Pattern classification as interpolation in N dimensions

By P. A. V. Hall*

Pattern classifications are considered equivalent to computing a special function, and the problem in designing a machine is to reconstruct and approximate this function given only a finite set of samples from the function. Interpolation theory is applied to the problem and a variety of realisations are suggested. While the theoretical viewpoint is new and general, the realisations are shown to be equivalent to many existing solutions; however, this theory allows a systematic approach to design, with a firm and general background to convergence and error.

(Received April 1968)

It is the intention here to develop a new approach to pattern classifications, to develop a general mathematical formalism for learning and pattern recognition. The problem mainly considered is that of classifying static spatial patterns, and particular reference is given to visual pattern recognition, though the basic theory is extendible in an obvious way to all situations of static pattern classification.

Starting with the usual model for pattern classifications, the problem is shown to be equivalent to interpolation as studied in numerical analysis. Several formulae for pattern classifications are thus obtained, and realisations are suggested. The formalism is then shown to be equivalent to existing approaches to classifications (in particular, Nilsson, 1965), but is more general in the sense that any classification can be systematically tackled, and in such a way that error rates can be kept to any arbitrarily small limit.

The basic model

The usual model of a visual classification system is assumed. Patterns are projected to a retina where n quantities are measured: computations on these n values produces a further ordered set of numbers on which a decision is based (see **Fig. 1**).

Patterns are distributions of light intensity over a two-dimensional 'visual field' (and thus the totality of patterns constitutes a function space,

$$D = \{d(x, y) \colon (x, y) \epsilon V \subset R^2\}).$$

The pattern is usually projected by an optical system on to some image field to form an image pattern. (Note: because of the nature of optical systems there is necessarily a degradation in the image; the functions of the image-pattern space are band-limited in spatial Fourier components (see Gabor, 1955).)

Measurements, n in number, are made upon the image-pattern, yielding n quantities which represent the pattern. These measurements are usually made by a photo-mosaic, often rectangular. We thus obtain a point in an n-dimensional vector space, which vector space provides an approximation to the original function space. This approximation improves as the number of measurements, n, is increased; and it is clearly necessary that n should not be too small, while on the other hand, if n is too large, the system would become unnecessarily expensive. For the purposes of this paper we assume that a sufficient value of n for the problem at hand is known (from psycho-physical experiments a 20×20 mosaic is often considered sufficient for character recognition: Uhr and Vossler (1961)), and that this representation is such that there is no loss of information (that is, if two light distributions are distinct as pattern-types, they remain distinct as points in the n-dimensional approximation space).

From these n quantities we wish to make one or more computations which will yield the classification of a previously unseen pattern, and herein lies the crux of the problem: what computations do we make? Classifications in mathematics are symbolised by characteristic (or indicator) functions—the function yields a value 1 if the object belongs to the class and a value 0 otherwise (see, for example, Sz.-Nagy (1965), p. 14). Thus here we hypothesise as many characteristic functions as there are pattern classes. If we knew explicitly these characteristic functions, our problem would be trivial—but instead we have an implicit definition of the functions within ourselves. We can experiment on our-

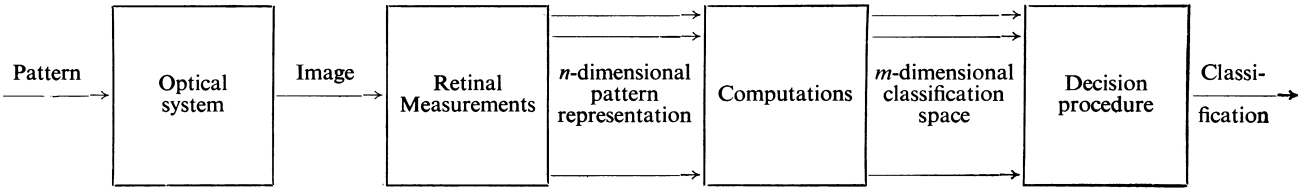

Fig. 1. Basic model for visual pattern recognition

[*] *Department of Electronic and Electrical Engineering, University College, London WC*1

selves and thereby obtain a set of samples from the characteristic functions.

An alternative view of the set of computations required is the discriminant function approach: we wish to perform one computation per pattern-class and then choose the maximum value to determine the classification (cf. Nilsson, 1965).

We unite the ideas of the preceding paragraphs and reduce our model and problem as follows:

Model: A function $d(x, y)$ representing the light distribution in the visual field is mapped to an n-dimensional space without loss of information; the n-dimensional pattern after the retina is mapped by an unknown function f to m quantities, one for each class, to give a point in an m-dimensional 'decision' space in which there is a standard decision procedure: this standard decision procedure could be either 'select the maximum', or select all those classifications whose value is greater than some threshold (say $\frac{1}{2}$); the former resolves all ambiguities and could further allow for uncertainties in regions between classifications, and the latter permits ambiguities, which may be desirable.

Problem: We are given a set of samples (N in number) of the function f^2 ($f\colon R^n \to R^m$) and must later use these to guess at the function values given some arbitrary pattern (a point in R^n). That is, we must interpolate between the sample points.

The samples are obtained as previously indicated: by objective experimentation in the real world. Sample patterns are presented to a person, or persons, and the classification point found by determining the set of m values, 0 or 1, indicating the classifications to which the pattern belongs.

The solution of the problem is now in a sense trivial, for interpolation in one dimension has been well-studied in numerical analysis, and the ideas generalise very readily to many dimensions. We turn now to this subject.

Interpolation in *n* dimensions

A function $f\colon R^n \to R^m$ can be considered as m functions $f_1, f_2, \ldots, f_m$ each mapping R^n to the real line, and we need only consider interpolation for each function separately.

A general solution to the interpolation problem, biased somewhat to our problem, is given below. The particular case of one dimension can be found in any standard textbook (e.g. Lanczos, 1957; Hamming, 1962); interpolation in two dimensions has also attracted some interest (Clenshaw and Hayes, 1965).

We wish to interpolate for a function f^2 ($R^n \to R$) given a set of samples from the function $\{(x_k, f(x_k))\colon k = 1, 2, \ldots, N\}$. Let the samples be drawn from some subset X of R^n, and let μ be some measure defined on X with $\mu(X)$ finite (for bounded sets X, μ will in general be the usual interval measure). The samples must then be distributed according to this measure μ, either at 'nodal points' of some 'mesh', or randomly with respect to the probability measure derived from μ by setting $p(Y) = \mu(Y)/\mu(X)$ with $Y \subset X$. Let $w(x)$ be a weighting function defined on $X (w(x) \geqslant 0)$. Let $\{\phi_i(x)\colon i = 0, 1, 2, \ldots\}$ be a complete orthogonal set of functions defined on X, orthogonality being with respect to both weighting function and measure.

$$\int_X \phi_i \phi_j w d\mu = 0 \quad \text{if} \quad i \neq j.$$

For later use in interpolation (or any analytic process) we approximate f by a linear combination of the first $M + 1$ of the ϕ_i's.

$$f \simeq g = \sum_{i=0}^{M} a_i \phi_i.$$

The coefficients a_i are obtained in such a way that they minimise some criterion of error (usually a least squares fit at the sample points). Clearly for non-triviality $M + 1 \leqslant N$.

We consider firstly the least squares criterion, and several cases arise.

(i) The functions ϕ_i can be chosen so that they are orthogonal over the sampling points: this means either carefully selecting the sample points given the functions, or vice versa. Then

$$\sum_{k=0}^{N} \phi_i(x_k)\phi_j(x_k)w(x_k) = 0 \quad \text{if} \quad i \neq j$$

and to minimise

$$\sum_{k=1}^{N} (g(x_k) - f(x_k))^2 w(x_k)$$

we compute the coefficients by the formula

$$a_i = \frac{\sum_{k=1}^{N} \phi_i(x_k) f(x_k) w(x_k)}{\sum_{k=1}^{N} \phi_i^2(x_k) w(x_k)}. \tag{1}$$

This case is not of much interest here, since the careful selection of sample points, or the construction of special functions, is impractical.

(ii) No careful selection of sampling points is made and sampling is random. Again we go for the least squares fit, and minimise the weighted sum of the squared deviations at the sample points. Thus

$$\frac{\partial}{\partial a_i}\left[\sum_{k=1}^{N} w(x_k)(f(x_k) - \sum_{j=0}^{M} a_j \phi_j(x_k))^2\right] = 0$$
$$i = 0, 1, \ldots, M.$$

This leads to $M + 1$ equations in $M + 1$ unknown a_i's. These are known as the normal equations. In matrix notation these become:

$$\Phi b = c \tag{2}$$

where

$$\left.\begin{aligned} b_j &= a_{j-1} \quad \text{the unknowns} \\ c_j &= \sum_{k=1}^{N} f(x_k)\phi_{j-1}(x_k)w(x_k) \\ \Phi_{ij} &= \sum_{k=1}^{N} \phi_{i-1}(x_k)\phi_{j-1}(x_k)w(x_k) \end{aligned}\right\} i, j = 1, 2, \ldots, M+1.$$

Solution of these equations will yield the requisite coefficients.

(iii) Rather than solving the above simultaneous equations, we may hope meaningfully to set the coefficients by

$$a_i = \frac{\sum_{k=1}^{N} f(x_k)\phi_i(x_k)w(x_k)}{\sum_{k=1}^{N} \phi_i^2(x_k)w(x_k)}. \qquad (3)$$

While this is the same as formula (1), our viewpoint is radically different. Effectively here we assume that the off-diagonal terms of the matrix 0 in the normal equations (2) are negligible: in fact it can be shown that these do converge to zero as the number of samples is increased with a properly conducted sampling procedure, because of the orthogonality of the functions $\{\phi_i\}$.

(iv) If the ϕ_i's are normal as well as orthogonal, then $(\mu(X)/N)$ times the denominator in (3) tends to unity as N tends to infinity. Thus for orthonormal ϕ_i we could use

$$a_i = \frac{\mu(X)}{N} \sum_{k=1}^{N} f(x_k)\phi_i(x_k)w(x_k). \qquad (4)$$

This converges to the integral which would define a_i if we knew the function f explicitly (see Hammersley and Handscomb, 1964).

Our interest in interpolation for pattern classification is mainly concerned with random sampling, determined by the measure μ. We now note that we can allow the weighting function $w(x)$ to generate a measure on X according to the simple prescription $\omega(Y) = \int_Y w(x)d\mu$ for all $Y \subset X$, when in all our preceding formulae the explicit appearance of $w(x)$ can be removed and sampling can be conducted randomly with respect to the probability distribution $p(Y) = \omega(Y)/\omega(X)$: ω now plays the identical role to that formerly played by μ. These two views are interchangeable and in both cases ω, or the combination of $w(x)$ and μ, give a measure of confidence in and a measure of the expected frequency of occurrence of the various samples. Clearly there is some advantage in sampling with respect to ω, if this can be achieved.

Hence we list formulae (5) and (6) to correspond to formulae (3) and (4).

$$a_i = \frac{\sum_{k=1}^{N} f(x_k)\phi_i(x_k)}{\sum_{k=1}^{N} \phi_i^2(x_k)} \qquad (5)$$

sampling with respect to the measure ω.

$$a_i = \frac{\omega(X)}{N} \sum_{k=1}^{N} f(x_k)\phi_i(x_k) \qquad (6)$$

sampling with respect to the measure ω.

With the preceding understanding of measure and weighting and random sampling, the sums converge to integrals, and as the number of samples is increased all of (2) to (6) converge to the overall least squares fit: thus as the number of functions used in the approximating expansion is increased, with M + 1 less than or equal to N, the approximating function converges to the unknown function 'in the mean'. Given any arbitrarily small number ϵ there exists some finite approximation (that is, an N and an M) such that the weighted sum of the squared deviations (or 'variance') is less than this ϵ. Moreover, if we denote by $T(\epsilon)$ the set of all points such that the deviation or error $|f - g|$ is greater than ϵ, then the measure of this set can be shown to converge to zero with the variance. For threshold decisions of threshold $\frac{1}{2}$, $T(\frac{1}{2})$ is just the set on which errors occur, and thus $\omega(T(\frac{1}{2}))$ is the error-rate. Consequently we conclude that the error-rate can be made arbitrarily small. The condition that the function f must fulfil is that it be square-integrable: for characteristic functions this means that the classification set must be measurable. (See Sz.-Nagy, 1965.)

Note: We can in principle preselect our sample points to fit some mesh (which depends upon the measure and weighting and orthogonal functions), and work systematically through the points. The problem is that we have no method for continuously increasing the number of samples with this approach, though by working from mesh to finer mesh convergence is still assured: however, in proceeding to a finer mesh we must either start from scratch again, or make a very large jump in the number of sample points.

(v) Finally, we see that in formulae (4) and (6), the factors $\mu(X)$ and $\omega(X)$, and the division by N, are not necessary for correct recognition when used in a pattern classifier and can be eliminated (though we must make due allowances for this with threshold decisions). Hence

$$a_i = \sum_{k=1}^{N} f(x_k)\phi_i(x_k)w(x_k) \qquad (7)$$

sampling with respect to measure μ.

$$a_i = \sum_{k=1}^{N} f(x_k)\phi_i(x_k) \qquad (8)$$

sampling with respect to measure ω.

Other error criteria

The least squares error criterion, while the most common, is not the only one possible. Two other criteria will now be discussed.

We could hope to minimise the maximum error. Clearly such an aspiration only has meaning for the approximation by continuous functions of continuous functions; for at jump discontinuities we necessarily must have an error of at least half the jump. So let us suppose that we do not in fact measure a characteristic function directly, but rather a probability distribution $p_A(x)$, the probability that a pattern x will be classified in category A, and that previously we obtain the characteristic function from this by a threshold operation, classifying x in A if $p_A(x)$ is greater than $\frac{1}{2}$. Assume that $p_A(x)$ is continuous and suppose that we have approximated $p_A(x)$ uniformly to within $\epsilon\,(<\frac{1}{2})$ by a function $g(x)$. We know that this is always possible under appropriate conditions, by the Weierstrasse–Stone Approximation Theorem (Sz.-Nagy, 1965). Then we see that in areas of uncertainty (mostly near the boundary) our use of the approximation is also uncertain: for patterns where $p_A(x)$ is within the interval $[\frac{1}{2} - \epsilon, \frac{1}{2} + \epsilon]$, we are prone to errors, but outside this we can guarantee correct classification.

Thus a minimax approximation is very meaningful, for we can constrain the errors to lie in regions where the classification is of doubtful validity anyway: with normal least squares methods we have no control over where the deviations occur. To find the minimax fit, we simply use the n-dimensional version of the technique used in one dimension; that is, we use Chebyshev polynomials in a least squares fit. For us this means using the generalised Chebyshev weighting (on a bounded interval): we can then use any of formulae (2) to (8) to obtain in the limit the minimax fit. However, if the real distribution of patterns is not Chebyshev, then we do not find a minimum of variance when later employing a realisation. Convergence is as previously discussed for least squares.

Minimax and least squares are the principal error criteria of numerical analysis: but here our complete problem goes beyond numerical analysis, for we always follow the approximation by a decision procedure, and the figure of merit for the total system is the error-rate (or a weighted 'cost' version of it): for us the error rate, as assumed previously, is the measure of the set in which errors occur.

Thus we are led to postulate an optimal error criterion for noiseless threshold decisions, for we do not care what the error in the approximation of the characteristic function is, as long as it is less than a half. Let us define a function $q(x)$ on the real line such that $q(x) = 0$ if $|x| < \frac{1}{2}$ and $q(x) = 1$ otherwise. If we are seeking an approximation of the form

$$C_A(x) \simeq g(x) = \sum_{i=0}^{M} a_i \phi_i(x)$$

and we minimise, with respect to the coefficients,

$$\int_X q(C_A(x) - g(x))w(x)d\mu \qquad \text{or} \sum_{k=1}^{N} q(C_A(x_k) - g(x_k))w(x_k)$$

we will obtain optimal values of the coefficients. We can alternatively regard the process as minimising the probability that the error $(C_A(x) - g(x))$ is greater than a half.

In order to be able to use differential techniques to find the minimum, we would like an analytic approximation of $q(x)$. We note that the function sequence $\{(1 - \exp[-(4x^2)^n]) : n = 1, 2, 3, \ldots\}$ converges to $\omega(x)$ and that the first order approximation leads to the least squares fit. For $n = 2$ we minimise the 4th moment of the error distribution and can find an iterative scheme to calculate the coefficients. Similarly for higher order approximations. Such iterative systems, unless they converge very rapidly, are likely to be very costly, and only in exceptional circumstances could they be justified.

We conclude that least squares techniques are generally preferable, for their mathematical elegance and simplicity of realisation. Minimax fits using Chebyshev polynomials gives us a technique for constraining errors to lie near the boundary of the classification sets. These criteria do not give the optimal decision procedure if error-rate is the correct performance criterion, but we have seen that all criteria converge together, and that least-squares is a first order approximation to the minimisation of error-rate.

Orthogonal functions in n dimensions

It is important to consider possible sets of orthogonal functions in R^n. Of most interest here will be the polynomials, where a given subset of R^n and a suitable measure will be associated with a system of orthogonal polynomials. The easiest n-dimensional polynomials to generate are those formed from the product of one-dimensional orthogonal polynomials. If $\{\pi_i(x): i = 0, 1, 2, \ldots\}$ are orthogonal and complete in one dimension, then the polynomial system

$$\{\pi_{i_1}(x_1)\pi_{i_2}(x_2)\ldots\pi_{i_n}(x_n): i_1, i_2, \ldots, i_n = 0, 1, 2, \ldots\}$$

will be orthogonal and complete in n dimensions with the degree of a polynomial being $(i_1 + i_2 + \ldots + i_n)$. We need some system for ordering the polynomials, and it would be most meaningful for the ordering to reflect the 'significance' of the functions. In the above manner we generate n-dimensional Chebyshev polynomials. Alternative to this scheme we can start from any complete function set and extract an orthogonal set by some suitable procedure (cf. Weisfeld, 1959).

There are n^p polynomials of degree p, and thus the number of functions to be considered increases astronomically with increasing p. This is seen to be the major problem in engineering pattern recognition systems: how do we reduce the number of functions (i.e. 'features') to a manageable small quantity? Intuitive insights can help us here though they are not essential, and constitute information additional to that of the samples. For example, for general functions one can use a 'localness' hypothesis to limit one's interest only to functions whose arguments come from restricted locality or 'receptive field' of the image and photomosaic (such as is done with the 'operators' of Uhr and Vossler (1961)).

Polynomials involve algebraic operations and are thus of value for digital realisations. Analogue devices can realise transcendental functions readily and naturally, and one should consider orthogonal function systems appropriate to the means of realisation. In the next section the topic of realisations (that is, the actual designing of a particular system) is taken up in general terms, assuming that the function system has been determined.

Realisations

The preceding sections provide theoretical guidance for the designing of machines which will recognise patterns. There are three approaches we can take: these correspond roughly to what have come to be known as *parametric*, *learning*, and *adaptive* systems.

The first possibility is that we use fixed number of samples and use one of the formulae (2) to (8) to compute the coefficients a_i. We can use any orthogonal function system we choose (bearing in mind that there is an implicit relationship between the procedure whereby we acquire the samples, and the measure and weighting function which define the function system) and can readily perform the analysis using a digital computer employing standard numerical techniques. In effect we have a number of parameters which are to be estimated from the samples.

Subsequently these parameters are used in building the system: the form of the system follows that of the model postulated at the start (Fig. 1).

It may not be desirable to use a fixed number of samples. We see that using equations (3) to (8) the extension to a variable number of samples is trivial, and we can design pattern classifiers with sequential decision properties. We can interpret the process of adding another sample to our set as 'learning' and at any stage the machine can classify as best as it is able with the material that it has learnt so far. A canonical realisation using formula (8) is schematised in **Fig. 2.** Other realisations might make learning local and autonomous at the a_i's.

It is only for formula (2) that the extension to variable N is problematic. Formula (2) involves the solution of the normal equations, and this is essentially a two-stage process: initially the equations are set-up from the samples, and then they are solved. Adding another sample may appear to throw away the preceding solution, but this can be circumvented by using an iterative technique for solving the normal equations. The iterative technique proposed is the Jacobi method of simultaneous displacement. The matrix equation (2) is replaced by the system

$$b^{(n+1)} = DXb^{(n)} + Dc$$

where D is diagonal matrix comprising the elements

$$d_{jj} = \left(\sum_{k=1}^{N} \phi_{j-1}^2(x_k)w(x_k)\right)^{-1} \quad j = 1, 2, \ldots, M + 1$$

and X is the matrix with diagonal elements zero and off-diagonal elements

$$x_{ij} = -\sum_{k=1}^{N} \phi_{i-1}(x_k)\phi_{j-1}(x_k)w(x_k) \quad i,j = 1, 2, \ldots, M + 1.$$

Thus $\Phi = D^{-1} - X$. c is as in (2).

This provides a linear dynamical system whose state yields the latest value of the parameters for the recognition process. Learning a new pattern means up-dating the matrices D and X, and the vector c. For orthogonal ϕ_i this scheme is not very interesting for the elements in X converge to zero; however, where the functions are not orthogonal (or equivalently, where the sampling process does not match the measure) the normal equations still give a least squares fit while the other formulae break down, and this scheme could prove useful in some circumstances.

The previous systems have one serious drawback. As the learning progresses and the number of samples increases, the numbers representing the state of learning or memory, grow without bounds. Some method for preventing this growth needs to be proposed; we must arrange to 'forget' events in the distant past. This then creates a further possible advantage for we have a memory of finite duration which means that changes in classification which occur over a long period are allowed for. In a sense the system is adaptive, and this adaptivity is often a design objective in its own right. However, we accrue the serious disadvantage that our techniques no longer converge: after a very long learning process we do not get arbitrarily close to the best fit parameters, but rather our estimates are distributed about the best parameters (but this is also true for any finite sample in the parametric and learning cases above). One possible forgetting technique is to store explicitly a fixed number of samples and to erase the longest held sample to store a new one. This is clearly likely to use storage inefficiently and a preferable technique would be to 'decay' all numbers at each step by multiplying them by some number just less than one. This is a common method; it is the discrete case of the exponential decay and constitutes a low pass filter. Other strategies could be employed, but these are the simplest conceptually.

In developing the theory we occasionally had recourse to intuitive notions such as 'learning', and 'features': we should now like to make explicit the way we interpret

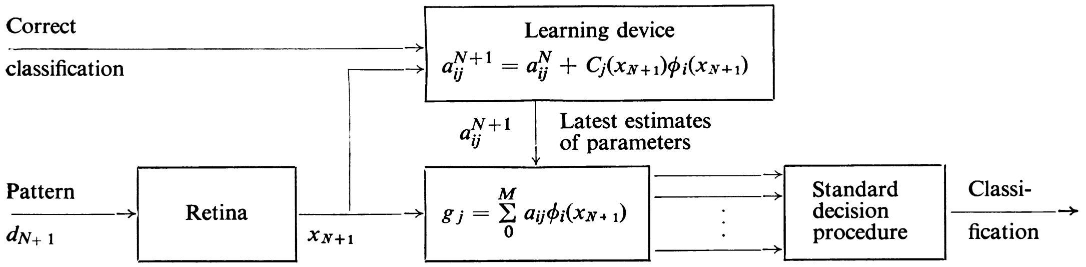
Correct
classification
Learning device
$a_{ij}^{N+1} = a_{ij}^{N} + C_j(x_{N+1})\phi_i(x_{N+1})$
a_{ij}^{N+1}
Latest estimates of parameters
Pattern
d_{N+1}
Retina
x_{N+1}
$g_j = \sum_{0}^{M} a_{ij}\phi_i(x_{N+1})$
Standard decision procedure
Classi-
fication

Fig. 2. Learning system realising formula 8

the theory. We see the process of 'generalisation' to be synonymous with interpolation and extrapolation, 'learning' as the estimation of parameters or 'weights' in a functional approximation used for interpolation. The individual functions ϕ_i in the linear expansion are 'features', 'properties', or 'characteristics', while the computation of the particular value of the function ϕ_i for a pattern is 'preprocessing', 'feature extraction' or 'property filtering'. The final threshold operation, or maximum selection, is the (standard) 'decision procedure'.

With these interpretations we see that many existing approaches are subsumed by the theory. For example, Bledsoe and Browning (1959) '*n*-tuples' are features in the form of polynomials of degree *n*, when we generalise their binary formulation of the problem to the case of continuous variables; while template matching uses only polynomials of degree zero and one (see Nilsson, 1965). The learning techniques developed here are similar to many existing approaches, with the difference that the emphasis is placed not upon the correct classification of the given samples (as in Nilsson, 1965, and others), but rather on the ultimate performance of the final machine for all possible patterns.

With an infinity of possible patterns, the theory accepts that there must be some errors in practice; but the sources of these errors (namely, the *n* retinal abstractions, the finite set of functions ϕ_i, the finite set of samples, and the usual physical or round-off noise) are all controllable and the magnitudes of each contribution to the error is reducible arbitrarily. Thus we deduce that, guided by the theory, we can attain a design for any classification problem compatible with some prespecified criterion of performance.

Conclusions

We have seen how learning in pattern classification can be viewed as a problem of interpolation, and that viewed as such, there are many techniques available in numerical analysis which can be applied to pattern classification. Realisations can then be obtained in a systematic manner, and in such a way that for almost any classification problem, by increasing system complexity any performance criterion can be met. This approach has also given an overview of pattern recognition and many seemingly different approaches have been shown to be closely related. Thus we are supplied with a general mathematical theory of learning and generalisation in pattern classification.

The basic model is extremely simple, and it is hoped that the ideas will readily extend to sequential patterns in general, and to more advanced 'intelligent' functions.

Acknowledgements

I should like to record my indebtedness to the Science Research Council who have supported me during the period in which the research here reported was carried out.

I should also like to extend my thanks to Dr. Chakravarti and Dr. W. K. Taylor for invaluable criticism during the formulation of this paper.

References

Bledsoe, W. W., and Browning, I. (1959). Pattern Recognition and Reading by Machine, *Proc. Eastern Joint Computer Conf.*, 1959. [Reprinted in Uhr (1966).]

Brick, D. B., and Owen, J. L. (1964). A Mathematical Approach to Pattern Recognition and Self-Organisation, in *Computers and Information Sciences*, Editors, Tou and Wilcox, Spartan Books, p. 139.

Clenshaw, C. W., and Hayes, J. C. (1965). Curve and Surface fitting, *Journal of Institute of Mathematics and its Applications*, Vol. 1, p. 164.

Gabor, D. (1955). Optical Transmission, in *Information Theory*, Editor Cherry, Butterworths, p. 26.

Hammersley, J. M., and Handscomb, D. C. (1964). *Monte Carlo Methods*, Methuen, London.

Hamming, R. W. (1962). *Numerical Methods for Scientists and Engineers*, McGraw-Hill, New York; Kōgakusha, Japan.

Lanczos, C. (1957). *Applied Analysis*, Pitman, London.

Nilsson, N. J. (1965). *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, McGraw-Hill, New York.

Sz.-Nagy, B. (1965). *Introduction to Real Functions and Orthogonal Expansions*, Oxford University Press.

Uhr, L. (1966). *Pattern Recognition*, J. Wiley, New York.

Uhr, L., and Vossler, C. (1961). A Pattern-recognition Program that Generates, Evaluates, and Adjusts its own Operators, *Proc. Western Joint Computer Conf.*, 1961, p. 555. [Reprinted in Uhr (1966).]

Weisfeld, M. (1959). Orthogonal Polynomials in Several Variables, *Numerische Mathematik*, Bd. 1, p. 38.