N70-26171

NASA TECHNICAL NOTE

NASA TN D-5790

NASA TN D-5790

INTRODUCTION TO LEARNING MACHINES

by Jack G. Sheppard
Manned Spacecraft Center
Houston, Texas 77058

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MAY 1970

1. REPORT NO. NASA TN D-5790	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.

4. TITLE AND SUBTITLE INTRODUCTION TO LEARNING MACHINES	5. REPORT DATE May 1970
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Jack G. Sheppard, MSC	8. PERFORMING ORGANIZATION REPORT NO. S-224
9. PERFORMING ORGANIZATION NAME AND ADDRESS Manned Spacecraft Center Houston, Texas 77058	10. WORK UNIT NO. 914-50-50-16-72
	11. CONTRACT OR GRANT NO.
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D.C. 20546	13. REPORT TYPE AND PERIOD COVERED Technical Note
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

The concept of pattern recognition, or machine learning, is introduced, and the two basic pattern-recognition techniques, parametric and nonparametric, are discussed. The basic theory and the operation of each technique are presented. The error-correction training technique also is discussed. A computer program that illustrates the error-correction procedure and data that indicate the performance to be expected with this technique on a class of curve-fitting problems are presented.

17. KEY WORDS (SUPPLIED BY AUTHOR) · Learning Machines · Pattern Recognition	18. DISTRIBUTION STATEMENT Unclassified - Unlimited

19. SECURITY CLASSIFICATION (THIS REPORT) None	20. SECURITY CLASSIFICATION (THIS PAGE) None	21. NO. OF PAGES 27	22. PRICE * $3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

CONTENTS

Section Page

SUMMARY 1

INTRODUCTION 1

SYMBOLS 2

HISTORY 4

PATTERNS 4

Selecting Pattern Components 5

Classifying Patterns 5

DISCRIMINANT FUNCTIONS 5

LINEAR FUNCTIONS 6

QUADRIC FUNCTIONS 7

THE ϕ FUNCTIONS 8

NONPARAMETRIC METHODS 9

PARAMETRIC METHODS 11

AN ERROR-CORRECTION EXAMPLE 14

CONCLUSIONS 18

REFERENCES 20

APPENDIX — AN nth-DEGREE ϕ MACHINE 21

FIGURES

Figure Page

1 Linear machine 6

2 Linear dichotomizer 7

3 Quadric ϕ machine 9

4 Error-correction training example 10

5 An nth-degree ϕ machine program 16

6 Linear (wide) curve-fit example 17

7 Example convergence times 17

8 Linear (narrow) curve-fit example 17

9 Quadratic curve-fit example 17

10 Cubic curve-fit example 18

11 Quartic curve-fit example (did not converge) 18

12 An nth-degree ϕ machine block diagram 22

INTRODUCTION TO LEARNING MACHINES

By Jack G. Sheppard
Manned Spacecraft Center

SUMMARY

Learning machines can perform routine intellectual tasks that are normally the exclusive domain of human beings. Learning machines attempt to recognize and to classify patterns and frequently are called adaptive pattern recognizers. Pattern recognition is presently an active field. Several hundred technical reports and articles on the subject are published annually on such diverse subjects as mechanisms of human vision, military target recognition, adaptive-control systems, and optimum communication receivers.

Learning machines, which operate on patterns, use a priori information or some inherent pattern-class characteristic to decide the category in which a pattern belongs. The key characteristic of learning machines is the ability to recognize general relationships from a limited set of observations. Two principal learning-machine structures are used. One structure assumes the separability of all categories and derives exact decision procedures. The other structure uses a probabilistic approach and attempts to make optimum decisions, always recognizing the possibility of error.

The majority of learning-machine research has emphasized machine structure; application to specific problems has been limited. The structural theory is now developed sufficiently to provide the tools for investigators to apply in various individual fields.

INTRODUCTION

For many years, scientists have been intrigued by the possibility of machines that can replace people in routine intellectual activities. A class of machines that are called learning machines or adaptive pattern classifiers now can perform some routine tasks previously performed only by human beings. These tasks include weather prediction, handwriting analysis, speech analysis, target recognition, and medical diagnosis. In many tasks, such as weather prediction, machines are superior to human beings in both speed and accuracy. In other tasks, such as speech recognition, the problems are so complex that the results are incomplete.

Many intellectual processes and methods of mechanization have been investigated, but most of these approaches were not feasible until the advent of high-speed digital computers. Consequently, most of the significant work on learning machines has been done within the last 15 years. As late as 1960, exact matching and correlation with stored references were the most sophisticated techniques in use. Since 1960, the concept of representing inputs as vectors in an n-dimensional space has gained acceptance. Based on this concept, the learning-machine theory has progressed rapidly. Presented in this technical note are a discussion of the two basic approaches to pattern recognition, the theory underlying each approach, and a computer-program-implementation example of one of the important learning procedures of the nonparametric technique.

SYMBOLS

A	square matrix
a	element of A
B	column vector
b	element of B
C	set of all pattern classification categories
C_i	ith classification category
c	element of C
c'	element of C
D	scalar
F	vector composed of functions of X
f	element of F
$g_i(X)$	ith discriminant function of X
$g'_i(X)$	the logarithm of $g_i(X)$
$L_x(i)$	conditional average loss
$L'_x(i)$	modified conditional average loss
ln	natural logarithm
M_i	mean vector of the ith category
$p(j/x)$	density on j given that x has occurred

$p(x)$	density on X
R	n-dimensional vector space defined by the set of all X; R is called the pattern space
R_i	set of all patterns that should be classified into the category C_i
$R \times C$	Cartesian product of the sets R and C
r	element of R
W	set of $n + 1$ numbers collectively called a weight vector
W'	adjusted weight vector
w_i	ith component of a weight vector
X	set of n measurements $(x_1, x_2, \ldots, x_n)$ collectively called a pattern
X^t	row vector formed by the transposition of the column vector X
$X \epsilon R_i$	X is an element of the set R_i
x_i	ith component of X
Y	augmented pattern vector
δ_{ij}	Dirac Delta Function
$\lambda(i/j)$	loss incurred in selecting i when j actually has occurred
Σ	covariance matrix of a multivariate Gaussian density
Σ^{-1}	inverse of the matrix Σ
σ	element of Σ
σ_i	ith component of Σ

Subscripts:

k	number of categories into which patterns are to be classified
M	dimension of ϕ space

Superscript:

t a matrix transposed

HISTORY

From earliest times, man has been interested in the mechanics of intellectual processes (ref. 1). Approximately 320 B.C., Aristotle formulated his logic with the purpose of systematizing rules of thought. Francis Bacon (1561 to 1626) sought to establish rules for the systematic acquisition of knowledge. Thomas Hobbes (1588 to 1679) was concerned with thought processes; that is, the way the mind progresses in an orderly and directed way from one thought to the next thought. John Locke (1632 to 1704) dealt with perception, which he considered to be "the inlet of all materials of knowledge." Locke was one of the first to document the relationship between repetitive perception and memory enhancement (ref. 2). George Boole exposed the intellectual process of generalization through which an observer draws conclusions that are greater and more comprehensive than the observations on which the conclusions are based (ref. 3). This concept, generalization from limited observation, distinguishes a learning machine from an ordinary process controller or computer.

The following men were pioneers in the development of learning machines. Rosenblatt (ref. 4) contributed the Perceptron, a two-dimensional array of optical sensors that is the basis of most image processors. Highleyman (ref. 5) contributed significantly to linear-machine theory. Braverman (ref. 6) applied the Bayesian decision theory, which led to the current work with parametric machines, and extended his work in collaboration with Abramson (refs. 7 and 8). Sebestyen (ref. 9) demonstrated the value of transformation on the pattern space for feature enhancement. Fralick (ref. 10) has shown that optimum solutions in a machine of fixed size can be obtained without a teacher.

PATTERNS

Learning machines operate on patterns. (Much of this discussion is based on the work of N. J. Nilsson, ref. 11.) A pattern is a set of measurements $X = (x_1, x_2, \ldots, x_n)$ that represents some phenomenon of interest. Such a set can be considered as a vector in an n-dimensional vector space. If the vector space accurately represents the phenomenon of interest, each vector in the space represents some state of the source phenomenon. If rules can be derived that allow nonambiguous mappings from the space to a set of outcomes, a machine can accomplish these mappings, and an automatic pattern classifier can be derived. Thus, learning machines involve two basic problems. One problem is the selection of the components of the pattern so that the pattern space accurately represents the phenomenon of interest. The other problem is the determination of the rules that allow nonambiguous mappings from the pattern space to a set of outcomes or classifications.

Selecting Pattern Components

The process of selecting the components of a pattern (ref. 12) involves the customary conflict of interest; that is, the balance between measuring enough things and the expense of making the measurements. The cost of processing, which is proportional to the number of components of the pattern, further augments this expense. A compromise must be derived that does not sacrifice critical information and yet is economical. The form of the pattern is another concern. To simplify the classification problem, the selected components should emphasize the differences among the patterns of the categories. Sometimes, as in target recognition, a choice is not available. In these cases, transformations on the pattern space may emphasize the correct characteristics and lead to simplified processing. This process of selecting and transforming the correct components of the pattern is intuitive and is not amenable to general treatment. Therefore, this area was not of prime interest to early investigators. However, the process is essential to any use of pattern-recognition techniques and will be a principal area of concern to anyone who attempts to use a learning machine on a particular problem.

Classifying Patterns

After the components of the pattern are selected, a phase that is more amenable to general treatment is reached. Two methods are most common. One method is based on the theory of finite dimensional vector spaces, and the other method is based on statistical decision theory. In the first case (nonparametric), an attempt is made to construct decision regions in pattern space by the use of hyperplanes, hyperspheres, and other surfaces as boundaries between regions. The problem is the selection of the correct surface types and the correct locations for these surfaces. In the other case (parametric), some statistical distribution is assumed to be a function of parameters, and optimal decision rules are used to make classifications. In either case, information is usually inadequate to complete the classifier, and a training technique is used to finish the job. These training techniques usually include presenting a series of previously classified patterns to the machine and correcting the response until the machine "learns" to make correct classifications.

DISCRIMINANT FUNCTIONS

In both parametric and nonparametric cases, the concept of function is useful in the general problem of mappings from the pattern space. The following definition will provide the basis for this discussion of discriminant functions.

> A function from a set R to a set C is a set g of ordered pairs in $R \times C$ with the property that if (r, c) and (r, c') are elements of g, then $(c = c')$.

Thus, given a pattern space R and a set C of categories into which the patterns are to be classified, a function maps each element of the pattern space into one and only one of the classification categories. The problem devolves into selecting the proper function so that the outcomes have some meaning.

Assume that each point in R belongs to one of the categories $C_1, C_2, \ldots, C_k$. Then R can be divided into the subsets $R_1, R_2, \ldots, R_k$ according to the category of classification. Consider a set of scalar functions of the vector $X \epsilon R$, $[g_1(X), g_2(X), \ldots, g_k(X)]$ chosen so that for all $X \epsilon R_i$

$$g_i(X) > g_j(X) \quad \text{for} \quad i, j = 1, 2, \ldots, k; \; i \neq j \tag{1}$$

Such functions are called discriminant functions. If such a set of functions can be found, and the outputs feed a maximum detector, the set will classify properly any vector $X \epsilon R$ (fig. 1).

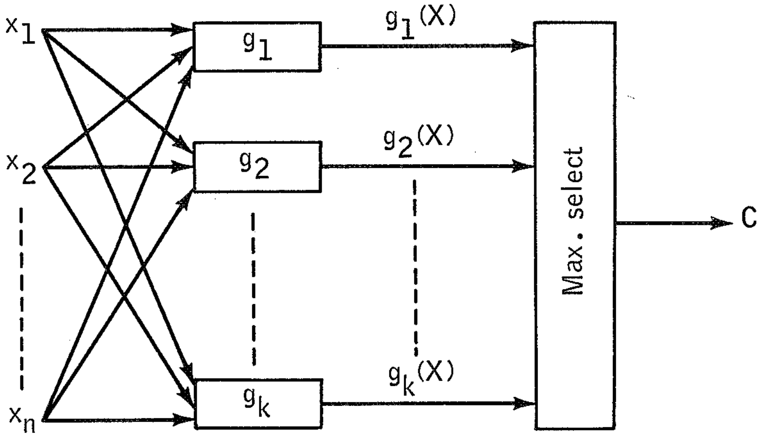


Figure 1. - Linear machine.

LINEAR FUNCTIONS

The simplest form of a discriminant function is

$$g_i(X) = w_1x_1 + w_2x_2 + \ldots + w_nx_n + w_{n+1} \tag{2}$$

This form is linear in the components of X and is called a linear discriminant function. Linear discriminant functions describe hyperplanes in n-space. Such functions are especially useful when the pattern space is to be dichotomized (fig. 2). Any pattern space that can be divided correctly by linear functions is called linearly separable.

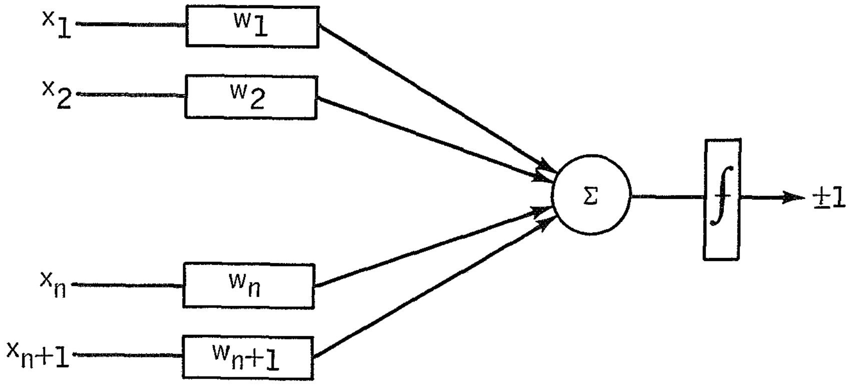


Figure 2. - Linear dichotomizer.

QUADRIC FUNCTIONS

Consider the equations

$$g(X) = X^tAX + X^tB + D \tag{3}$$

$$g(X) = [x_1 x_2]\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [x_1 x_2]\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + D \tag{4}$$

$$g(X) = a_{11}x^2_1 + (a_{12} + a_{21})x_1x_2 + a_{22}x^2_2 + b_1x_1 + b_2x_2 + D \tag{5}$$

This "second-degree equation" specializes to the ellipse and hyperbola. In generalized n-space, equation (3) is called a quadric discriminant function. Depending on whether the associated quadratic form (X^tAX) is positive definite, positive semidefinite, or nondefinite, a quadric function specializes to a hyperellipsoid, a hyperellipsoidal cylinder, or a hyperhyperboloid. The orientations of these surfaces are controlled by the eigenvectors of A. Because of the added flexibility of form, the quadric function is a more powerful tool than the hyperplane.

THE ϕ FUNCTIONS

In expanded form, equation (3) is

$$g(X) = \sum_{i=1}^{n} w_{ii} x_i^2 + \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} w_{ik} x_i x_k + \sum_{i=1}^{n} w_i + w_{n+1} \tag{6}$$

Let F be a vector composed of functions of X

$$\left(F = f_1, f_2, \ldots, f_M\right) \tag{7}$$

Let the first n of these functions be

$$\left(x_1^2, x_2^2, \ldots, x_n^2, \ldots\right) \tag{8}$$

the second n(n - 1)/2 functions be

$$\left(\ldots, x_1 x_2, x_1 x_3, \ldots, x_{n-1} x_n, \ldots\right) \tag{9}$$

and the last n functions be

$$\left(\ldots, x_1, x_2, \ldots, x_n\right) \tag{10}$$

Equations (7) to (10) are the basis for figure 3.

Note that although the quadric function is nonlinear in X, the implementation shown in figure 3 is linear in F. A one-to-one transformation has been made from the pattern space to a function space to provide the capability to use the linear-machine implementation of the preceding discussion. This technique is valuable in the machine-training phase and can be generalized to higher order, more powerful functions. Any discriminant function of the preceding form, with the pattern vector applied to a processor followed by a linearly weighted summer, is a ϕ function. These ϕ functions are powerful in terms of the types of surfaces implemented and in view of the fact that the ϕ functions are trained as linear machines.

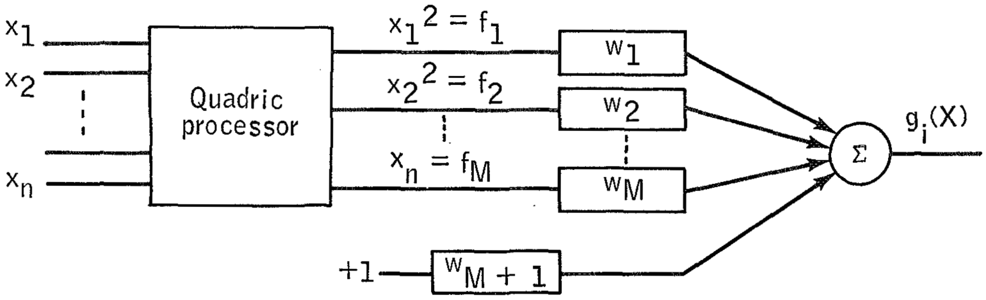


Figure 3. - Quadric ϕ machine.

NONPARAMETRIC METHODS

In nonparametric methods, the members of different catagories generally are assumed to be separable from each other by some forms of surfaces, and attempts are made to identify those surfaces. Usually, some form of ϕ machine, which is trained as a linear machine, is used. The error-correction method of training a linear dichotomizer, which is described in this section, can be generalized to the training of n-category ϕ machines.

The discriminant function of a linear dichotomizer is

$$\begin{aligned} g(X) &= w_1x_1 + w_2x_2 + \ldots + w_nx_n + w_{n+1} \\ &= \left(w_1, w_2, \ldots, w_n, w_{n+1}\right)\cdot\left(x_1, x_2, \ldots, x_n, 1\right) \qquad (11) \\ &= W \cdot Y \end{aligned}$$

where $W = (w_1, w_2, \ldots, w_n, w_{n+1})$ is called the weight vector and $Y = (x_1, x_2, \ldots, x_n, 1)$ is called an augmented pattern vector. The set of all W forms a vector space called the weight space. The set of all W such that

$$W \cdot Y = 0 \qquad (12)$$

is a hyperplane through $(0, 0, \ldots, 0)$ called the pattern hyperplane of Y. Consider a simple example in a two-dimensional weight space with three patterns in a training set. (Patterns are then real numbers.)

Let

$$Y_1 = (1, 1), Y_2 = (-5, 1), Y_3 = \left(\frac{1}{2}, 1\right), W = \left(w_1, w_2\right) \qquad (13)$$

then, the following three equations describe the pattern hyperplanes associated with the three patterns in this training set (fig. 4).

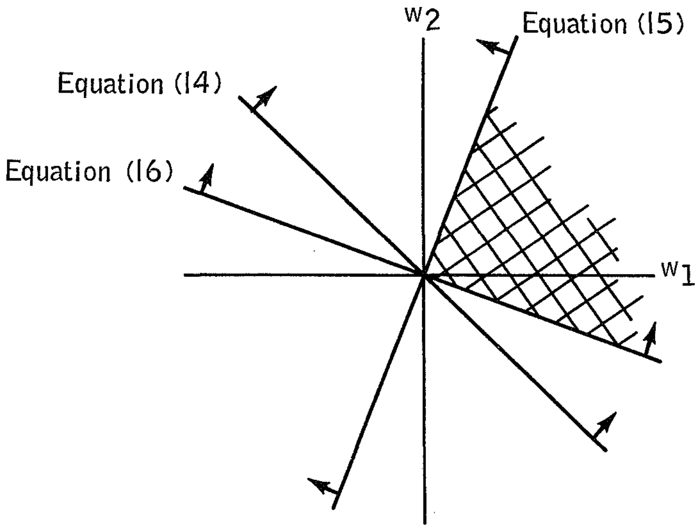


Figure 4. - Error-correction training example.

$$W \cdot Y_1 = w_1 + w_2 = 0 \quad \text{or} \quad w_1 = -w_2 \tag{14}$$

$$W \cdot Y_2 = -5w_1 + w_2 = 0 \quad \text{or} \quad w_2 = 5w_1 \tag{15}$$

$$W \cdot Y_3 = \frac{1}{2} w_1 + w_2 = 0 \quad \text{or} \quad w_1 = -2w_2 \tag{16}$$

Consider the hyperplane in equation (14) (fig. 4) and any W such that $w_2 > -w_1$. Then $W \cdot Y_1 > 0$ and an arrow on the hyperplane indicates the positive side. Similarly, the hyperplanes in equations (15) and (16) (fig. 4) have positive sides; that is, any weight vector on that side will give a positive response when coupled with that pattern vector. Suppose that Y_1 and Y_3 belong to category 1 and Y_2 belongs to category 2. Then, a proper discriminant function should be

$$g(X) = W \cdot Y > 0 \quad \text{for} \quad x_1, x_3 = W \cdot Y < 0 \quad \text{for} \quad x_2 \tag{17}$$

Any weight vector in the crosshatched section of figure 4 satisfies these relationships. Such a section is called the solution region. In error-correction training, the procedure starts with an arbitrary weight vector that is adjusted until a vector in the solution region is found.

Start with the weight vector (-1.5, 1) and examine the discriminant function output for Y_3, which is correctly positive. No adjustment is required. For Y_1, the response is incorrectly negative, and adjustment is required. In figure 4, the most advantageous direction to move W is perpendicular to equation (14). This movement is accomplished by taking

$$W' = W + Y_1 \tag{18}$$

which produces a weight vector that gives a correctly positive answer for Y_1. By the use of W', Y_2 produces an incorrectly positive number. Then, take

$$W'' = W' - Y_2 = (-0.5, 2) - (-5, 1) = (4.5, 1) \tag{19}$$

This weight vector is in the solution region and will give correct answers to all the training patterns. The machine is trained.

Thus, the nonparametric training method involves successively testing each vector in a training set and making corrections until the machine ceases to make errors. For separable sets, this procedure converges to a solution after only a finite number of operations. However, in the correction of one error, the weight vector may be moved into a region causing an error from another pattern vector. Thus, the training process may terminate only after many cycles through the training set. In cases in which the sets are not separable, this training process will never terminate, and other techniques, such as the nearest neighbor approach (ref. 11), must be used. The appendix contains a computer implementation of this technique for an nth-degree ϕ machine.

PARAMETRIC METHODS

Parametric methods depend on a field of mathematical statistics that is called decision theory (ref. 13). Decision theory involves making optimum decisions in the absence of deterministic conditions. For instance, two sets may be nonseparable but distributed in ways that allow a best choice between the two.

The idea of loss is inherent in the concept of best choice. That is, what is lost because of an incorrect decision? A loss function assigns values to the losses incurred in a variety of error situations. Many such loss functions exist, but only the symmetrical loss function will be considered.

$$\lambda(i/j) = 1 - \delta_{ij} \qquad \delta_{ij} = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases} \tag{20}$$

where $\lambda(i/j)$ is the loss caused by deciding on i, when actually j has occurred. Thus, a loss of 1 occurs for any error, and a loss of 0 occurs for a correct decision. Using this loss function, an optimum classifier will be derived for pattern sets distributed in a Gaussian manner about the means.

The conditional average loss in classifying a vector X into category i can be written as

$$L_x(i) = \sum_{j=1}^{k} \lambda(i/j)p(j/x) \tag{21}$$

By Bayes rule

$$p(j/x) = \frac{p(x/j)p(j)}{p(x)} \tag{22}$$

yielding

$$L_x(i) = \frac{1}{p(x)} \sum_{j=1}^{k} \lambda(i/j)p(x/j)p(j) \tag{23}$$

where $L_x(i)$ is to be minimized with respect to i. Because p(x) is not a function of i, p(x) may be deleted from the equation.

$$L'_x(i) = \sum_{j=1}^{k} \lambda(i/j)p(x/j)p(j) \tag{24}$$

Substitution of equation (20) into equation (24) yields

$$L'_x(i) = \sum_{j=1}^{k} \left(1 - \delta_{ij}\right) p(x/j)p(j)$$

$$= \sum_{j=1}^{k} p(x/j)p(j) = p(x) - p(x/i)p(i) \tag{25}$$

This expression is minimized with respect to i by maximizing $p(x/i)p(i)$.

Thus, the optimum discriminant function is

$$g_i(X) = p(x/i)p(i) \tag{26}$$

Because a logarithmic form is more useful for the following discussion, use is made of the monotonic property of the logarithm to give

$$g'_i(X) = \ln g_i(X) = \ln p(x/i) + \ln p(i) \tag{27}$$

For a multivariate Gaussian distribution

$$p(x/i) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma_i\right|^{1/2}} \exp\left[-\frac{1}{2}\left(X - M_i\right)^t \Sigma_i^{-1}\left(X - M_i\right)\right] \tag{28}$$

where X is the pattern vector in column form, M is the mean vector in column form, and

$$\sigma = \begin{bmatrix} \sigma_{ii} & \cdots & \sigma_{in} \\ & & \\ \sigma_{ni} & \cdots & \sigma_{nn} \end{bmatrix} \tag{29}$$

called the covariance matrix. Then

$$g_i'(X) = -\frac{1}{2}\ln 2\pi - \frac{1}{2}\ln\left|\Sigma_i\right| - \frac{1}{2}\left[\left(X - M_i\right)^t \Sigma_i^{-1}\left(X - M_i\right)\right] + \ln p_i \tag{30}$$

which simplifies to

$$g_i'(X) = a_i - \frac{1}{2}\left[\left(X - M_i\right)^t \Sigma_i^{-1}\left(X - M_i\right)\right] \tag{31}$$

with

$$a_i = \ln p_i - \frac{1}{2}\ln\left|\Sigma_i\right| \tag{32}$$

which is recognized as a form of the quadric function

$$g_i(X) = X^tAX + X^tB + D \tag{33}$$

Thus, quadric discriminant functions are optimum for Gaussian patterns. Obviously, a different loss function or distribution would produce a different result, but the Gaussian distribution approximates many natural phenomena and is useful. The mean vector and the covariance matrix are estimated from a training set in any of a variety of ways.

AN ERROR-CORRECTION EXAMPLE

The ϕ machines are similar to polynomial curve-fitting programs in that without a priori knowledge of the exact order of the required surface, the machines attempt to fit surfaces to sets of data. An example of an nth-degree ϕ machine, which is discussed in this section, is a computer implementation that fits an nth-degree polynomial between separable sets of data in two-dimensional Cartesian coordinates. That is, the discriminant function is

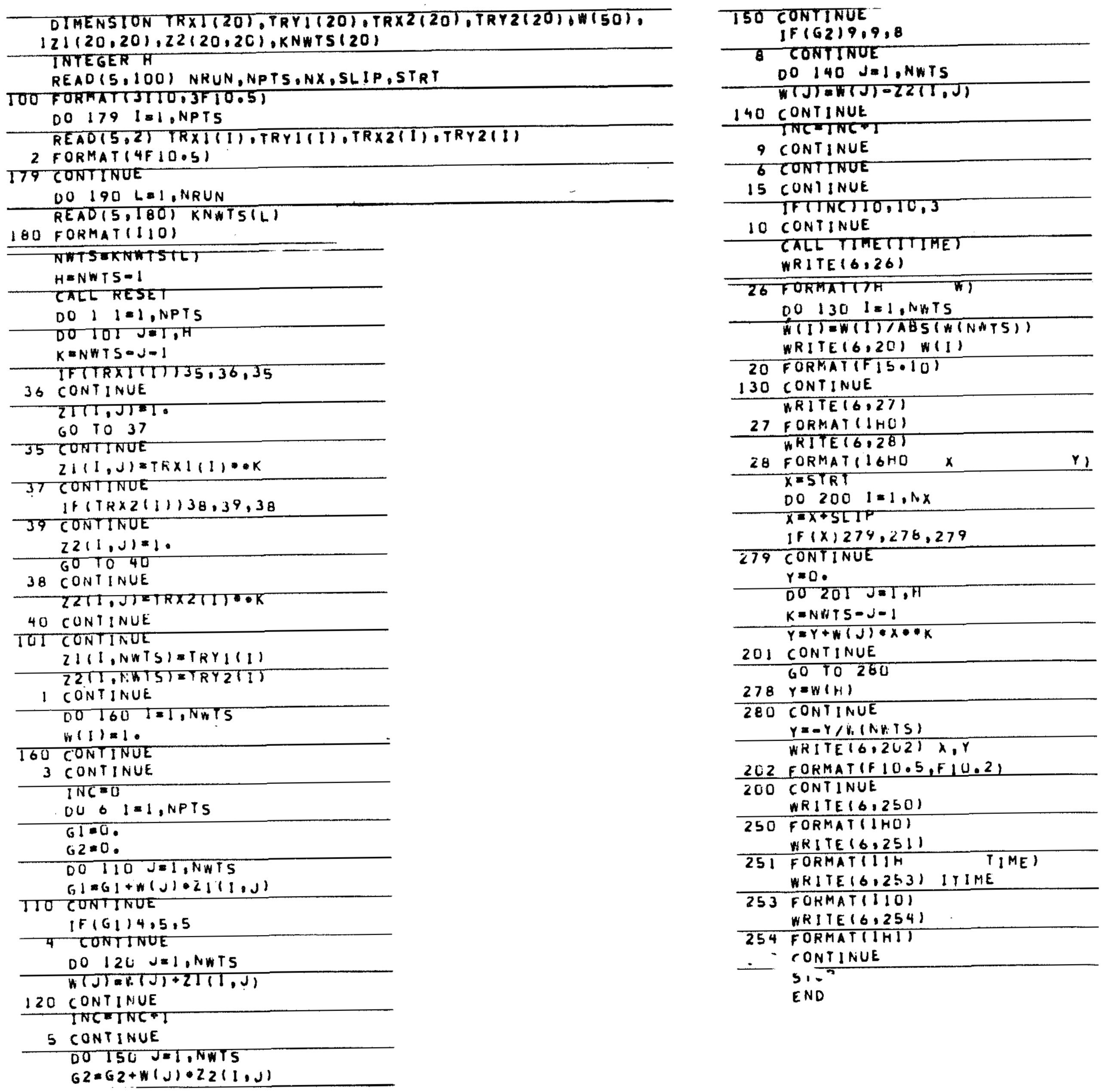
$$g(x, y) = w_n x^{n-2} + w_{n-1} x^{n-3} + \ldots + w_3 x + w_2 + w_1 y \tag{34}$$

Several sets of data were used in an investigation into the convergence rates and processing times to be expected with this class of problem.

This example makes use of the error-correction method described previously. An attempt was made to remove the time-consuming iterative processes from the actual training loop. For instance, in fitting a fifth-degree polynomial, the fifth, fourth, third, and second powers of the abscissa value are used repeatedly in the training loop. To save time, these powers of the abscissa are calculated and stored as new variables in the program before the error-correction iterations begin. Thus, only those operations involved in the calculation of the discriminant value and the adjustment of the weight vector are retained as iterative processes. The actual computer program is shown in figure 5, and a discussion of the mechanics is in the appendix.

Five sets of data and the machine (Univac 1108) convergence times for various orders of discriminant functions are shown in figures 6 to 11. First set of data (fig. 6) involved two straight lines separated by a relatively large space. Convergence times for the first- to fourth-degree curves are shown in figure 7. The second set of data (fig. 8) also involved two straight lines, but these lines had considerably reduced spacing. Processing times increased slightly, and the fourth-degree polynomial did not converge in 10 minutes of machine time. Similarly, parabolic (fig. 9), cubic (fig. 10), and quartic (fig. 11) sets of data were tried with relatively narrow separations. In no case did the fourth-degree discriminant function converge within 10 minutes.

In figure 7, the processing time is plotted on a logarithmic scale. With this scale, the processing times form a basically straight line with respect to the order of the fitted polynomial. Because the processing time increases exponentially with order, an increasingly high price is paid for additional dimensions or components in the training vector. The actual shapes of the two sets of data are relatively insignificant; that is, if the form factors remain approximately the same, the program will fit a third-degree equation to sets of cubic data almost as well as to sets of linear data. Because of the significant amount of processing time that is required, this particular technique would not be an effective tool for fitting high-order polynomials. Other techniques that use pattern recognition to fit curves have been developed (ref. 14). In general, these techniques allow some error and use elaborate gerrymandering to decrease processing time to a minimum.



```
      DIMENSION TRX1(20),TRY1(20),TRX2(20),TRY2(20),W(50),
     1Z1(20,20),Z2(20,20),KNWTS(20)
      INTEGER H
      READ(5,100) NRUN,NPTS,NX,SLIP,STRT
  100 FORMAT(3I10,3F10.5)
      DO 179 I=1,NPTS
      READ(5,2) TRX1(I),TRY1(I),TRX2(I),TRY2(I)
    2 FORMAT(4F10.5)
  179 CONTINUE
      DO 190 L=1,NRUN
      READ(5,180) KNWTS(L)
  180 FORMAT(I10)
      NWTS=KNWTS(L)
      H=NWTS-1
      CALL RESET
      DO 1 I=1,NPTS
      DO 101 J=1,H
      K=NWTS-J-1
      IF(TRX1(I))35,36,35
   36 CONTINUE
      Z1(I,J)=1.
      GO TO 37
   35 CONTINUE
      Z1(I,J)=TRX1(I)**K
   37 CONTINUE
      IF(TRX2(I))38,39,38
   39 CONTINUE
      Z2(I,J)=1.
      GO TO 40
   38 CONTINUE
      Z2(I,J)=TRX2(I)**K
   40 CONTINUE
  101 CONTINUE
      Z1(I,NWTS)=TRY1(I)
      Z2(I,NWTS)=TRY2(I)
    1 CONTINUE
      DO 160 I=1,NWTS
      W(I)=1.
  160 CONTINUE
    3 CONTINUE
      INC=0
      DO 6 I=1,NPTS
      G1=0.
      G2=0.
      DO 110 J=1,NWTS
      G1=G1+W(J)*Z1(I,J)
  110 CONTINUE
      IF(G1)4,5,5
    4  CONTINUE
      DO 120 J=1,NWTS
      W(J)=W(J)+Z1(I,J)
  120 CONTINUE
      INC=INC+1
    5 CONTINUE
      DO 150 J=1,NWTS
      G2=G2+W(J)*Z2(I,J)
  150 CONTINUE
      IF(G2)9,9,8
    8  CONTINUE
      DO 140 J=1,NWTS
      W(J)=W(J)-Z2(I,J)
  140 CONTINUE
      INC=INC+1
    9 CONTINUE
    6 CONTINUE
   15 CONTINUE
      IF(INC)10,10,3
   10 CONTINUE
      CALL TIME(ITIME)
      WRITE(6,26)
   26 FORMAT(7H      W)
      DO 130 I=1,NWTS
      W(I)=W(I)/ABS(W(NWTS))
      WRITE(6,20) W(I)
   20 FORMAT(F15.10)
  130 CONTINUE
      WRITE(6,27)
   27 FORMAT(1H0)
      WRITE(6,28)
   28 FORMAT(16H0    X          Y)
      X=STRT
      DO 200 I=1,NX
      X=X+SLIP
      IF(X)279,278,279
  279 CONTINUE
      Y=0.
      DO 201 J=1,H
      K=NWTS-J-1
      Y=Y+W(J)*X**K
  201 CONTINUE
      GO TO 280
  278 Y=W(H)
  280 CONTINUE
      Y=-Y/W(NWTS)
      WRITE(6,202) X,Y
  202 FORMAT(F10.5,F10.2)
  200 CONTINUE
      WRITE(6,250)
  250 FORMAT(1H0)
      WRITE(6,251)
  251 FORMAT(11H       TIME)
      WRITE(6,253) ITIME
  253 FORMAT(I10)
      WRITE(6,254)
  254 FORMAT(1H1)
      CONTINUE
      STOP
      END
```

Figure 5. - An nth-degree ϕ machine program.

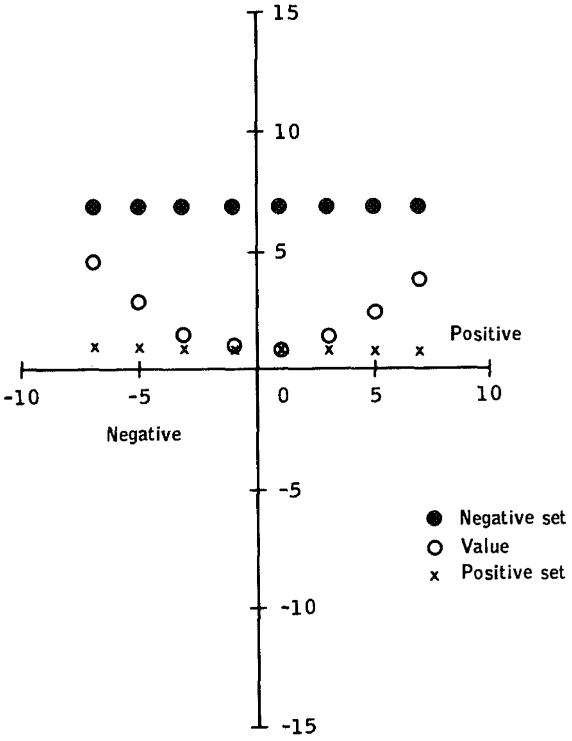


Figure 6. - Linear (wide) curve-fit example.

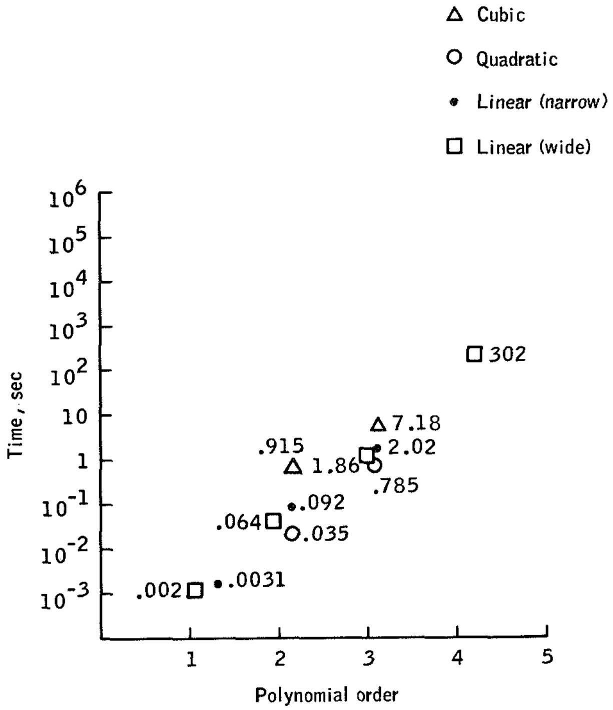


Figure 7. - Example convergence times.

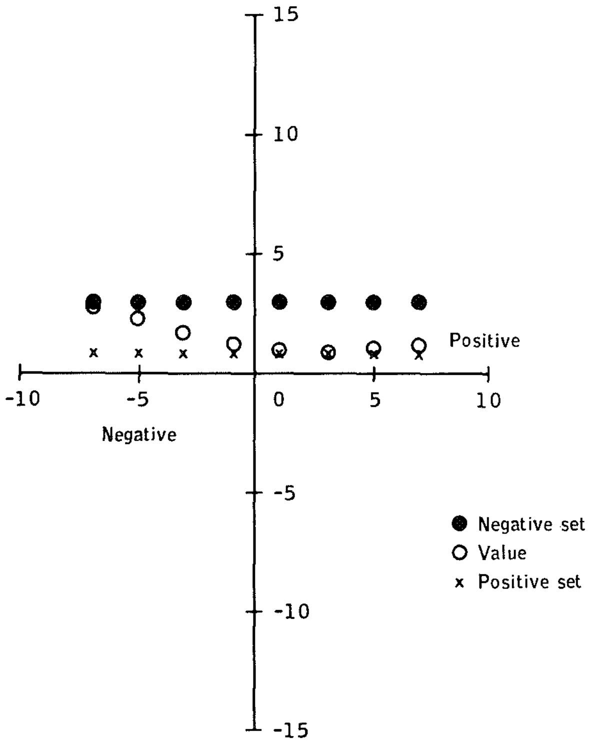


Figure 8. - Linear (narrow) curve-fit example.

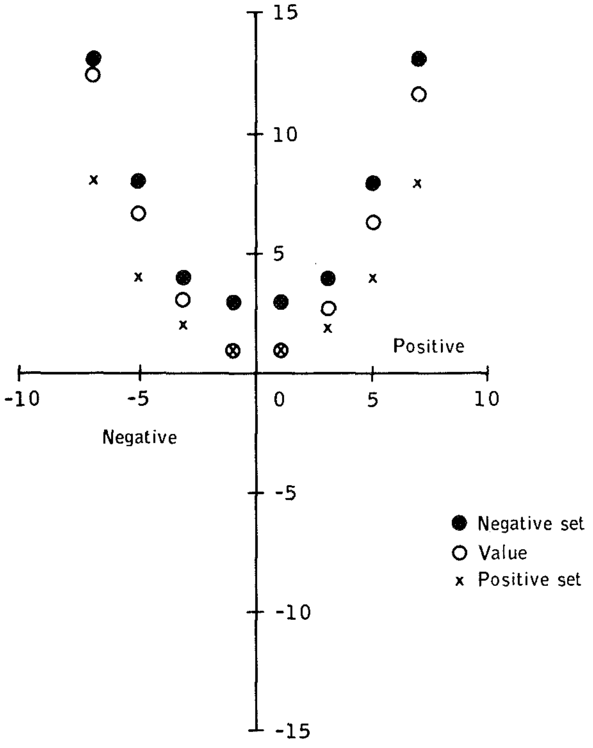


Figure 9. - Quadratic curve-fit example.

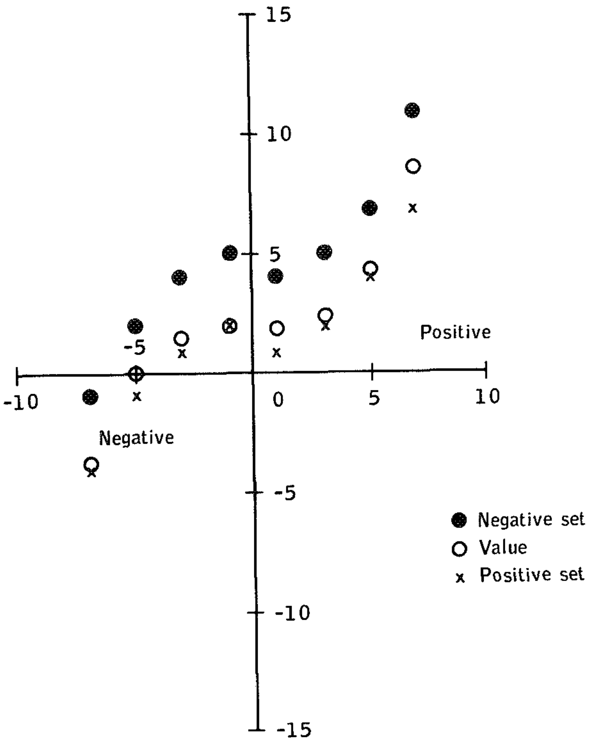


Figure 10. - Cubic curve-fit example.

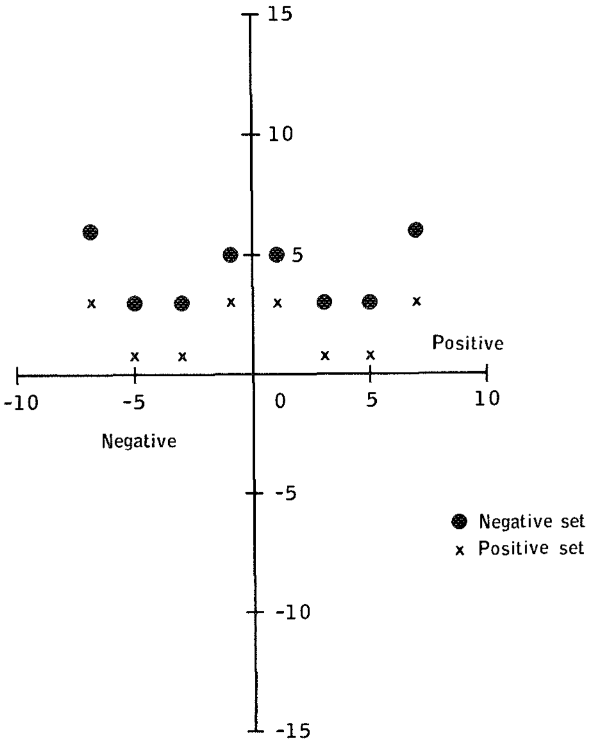


Figure 11. - Quartic curve-fit example (did not converge).

CONCLUDING REMARKS

Pattern recognition is based on the idea of examining of set of measurements (called a pattern) of a phenomenon and deriving decisions from the characteristics of the pattern. Usually, one of two decisionmaking procedures is used. The nonparametric procedure is a deterministic approach that seeks out separating boundaries or some other rules that allow precise decisions. The parametric machines attempt to make optimum decisions based on the statistical properties of patterns, always recognizing the probabilities of making an error.

The components of the pattern must be selected carefully to represent, with as few components as possible, the phenomenon of interest. This careful selection of components is necessary because processing time increases as a function of the number of components of the pattern. In the example, processing time increases exponentially with the number of components.

Pattern recognition is an active field. Several hundred technical reports and journal articles on the subject are published annually. Most large engineering schools have established specialty areas in the field of learning machines. The general theory is now well established and may be adapted easily to the needs of the individual user.

Manned Spacecraft Center
National Aeronautics and Space Administration
Houston, Texas, February 20, 1970
914-50-50-16-72

REFERENCES

1. Johnson, David L.: Machine Learning for General Problem Solving. AD 608 544, Dept. of Electrical Eng., Univ. of Wash. (Seattle, Wash.), Oct. 1964.

2. Locke, John: An Essay Concerning Human Understanding.

3. Boole, George: An Investigation of the Laws of Thought. Dover Publications, 1854.

4. Rosenblatt, F.: The Perceptron: A Perceiving and Recognizing Automation. Cornell Lab. Rept. 85-460-1, Jan. 1957.

5. Highleyman, W. H.: Linear Decision Functions, with Application to Pattern Recognition. Ph. D. Dissertation, Elect. Eng. Dept., Polytechnic Institute of Brooklyn, June 1961.

6. Braverman, D.: Machine Learning and Automatic Pattern Recognition. Stanford Electronic Lab. Tech. Rept. No. 2003-1, Feb. 17, 1961.

7. Braverman, D.: Learning Filters for Pattern Recognition. IRE Transactions on Information Theory, vol. IT-8, July 1962.

8. Abramson, N.; and Braverman, D.: Learning to Recognize Patterns in a Random Environment. IRE Transactions on Information Theory, vol. IT-8, Sept. 1962.

9. Sebestyen, G. S.: Decision Making Processes in Pattern Recognition. The MacMillan Co., 1962

10. Fralick, S. C.: Learning to Recognize Patterns Without a Teacher. Stanford Research Lab. Rept. SEL-65-011, March 1965.

11. Nilsson, N. J.: Learning Machines. McGraw-Hill, 1965.

12. Abramson, N.; Braverman, D.; and Sebestyen, G. S.: Pattern Recognition and Machine Learning. IEEE Transactions on Information Theory, Oct. 1963.

13. Abramson, N.: An Introduction to Bayes Decision Procedures. Proceedings of Symposium on Decision Theory and Application to Electronic Equipment Development, Rome Air Development Center, May 1960.

14. Johnson, David L.: A Pattern Recognition Model for On-Line Curve Fitting: An Application of Threshold Theory. AFOSR 68-0037, Dept. of Electrical Eng., Univ. of Wash. (Seattle, Wash.), Aug. 1967.

APPENDIX

AN nth-DEGREE ϕ MACHINE

The computer program that is shown in figure 5 uses FORTRAN V and has been run on the Univac 1108 computer. A block diagram is shown in figure 12.

The following list is a description of the key data cards that are used in the operation of this program.

1. Initial Data Card — This card contains the key parameters that are used in starting the program. Because maximum flexibility was desired, the program was designed to do, with one input, several training runs with different orders of discriminant functions. The following parameters must appear on the initial data card.

a. NRUN (integer value in columns 1 to 10) — Controls the number of training runs
b. NPTS (integer value in columns 11 to 20) — Specifies the number of vectors in each training set
c. NX (integer value in columns 21 to 30) — Specifies the number of descriminant values to be calculated after training
d. SLIP (floating point number in columns 31 to 40) — Sets the spacing between the discriminant values to be calculated
e. STRT (floating point number in columns 41 to 50) — Sets the value of the first point at which a discriminant value is to be calculated

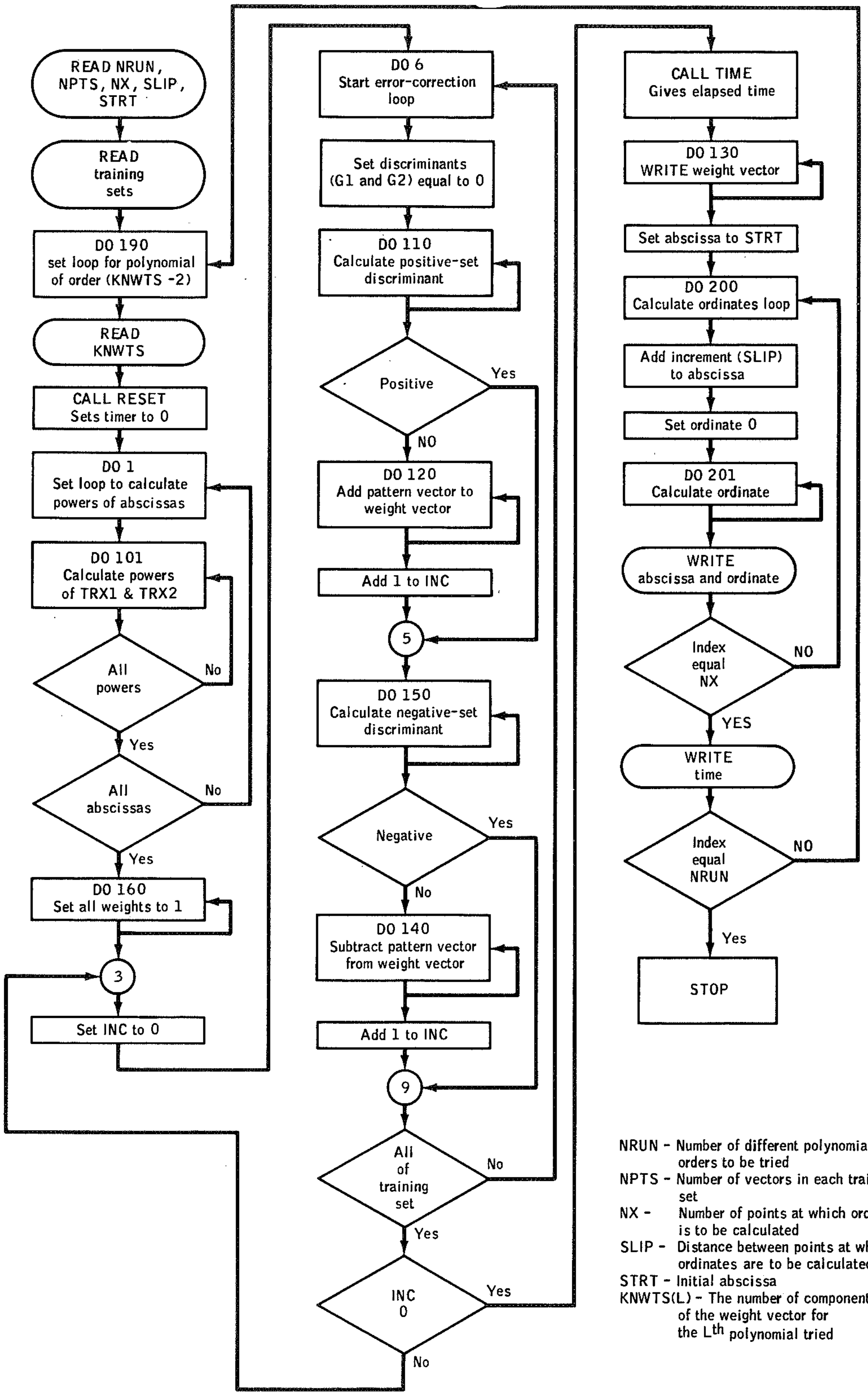
2. Training-Vector Data Cards — There are the number NPTS of these cards. All vectors belonging to one set are separated from those of the other set. Each set must contain an equal number of vectors. The vectors are punched, one from each set per card, with the abscissa and ordinate from set 1 in columns 1 to 10 and 11 to 20, respectively, and the values from set 2 in columns 21 to 30 and 31 to 40, respectively.

3. Discriminant-Order Cards — There are the number NRUN of these cards. An integer value, which is the order of the discriminant function to be used during that run, is placed in columns 1 to 10.

This program is a straightforward representative implementation of the error-correction training method. The following list is a step-by-step explanation of the block diagram in figure 12.

1. After reading this Initial Data Card and the Training-Vector Data Cards, the program sets up the outer loop (DO 190) that counts the number of runs. The program then reads the order of the first discriminant function to be derived and sets the internal timer to 0.

2. The DO 1 loop, which maps the input training vectors into ϕ space, is the ϕ processor. The results of this loop are two sets of vectors in memory that have


READ NRUN, NPTS, NX, SLIP, STRT
READ training sets
DO 190 set loop for polynomial of order (KNWTS -2)
READ KNWTS
CALL RESET Sets timer to 0
DO 1 Set loop to calculate powers of abscissas
DO 101 Calculate powers of TRX1 & TRX2
All powers
No
Yes
All abscissas
No
Yes
DO 160 Set all weights to 1
3
Set INC to 0
DO 6 Start error-correction loop
Set discriminants (G1 and G2) equal to 0
DO 110 Calculate positive-set discriminant
Positive
Yes
NO
DO 120 Add pattern vector to weight vector
Add 1 to INC
5
DO 150 Calculate negative-set discriminant
Negative
Yes
No
DO 140 Subtract pattern vector from weight vector
Add 1 to INC
9
All of training set
No
Yes
INC 0
Yes
No
CALL TIME Gives elapsed time
DO 130 WRITE weight vector
Set abscissa to STRT
DO 200 Calculate ordinates loop
Add increment (SLIP) to abscissa
Set ordinate 0
DO 201 Calculate ordinate
WRITE abscissa and ordinate
Index equal NX
NO
YES
WRITE time
Index equal NRUN
NO
Yes
STOP

NRUN - Number of different polynomial orders to be tried
NPTS - Number of vectors in each training set
NX - Number of points at which ordinate is to be calculated
SLIP - Distance between points at which ordinates are to be calculated
STRT - Initial abscissa
KNWTS(L) - The number of components of the weight vector for the L^{th} polynomial tried

Figure 12. - An nth-degree ϕ machine block diagram.

NWTS number of components each, instead of two components each. After this loop, the machine operates linearly.

3. All weight-vector components are set to 1.

4. The error counter, INC, is set to 0.

5. The DO 6 loop starts the error-correction procedure. A discriminant value is calculated for each training vector. A negative error causes the training vector to be added to the weight vector. Conversely, a positive error causes the training vector to be subtracted from the weight vector. If an error occurs, INC is incremented by 1.

6. After all training vectors have cycled through DO 6, INC is tested. If INC is not 0, the program cycles back to 3, and the process is repeated. If INC is 0, the computer does CALL TIME, which gives the time elapsed since CALL RESET.

7. The program prints the components of the weight vector.

8. The program calculates and prints the desired number of discriminant values at the desired abscissas.

9. The program prints the time required for training.

10. If the DO 190 index has not been satisfied, the program recycles. Otherwise, the program STOPS.

NASA-Langley, 1970 —— 8 S-224

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546