Two Viewpoints of k-Tuple Pattern Recognition

ROB J. ROY, MEMBER, IEEE, AND JAMES SHERMAN, STUDENT MEMBER, IEEE

Abstract—**This paper presents two viewpoints of the *k*-tuple pattern recognition scheme proposed by Browning and Bledsoe. The first shows that *k*-tuple pattern recognition is a statistical approximation technique. In effect, the recognition is accomplished by approximating a higher order probability distribution by use of the first-order distributions. Using this viewpoint, and Lewis' measure of characteristic selection, several alternative approximations are offered. The second viewpoint is that recognition is a special case, or subclass, of a Φ learning machine. It can be shown that if the input pattern vector *X* is first processed by a Φ-processor (in this case a *k*th order polynomial) and then certain terms discarded, the resulting learning machine is identical to a *k*-tuple pattern recognition machine.**

k-Tuple Pattern Recognition

THE technique of *k*-tuple pattern recognition[1]–[4] (Browning-Bledsoe pattern recognition) examines certain *k*-tuples of pattern points and attempts to separate patterns on the basis of these subpatterns.

Consider then a 2-dimensional pattern containing N binary pattern points. From the N pattern points, M sets of k pattern points are selected. These k-tuples may be chosen in a random manner, but would generally be selected to be disjoint, so that the measurements on the patterns will more nearly represent independent pattern characteristics or features. Each k-tuple, or pattern feature, has 2^k possible values, corresponding to the 2^k possible binary patterns. A given pattern X is then described by an M-dimensional vector, each component of the vector being a feature value. Thus, the pattern X has been transformed from an N-dimensional vector with binary components to an M-dimensional vector with 2^k valued components. The original pattern can be reconstructed only if each pattern point is included in at east one of the M k-tuples.

Assume that there are R categories. For each of the possible R categories and for each of the feature values, a number M_{ijr} is stored, as in Fig. 1. The subscripts denote the following:

- i ith category
- j jth feature
- r rth value of feature j.

The memory element M_{ijr} is chosen to be the logarithm of the joint frequency of occurrence of category i and the rth value of feature j. These values for the entries in the memory matrix are chosen on the basis of a learning set of data. Once the learning phase is completed, the test data are presented. The decision making is then based upon a set of discriminant functions, one for each of the R categories. The discriminant function for the ith category is given by

FEATURE	CATEGORY	
$\alpha_1 = X_1\ X_2$	R_1	R_2
0 0 0 1 1 0 1 1	LOG FREQ.$(R_1, X_1X_2 = 00)$ $M_1(\alpha_1 = 10) = M_{112}$	
$\alpha_2 = X_3\ X_4$		

Fig. 1. Browning–Bledsoe (*k*-tuple recognition).

$$g_i(X) = \sum_{r=0}^{2^k-1} \sum_{j=1}^{M} \alpha_{jr}(X) M_{ijr} \tag{1}$$

where

α_{jr}	rth value of feature j
M	number of k-tuples
$\alpha_{jr}(X)$	equals 1 if α_{jr} is found in pattern X equals 0 if α_{jr} is not found in pattern X
X	binary input pattern.

The correct category is chosen as that category which has the largest discriminant function.

Since the memory elements are the logarithms of the joint frequency functions $f(R_i, \alpha_{jr})$, the discriminant function could be expressed as the following product

$$g_i(X) = \prod_{r=0}^{2^k-1} \prod_{j=1}^{M} [f(R_i, \alpha_{jr})]^{\alpha_{jr}(X)}. \tag{2}$$

Alternatively, the discriminant function can be expressed in the form of a disjunctive logic expression. The general form for a 2-tuple discriminant function would be

$$g(X) = \sum_{\text{all } \alpha} (M_0 \bar{x}_i \bar{x}_j + M_1 x_i \bar{x}_j + M_2 \bar{x}_i x_j + M_3 x_i x_j)$$

where x_i, x_j are the ith and jth binary pattern points comprising the 2-tuple.

For this particular form, the substitution $\bar{x}_i = 1 - x_i$ and $\bar{x}_j = 1 - x_j$ can be made. Consequently

$$g(X) = \sum_{\alpha} (M_3 + M_0 - M_1 - M_2) x_i x_j + (M_2 - M_0)\ x_i + (M_1 - M_0) x_j + M_0$$

$$= \sum_{\alpha} w_{ij} x_i x_j + w_i x_i + w_j x_j + w_T. \tag{3}$$

Manuscript received March 7, 1967; revised June 16, 1967. This work was supported by the National Aeronautics and Space Administration under Research Grant NGR-33-018-014.

The authors are with the Rensselaer Polytechnic Institute, Electrical Engineering Department, Troy, N. Y.

The following two viewpoints of this technique are based upon the different forms of (2) and (3).

Statistical Approximation[5]–[8]

A well-known result of pattern recognition[9] is that the minimum error rate is obtained if classification is based on selection of that category which has the maximum joint probability $P(\boldsymbol{X}, i) = P(\boldsymbol{X}/i)P(i)$. The conditional probabilities $P(\boldsymbol{X}/i)$ are M-dimensional

$$P(\boldsymbol{X}/i) = P(\alpha_1, \alpha_2, \ldots \alpha_M/i)$$

where α_j is the jth feature of pattern $\boldsymbol{X}$. Clearly, the measurement and storage of an Mth-order conditional probability represents an extremely difficult problem. For this reason, the Mth-order probabilities must be approximated using lower order probabilities. Fortunately, this problem has been investigated by Lewis.[6] The results of this investigation are as follows. If a set of lower order distributions is given, the possible approximations to the high order distributions are those functions which reduce to the given lower order distributions when properly summed. It is also assumed that there is a fixed amount of information inherent in a process that generates a finite set of sequences. Some of this information is contained in the probability distribution and the rest in the reception of the sequences. The approximation chosen should be such that a maximum of information is transmitted by the reception of the sequences, and minimum information by the probability distribution. In this manner the approximation will be as unbiased or as random as possible. An approximation which satisfies this criterion will have a flat distribution, as a peaked distribution gives a considerable amount of a priori knowledge about the sequences. A flat distribution gives very little a priori information about which sequence will occur, and the sequences themselves give a maximum amount of information.

The solution to the preceding requirements is a higher order distribution made up of a product of its lower order component distributions such that the high order distribution reduces to the given lower order component distributions when properly summed. Lewis has shown that given a set of lower order probabilities $P_1, P_2, \ldots P_n$ such that the product approximation

$$P = P_1 P_2 \ldots P_n \tag{4}$$

satisfies the summation criterion, this approximation contains the smallest amount of information of all possible approximations which satisfy the summation criterion. Consequently, this is the best way to use the partial information available. The product approximation uses the data retained to approximate the total information available, thereby reducing the storage requirements considerably.

Relationship Between k-Tuple Pattern Recognition and Statistical Approximation

The k-tuple memory matrix is filled by storing the logarithm of the joint frequency of each feature value α_{jr} and each category i. Thus, for R categories and M features (k-tuples) the memory matrix has $RM2^k$ entries of the form $\log P\ [\alpha_{jr}, i]$. For a given input pattern, the features are extracted and the memory matrix is searched to obtain the proper set of logarithms for all categories. These logarithms are then summed, for each category, over the set of input feature values. Thus

$$g_i(\boldsymbol{X}) = \sum_j \log P[\alpha_{jr}, i]. \tag{5}$$

As a result, there are R discriminant functions $g_i(\boldsymbol{X})$. The proper category is chosen to be that category with the maximum discriminant function. Note, however, that these R discriminant functions are *exactly* equal to the probability approximations $P(\boldsymbol{X}, i)$ given by Lewis.

$$g_i(\boldsymbol{X}) = \prod_j P(\alpha_j/i)P(i) \approx P(\boldsymbol{X}, i). \tag{6}$$

Consequently, for the partial data used, this is the best use of the available information. Therefore it is not surprising that these techniques gave the best results, although Bledsoe and Bisson[4] state that "Even though the maximum likelihood method scored better than the other methods tried in this small study, it would be a mistake to claim that such a result should have been expected beforehand." On the contrary, in the absence of any a priori information concerning the distribution of the conditional probability distributions, this technique should be expected to yield the best results.

Φ-Processor Learning Machine

The general class of learning machines which use linear, quadratic, or polynomial type of decision surfaces can be constructed using a Φ-processor followed by a linear machine. Figures 2–4 show a linear, quadratic, and Φ learning machine, respectively. A Φ learning machine is a Φ-processor followed by a linear machine. The input to the Φ-processor is the pattern vector $\boldsymbol{X}$, and the output is a vector $\boldsymbol{Y}$ whose components $f_i(\boldsymbol{X})$ are linearly independent, real, single-valued functions of $\boldsymbol{X}$. The components $f_i(\boldsymbol{X})$ are given by

1) Linear decision surfaces: $f_i(\boldsymbol{X}) = x_i,\ i = 1, \ldots, N$
2) Quadric decision surfaces: $f_i(\boldsymbol{X}) = x_j^n x_k^m$
$j, k = 1, \ldots, N \qquad n, m = 0$ and 1
3) kth-order polynomial surfaces: $f_i(\boldsymbol{X}) = \prod_{j=1}^{k} x_{p_j}^{n_j}$
$n_j = 0$ and $1\ j = 1, \ldots, k \qquad p_j = 1, \ldots, N$.

Once the form of the Φ-processor is decided upon, the weights of the linear machine following the Φ-processor are found by the methods used for linear decision functions.

Note that a k-tuple machine is equivalent to a kth-order polynomial Φ-machine with only some of the terms re-

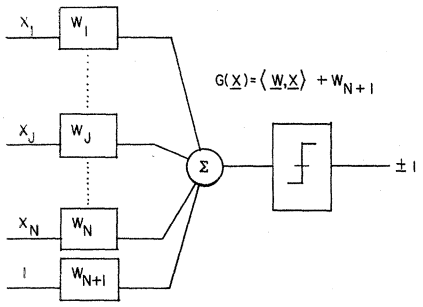

Fig. 2.

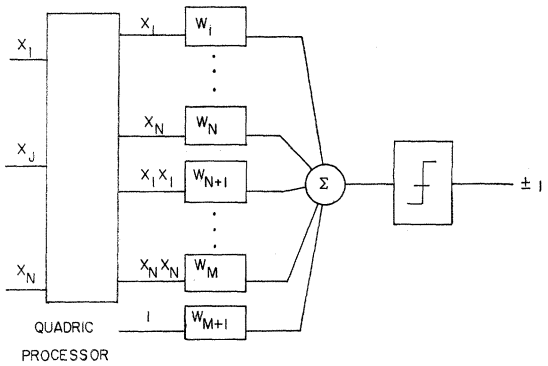

Fig. 3.

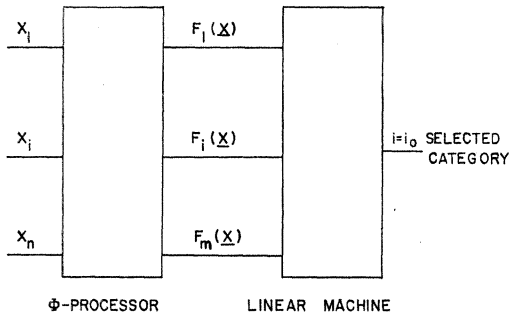

Fig. 4.

tained. Such a Φ-machine has $R(M2^k - M + 1)$ weights in the linear machine compared to $R(2^kM)$ matrix elements (or weights) in the Browning and Bledsoe machine (R categories, M k-tuples).

This correspondence will be shown for $k = 2$ and can be readily extended for any $k \leq N$. A more precise statement of the correspondence between the k-tuple Browning and Bledsoe machine and the k-order polynomial Φ-machine will then be given.

The discriminant function for the ith-category in the Browning and Bledsoe method is given by (1). This discriminant function contains terms of the form

$$x_\mu x_\nu M_{ij3} + x_\mu \bar{x}_\nu M_{ij2} + \bar{x}_\mu x_\nu M_{ij1} + \bar{x}_\mu \bar{x}_\nu M_{ij0} \qquad (7)$$

where x_μ and x_ν are the components of the jth 2-tuple.

A possible structure for implementing a 2-tuple Browning and Bledsoe machine is shown in Fig. 5. The discriminators are similar to Φ-machines, where the M_{ijr} are the weights of a linear machine. Although the outputs

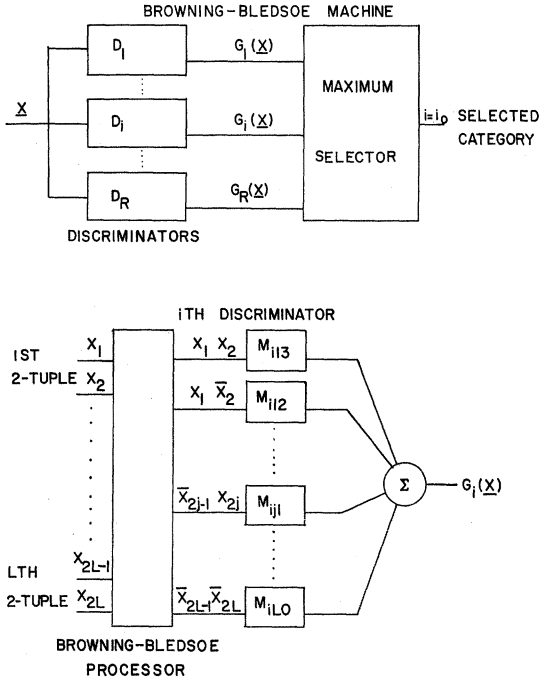

Fig. 5.

of the processor are real, single-valued functions of $\boldsymbol{X}$ independent of the weights, they are not linearly independent functions. The components of each 2-tuple must satisfy the disjunctive relationship

$$x_\mu x_\nu + x_\mu \bar{x}_\nu + \bar{x}_\mu x_\nu + \bar{x}_\mu \bar{x}_\nu = 1. \qquad (8)$$

Using (8), the Browning and Bledsoe discriminators can be transformed into equivalent Φ-machines such that $g_i(\boldsymbol{X})$ for the Browning and Bledsoe machine equals $g_i(\boldsymbol{X})$ for the Φ-machine.

The terms of (7) become

$$M_{ij3}x_\mu x_\nu + M_{ij2}x_\mu(1 - x_\nu) + M_{ij1}(1 - x_\mu)x_\nu + M_{ij0} \times (1 - x_\mu)(1 - x_\nu) = x_\mu x_\nu(M_{ij3} - M_{ij2} - M_{ij1} + M_{ij0}) + x_\mu(M_{ij2} - M_{ij0}) + x_\nu(M_{ij1} - M_{ij0}) + M_{ij0}. \qquad (9)$$

Considering now a Browning and Bledsoe machine with M 2-tuples where the 2-tuples of measurements are written as

$$x_{2j-1} \text{ and } x_{2j} \quad j = 1, \ldots, M$$

the discriminant functions become

$$g_i(\boldsymbol{X}) = \sum_{j=1}^{M} \{x_{2j-1}x_{2j}[M_{ij3} - M_{ij2} - M_{ij1} + M_{ij0}] + x_{2j-1}[M_{ij2} - M_{ij0}] + x_{2j}[M_{ij1} - M_{ij0}]\} + \sum_{j=1}^{M} M_{ij0}. \qquad (10)$$

This is a quadric Φ-machine with $2M$ linear terms and M cross product terms. Since there are R categories, the total number of weights is $(3M + 1)R$. This can be extended to the case of a Browning and Bledsoe machine with M k-tuples where the k-tuples of measurement

are written as

$$x_{kj-q}\ j = 1, \ldots, M;\ q = 0, 1, \ldots, k-1.$$

The terms of the equivalent kth order polynomial Φ-machine that are retained can be written as

$$f_p(\boldsymbol{X}) = \prod_{q=0}^{k-1} x_{kj-q}^{n_q} \quad n_q = 0 \text{ and } 1;\ q = 0, 1, \ldots, k-1$$

$$j = 1, \ldots, M$$

where $f_p(\boldsymbol{X}) = 1$ or at least one $n_q = 1$. There are $2^k - 1$ possible combinations of ordered sequences of the n_q not including all zeros. Since there are M k-tuples there are a total of $M(2^k - 1)$ functions of $\boldsymbol{X}$. The Browning and Bledsoe machine has been transformed into a linear machine in E^D space where $D = M(2^k - 1)$. For a R category machine there are $R(2^kM - M + 1)$ weights. These weights are simple linear combinations of the memory matrix elements M_{ijr} of the Browning and Bledsoe machine.

The preceding discussion points out that the k-tuple pattern recognition technique can be viewed as a degenerate form of a general kth-order machine. The degeneracy is due to the selection of only certain k-tuples. Thus, not all cross products, nor all linear terms, may be present in the k-tuple machine. Consequently, the k-tuple technique is not as general as the Φ-processing technique. The designer of the k-tuple machine has a priori assumed how the decision surfaces will separate the categories. These assumptions may not be justified for the problem chosen. For example, consider the case of a 2×2 grid with binary elements x_1, x_2, x_3, x_4. If the 2-tuples chosen are $x_1\ x_2$ and $x_3\ x_4$, then the discriminant functions are of the form

$$w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_{12}x_1x_2 + w_{34}x_3x_4 + w_T.$$

Assuming this set of 2-tuples has, in effect, assumed the type of decision surfaces on each pair of coordinates.

Notice that although the separating curves in both the $x_1 - x_2$ and $x_3 - x_4$ planes are quadric, the separating curves in the $x_1 - x_3$, $x_1 - x_4$, $x_2 - x_3$, and $x_2 - x_4$ planes are *linear*. Consequently, the choice of the k-tuples determines the form of the separating surfaces. More importantly, selection of k-tuples may leave out many of the coordinate points. This means that separation is impossible in any set of coordinates which have been omitted. The seriousness of this problem depends upon how good (in terms of error rate) the separation is using the other coordinates.

Conclusion

This paper has shown two different ways to view k-tuple pattern recognition. One viewpoint is that of statistical approximation, while the other interprets k-tuple recognition in terms of a general kth-order Φ-machine. Both viewpoints give insight into this technique, and suggest alternative approaches to pattern recognition. For example, since k-tuple recognition is not an iterative technique, the weights are determined faster than in the self-correcting Φ-machine. Consequently, the k-tuple technique may be used to initialize the weights of the Φ-machine. Then the self-correcting algorithms would be applied to obtain a better solution. This would reduce the long iteration times required to adjust the weights of the Φ-machine, yet retain the advantage of self-correction.

References

[1] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," *1959 Proc. Eastern Joint Computer Conf.*, Sandia Corp., Albuquerque, N. Mex., Reprint SCR-132, March 1960.

[2] W. W. Bledsoe, "Further results on the n-tuple pattern recognition method," *IRE Trans. Electronic Computers* (*Correspondence*), vol. EC-10, p. 96, March 1961.

[3] W. H. Highleyman, "Further comments on the n-tuple pattern recognition method," *ibid.*, p. 97.

[4] W. W. Bledsoe and C. L. Bisson, "Improved memory matrices for the n-tuple pattern recognition method," *IRE. Trans. Electronic Computers* (*Correspondence*), vol. EC-11, pp. 414–415, June 1962.

[5] G. P. Steck, "Stochastic model for the Browning-Bledsoe pattern recognition scheme," *IRE Trans. Electronic Computers*, vol. EC-11, pp. 274–282, April 1962.

[6] P. M. Lewis, II, "Approximating probability distributions to reduce storage requirements," *Inform. and Control*, vol. 2, no. 3, pp. 214–225, 1959.

[7] ——, "The characteristic selection problem in recognition systems," *IRE Trans. Information Theory*, vol. IT-8, pp. 171–178, February 1962.

[8] L. A. Kamentsky and C. N. Liu, "A theoretical and experimental study of a model for pattern recognition," in *1964 Computer and Information Science Symposium*. Washington, D. C.: Spartan, pp. 194–219.

[9] N. J. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965.