$, $k=1, 2, \cdots, c$, constitutes the outputs, which are analog.

The final layer consists of the usual process of selecting the (algebraically) largest output of the second layer. Since T's are nonpositive, the selection may simply be based upon the least magnitude. The output of the final layer is the recognition decision.

Special Case of Independence

For comparison, a special case where the point signals are mutually independent may be considered. Eq. (1) then reduces to the following form:

$$P(v|\,a_k) = \prod_{i,j} P(v_{ij}|\,a_k), \tag{8}$$

which amounts to stating that parameter $\beta_m(i, j, k)$ and, consequently, parameter $\gamma_m(i, j, k)$, are independent of the index m. The subscript m can then be dropped. Thus

$$\left.\begin{aligned} &\beta_m(i,j,k) = \beta(i,j,k) \\ \text{and} \\ &\gamma_m(i,j,k) = \gamma(i,j,k) = 1 - \beta(i,j,k) \end{aligned}\right\}, \tag{9}$$

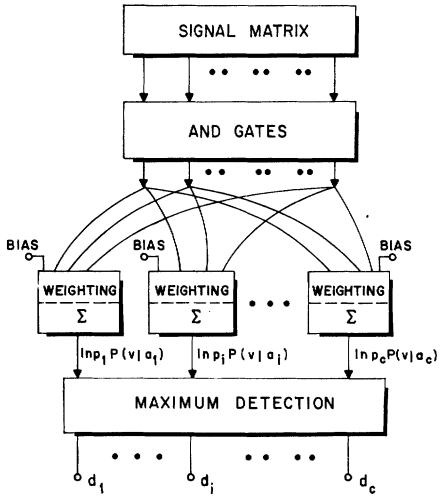


Fig. 3—Neighbor-dependence recognition network.

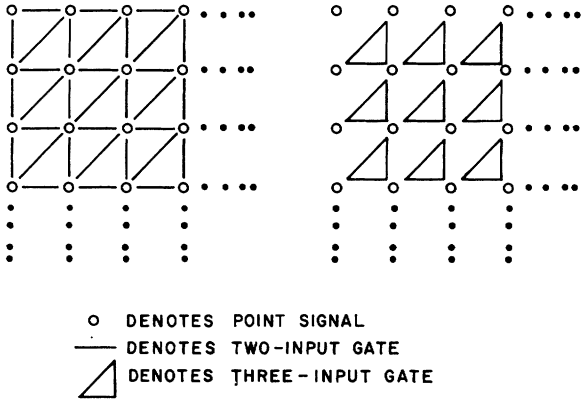


Fig. 4—Arrangement of AND gates.

for $m=0$, 1, 2, and 3. Now $\beta(i, j, k)$ is simply the probability that the (i, j)th element of character a_k is 0; $\gamma(i, j, k)$ is the corresponding probability that the element is 1.

By virtue of (9), all of the weights, except $b(k)$ and $w_1(i, j, k)$, as defined in (6), vanish, and (5) becomes

$$T(v|\,a_k) = b(k) + \sum_{i,j} w_1(i,j,k)v_{ij}, \tag{10}$$

where

$$\left.\begin{aligned} b(k) &= \ln p_k + \sum_{i,j} \ln \beta(i,j,k), \\ \text{and} \\ w_1(i,j,k) &= \ln \frac{\gamma(i,j,k)}{\beta(i,j,k)} \end{aligned}\right\}. \tag{11}$$

For this special case, $T(v|a_k)$ is linear in v_{ij}'s, and the corresponding recognition network consists of a set of weighting and summing networks and a maximum selection circuit. The diagram is the same as that in Fig. 3, except that the layer of AND gates is no longer required.

A Geometric Interpretation

Consider the case of independence. The decision rule (10) partitions the $r \times s$ dimensional space, where the discrete signal space v is imbedded, with a set of $c(c-1)/2$ hyperplanes:

$$\sum_{i,j} \{w_1(i,j,k) - w_1(i,j,m)\} v_{ij} + b(k) - b(m) = 0 \quad (12)$$

for all k and m, with $k \neq m$. There is one hyperplane between the members of each pair of characters. The hyperplane separating the kth and mth pattern classes is perpendicular to the vector joining the vectors which represent the sets of weights $\{w_1(i,j,k)\}$ and $\{w_1,(i,j,m)\}$, and is at a distance of $[b(m)-b(k)] \div$ (the length of that joining vector). All of these hyperplanes are not independent; a change in any set of weights alters the $c-1$ hyperplanes associated with that set.

For the dependence model, the set of weighting and summing networks and the maximum selection circuit similarly mechanize a set of $c(c-1)/2$ partitioning hyperplanes. However, these hyperplanes are in a different space. They are not hyperplanes in the original $r \times s$ space, but rather are in an extended space, the extension being introduced by combining the signal v_{ij}'s, as mechanized by the AND gates. This interpretation may be made clearer by considering a simple example. An alphabet of two characters consists of sets of patterns $\{(0,0), (1,1)\}$ and $\{(0,1), (1,0)\}$, respectively, as shown in Fig. 5(a). These two sets are not linearly separable. By introducing the combined signal $v_1 \cdot v_2$, the given pattern classes are now represented by $\{(0,0,0), (1,1,1)\}$ and $\{(0,1,0), (1,0,0)\}$, respectively, as shown in Fig. 5(b). This new configuration in the three-dimensional space is now linearly separable, and the separating hyperplane in this extended space is mechanizable by the dependence model. Loosely speaking, by generating the combined features, the original pattern classes are spread further in the extended space, thus enhancing the possibility of linear separation.

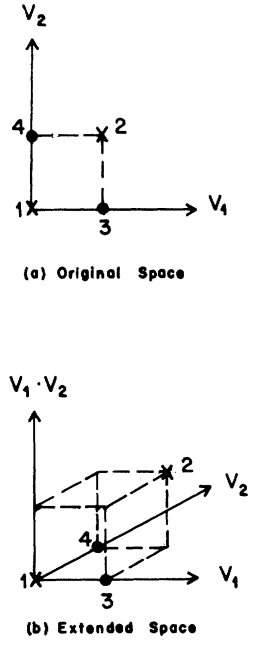
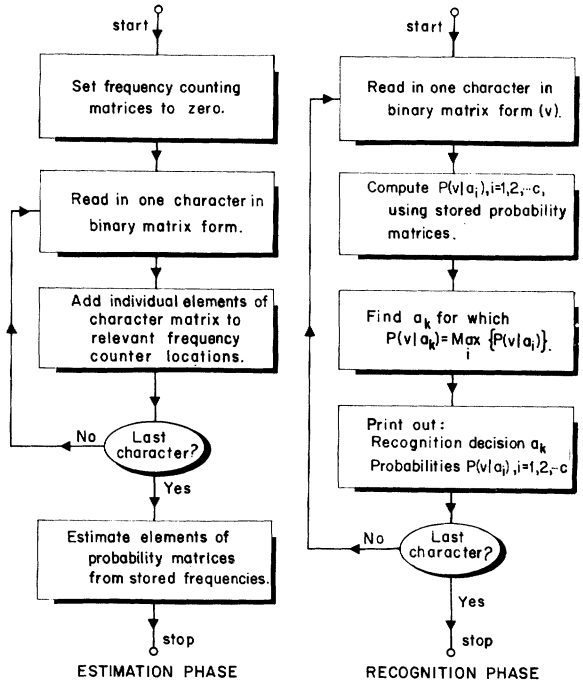


Fig. 5—Extension of signal space.

A Computer Simulation

Some simulations of the recognition networks were performed on a digital computer (the Burroughs 220) to obtain a relative evaluation. The computer program consists of two main operations—statistical estimation and recognition—which are described here briefly. A flow chart of the simulation program is shown in Fig. 6.

For statistical estimation, the computer is provided with binary quantizations of samples of each character in the alphabet that it is later expected to recognize, each sample accompanied by proper identification of the character represented by the sample. The parameters to be estimated are the probabilities $\beta_m(i,j,k)$ (or $\gamma_m(i,j,k)$), as defined in Table I. These probabilities, in turn, determine the values of weights as stated by (6). The relative frequencies of occurrence of the various samples are used as estimators for the parameters. For example, the ratio of the number of samples of character a_2 having ZERO at location (4, 6), ONE at location (3, 7), and ZERO at location (4, 7) to the number of samples of character a_2 having ZERO and ONE, respectively, at location (4, 6) and (3, 7) is the estimated value of β_1 (4, 7, 2).



The sequence of operations is the same for methods 1 and 2; the program can carry out the computations for either one of the methods singly, or for both simultaneously.

Fig. 6—Flow chart of computer simulation.

However, the following exceptions are made (primarily for small sample sizes) to avoid the appearance of zero factors in the products of (1), when evaluating probabilities during the recognition phase. If the estimated value of a β is $0/N$ (or N/N), where N denotes the number of samples pertinent to estimating that β, then the β is replaced by ϵ/N(or $1-\epsilon/N$). (Here, ϵ is a small positive constant.) If the estimate of a β is $0/0$—that is, there is no sample pertinent to estimating this conditional probability—then this β is taken as $1/2$. For a matrix of size $r\times s$, the number of parameters to be estimated is $4rs-2(r+s)+1$ per character of the alphabet. The occurrence of characters is taken as equally probable ($p_1=p_2\cdots=p_c$) in the simulation. If desired, other distribution can be used, and, if necessary, the values can be estimated from the sample.

Recognition is based upon the stored parameter values obtained in the estimation phase. The program computes, for each input pattern, the associated conditional probabilities $p_kP(v|a_k)$ or $T(v|a_k)$, $k=1, 2, \cdots, c$, then selects the largest probability, and classifies the input pattern as the character corresponding to the largest conditional probability.

Examples of Hand-Printed Alphanumeric Characters

The performance of any recognition method depends not only upon the system itself, but also upon the class of characters encountered by the system. Comparisons among various recognition methods are difficult, especially if the methods do not operate on the same data and the same pattern representation. To establish a reference for comparison, the results reported here are based upon the set of hand-printed characters prepared and used by W. H. Highleyman[6,7] of the Bell Telephone Laboratories.

The data consist of 50 sets of 36 hand-printed characters (ten arabic numerals and 26 upper-case alphabetic letters), each set printed by a different person. These persons were required to print neatly on $\frac{1}{4}$-inch quadrilled paper at a size approximating the ruled boxes on the paper. (Some samples of the data are given by Highleyman[7] in Figs. 9 and 10.) The data were then automatically reduced to a 12×12 binary matrix by an optical matrix scanner, and encoded on punched cards. The characters were roughly centered by using center of gravity alignment. However, the character size was not normalized; the size variation is about two to one. These data cards, employed through the courtesy of W. H. Highleyman, are the input to the simulation program. Computer printouts of two sets of the quantized data of Highleyman are reproduced in Fig. 7 (next page.) (The characters have been repositioned in the figure to conserve space.)

For simulation of the recognition methods discussed, the data—all 50 sets—were read into the computer to establish the weights of the recognition network. The same 1800 patterns were then read, one by one, into the computer for recognition. No rejection option was allowed in the simulation reported here. Two computer runs were made, one for numerals, the other for numerals and letters. For the numerals alone, the nearest-neighbor method yields a recognition rate of 97.2 per cent and an error rate of 2.8 per cent. For the alphanumeric case, the corresponding rates are 93.3 per cent and 6.7 per cent. The distribution of errors for the alphanumeric case is tabulated in Table II.

In any experiment of this sort, the absolute performance is not too significant. Rather, relative performance is usually more meaningful. To provide a reference for comparison, and to ascertain the effect of neighbor dependence upon recognition performance, the linear system as characterized by (10) was also simulated, and operated upon the same data (numerals only). The resultant error rate of 20.4 per cent is appreciably higher than that of the nearest-neighbor method. The effect of dependence is significant. The use of (1) offers a better approximation to the unknown distribution than does that of (8).

Another simulation trial was made. Arbitrarily, the first 40 sets of alphanumeric data were used in the estimation phase to establish the weights of the recognition network. The remaining ten sets of data were then read as unknown, for recognition. The resultant recognition rate was 58.3 per cent. The two contributing factors in the decrease in performance are the smallness of the design sample size and the primitiveness of the pattern representation.

No accurate account of computer time was kept; the calculation of weights for the entire alphabet of 36 characters from 1800 samples took approximately one-half hour, and the recognition took about 45 seconds per sample. No special effort was made to minimize computation time, and the calculation of conditional probabilities in the recognition phase was carried out sequentially, one pattern class at a time.

The sample is too small to permit conclusive comparisons among various methods, but it is hoped that the results do provide some indication of relative performance. Admitting the inadequacy of sample size, it is of interest to compare the recognition results of several methods operating upon the same data. Results are summarized in Table III. Methods 1 and 2 refer, respectively, to the nearest-neighbor model (1) and the independence model (8), described in this paper. Methods 3 and 4 are described by Highleyman[6,7]; only numeric results are reported in Highleyman.[6]

[6] W. H. Highleyman, "Linear Decision Functions with Application to Pattern Recognition," Ph.D. dissertation, Elec. Engrg. Dept., Brooklyn Polytechnic Institute, Brooklyn, N. Y.; June, 1961. A summary appears in Proc. IRE, vol. 50, pp. 1501–1514; June, 1962.

[7] W. H. Highleyman, "An analog method for character recognition," IRE Trans. on Electronic Computers, vol. EC-10, pp. 502–512; September, 1961.

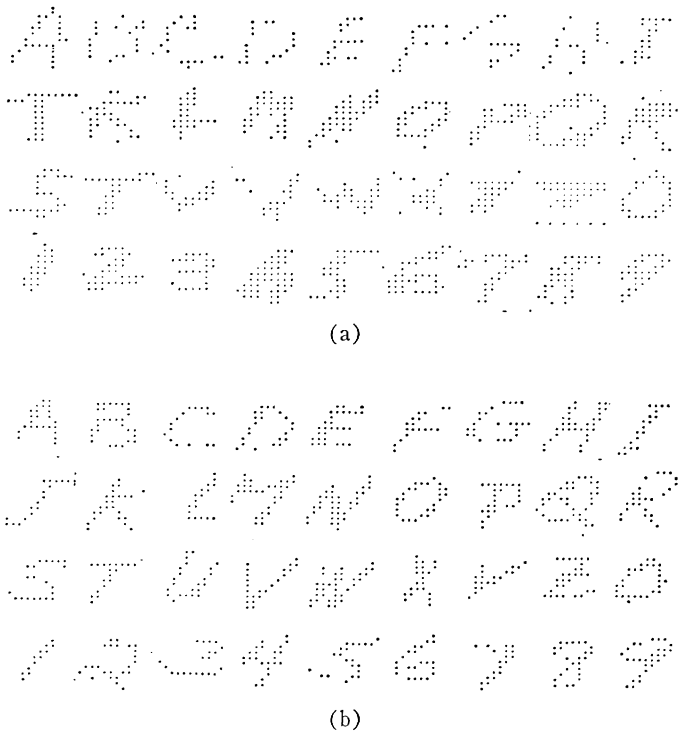


Fig. 7—Computer printouts of two sets of quantized alphanumeric characters, read from punched cards supplied by W. H. Highleyman.

TABLE II

DISTRIBUTION OF ERRORS, 50 SETS OF 36 HAND-PRINTED CHARACTERS

Characters	Number of Errors	Distribution of Errors
A	1	R
B	1	6
C	2	L, 6
D	4	P, Q, U, 0
E	2	F, O
F	3	I, T, Z
G	3	J, 6(2)
H	3	N(3)
I	20	J, 1(19)
J	5	I, L, T(2), U
K	0	
L	2	I, X
M	2	A, H
N	3	K, O, 4
O	6	D(2), 0(4)
P	3	F(2), R
Q	2	O, 9
R	3	H, O, P
S	3	G, J, 8
T	2	7(2)
U	2	V, 4
V	1	Y
W	2	N, V
X	3	K, Y(2)
Y	3	U, V, 1
Z	1	I
0	10	O(5), P, Q, 3, 4, 6
1	0	
2	3	E, I
3	1	8
4	2	A, 8
5	5	B, S(3), 6
6	4	L(2), X(2)
7	6	I(2), J, X, 2, 9
8	4	B(2), Y, 9
9	4	A, J, 7, 8

TABLE III

COMPARISON OF RECOGNITION RESULTS (BASED UPON 50 SETS OF HAND-PRINTED CHARACTERS)

Method	Numeric			Alphanumeric*	
	Recognition Rate	Error Rate	Rejection Rate	Recognition Rate	Error Rate
1) Nearest-Neighbor; Eq. (1)	97.2 per cent	2.8 per cent	none*	93.3 per cent	6.7 per cent
2) Linear; Eq. (8)	79.6 per cent	20.4 per cent	none*	(not simulated)	
3) Highleyman[6]	94.0 per cent	4.2 per cent	1.8 per cent	(not given)	
4) Highleyman[7]	83.0 per cent	17.0 per cent	none*	77.2 per cent	22.8 per cent

* Rejection parameter set at zero; error rate is reduced if rejection is used.

CONCLUSIONS

The statistical approach of an earlier paper[1] is followed here. This paper illustrates the manner in which the detailed structure of a recognition network can be systematically derived from the *a priori* knowledge of the functional form of probability distributions. The resultant networks are, in general, nonlinear—linear, if statistical independence is valid. A simple nonlinear recognition network is derived to accommodate interdependence and nonlinear relations among signals. Formulas for recognition weights are also obtained, which in turn lead to a simple, straightforward method of estimating the values of the weights based upon samples. The nearest-neighbor method is simulated to verify that the effect of dependence upon recognition performance is significant, and performance results based upon a rather small sample of hand-printed characters are compared with simulation results of some previously published methods.

ACKNOWLEDGMENT

The author wishes to thank his colleagues, J. W. Seward and Miss R. C. Baldwin of Burroughs Laboratories, Paoli, Pa., for programming the computer simulation. Thanks are also due Dr. W. H. Highleyman, of Bell Telephone Laboratories, Murray Hill, N. J., for making available the hand-printed characters in binary matrix representation, which constitute the input of the simulation and permit comparisons (by virtue of the use of identical inputs) among recognition methods.