4-1977

A Comparison of Various Pattern Recognition Techniques

Larry Wood
Western Kentucky University

Wood,

Larry H.

1977

A COMPARISON OF VARIOUS PATTERN RECOGNITION TECHNIQUES

A Thesis
Presented to
the Faculty of the Department of Chemistry
Western Kentucky University
Bowling Green, Kentucky

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Larry H. Wood
April 1977

A COMPARISON OF VARIOUS PATTERN RECOGNITION TECHNIQUES

Approved May 12, 1977 :
(Date)

Earl F. Pearson
Director of Thesis

Curtis C. Wilkins

John W Reasoner

Elmer Gray
Dean of the Graduate School

ACKNOWLEDGEMENT

I wish to express my gratitude to Dr. Earl F. Pearson, my research director, who gave of his time and knowledge in doing this research. His guidance and support are greatly appreciated. I want to express thanks to the faculty and staff of the Chemistry Department at Western Kentucky University for their suggestions and help in the preparation of this work and during the time I attended school there.

Larry H. Wood

TABLE OF CONTENTS

LIST OF TABLES

A COMPARISON OF VARIOUS PATTERN RECOGNITION TECHNIQUES

Larry H. Wood April 1977 43 pages

Directed by: Earl F. Pearson

Department of Chemistry Western Kentucky University

A comparison of the reliability of three pattern recognition classifiers has been made using data having a great amount of variation. The basic concepts of the Linear Learning Machine, the K Nearest Neighbor Classifier, and the Potential Function Classifier are presented. Prediction of whether a student would pass or fail freshmen Chemistry 120 was made, based on various test results. The Linear Learning Machine was found to be an unworkable classifier for this kind of data. Both the Potential Function Classifier and the K Nearest Neighbor classifier were acceptable with the Potential Function Classifier being generally a better classifier.

I. INTRODUCTION

Pattern Recognition is the technique of using automatic procedures to classify samples of data into discrete categories based on the similarities they exhibit to known groups of samples. The power of this technique is demonstrated in its ease of application to extremely difficult and sometimes unsolvable problems. It is now being routinely applied to problems which were formerly considered to be approachable only by humans.

Research in this area has been stimulated by the promise this technique holds. The perfection of such machines will not only allow for better man-machine interfacing but also will allow their substitution for humans in the performance of routine information processing tasks; tasks that computers will be able to perform more quickly, accurately, safely, and inexpensively than humans.

Considerable research has been done in the theory and methods of pattern recognition(1-12) and it has been applied with considerable success to the classification of various types of spectral data, i.e., mass spectra(13-16), NMR spectra(17,18), Infrared spectra(19,20), etc. However, few applications have been made on data which shows considerable overlap of categories and variation of data within each category. This paper will deal with such an application and the various attempts to improve the technique.

II. HISTORICAL

Initial efforts in the study of automatic pattern recognition may be traced back to the early 1950's when the digital computer first became a readily-available information processing tool. These early attempts were quite limited in scope due to their simplicity. In the late 1950's studies of perception by the human brain and the similarities of computers and the brain led to a better model for information, storage and organization. The major approaches at this time to the pattern recognition problem were based primarily upon statistical decision theory and threshold logic principles. With the advent of larger, more powerful and more efficient computers the research in pattern recognition system design gained momentum. The need for more efficient communication between man and machine became evident as computers were applied to more systems. Research to this time had been in the domain of applied mathematicians, statisticians, and computer-oriented engineers.

According to Shoenfeld and DeVoe(21), the first chemical application of pattern recognition was made in 1964 when Tal'roze used it in the identification of organic substances from their mass spectral lines. Activity picked up in the late 1960's and early 1970's with most of the work being done in the various areas of spectroscopy. Most of the work in this area has been done by several researchers: T. L. Isenhour, University of North Carolina; P. C. Jurs, Pennsylvania State University; B. R. Kowalski, University of Washington; C. F. Bender and S. L. Grotch, UCRL, Livermore; C. W. Wilkins, University of Nebraska; and others.

Recent literature surveys and reviews demonstrate the great variety of applications possible with pattern recognition.[(21-26)]

At present, work is being done to compare different pattern recognition systems and to improve their versatility and accuracy, to reduce computer usage time, and to develop even better pattern recognition techniques.

III. THEORETICAL

Pattern recognition is an important category in the much broader field of artificial intelligence. It can be said that the ultimate goal of artificial intelligence is to construct machines that will perform the same functions as the human brain but at a faster rate and with a greater degree of reliability. Because of the many approaches that can be taken to it, pattern recognition itself is a quite broad field. Numerous articles and books have been published dealing with the statistics and theory of the pattern recognition system.(1-12)

Basically there are two approaches to developing the pattern recognition system--parametric and nonparametric. Parametric methods assume that the probability density functions are known or can be estimated. Bayes strategies are employed in the learning nad decision process. This approach is useful for some kinds of data but for chemical data it is, in most cases, impractical. Most sets of chemical data are only partially complete so little can be known or even assumed about the underlying statistical distributions of the data. Thus a parametric approach is not possible.

The nonparametric approach has the advantage of being totally emperical, that is, it needs to make no assumptions about any kind of relationship between a set of data points and a category. This allows it to be applied to both simple and very complex data with equal ease.

A simplified block diagram of a pattern recognition system is shown in Figure 1. There are two problems the pattern recognition system must deal with as emphasized by the two blocks--feature extraction and classification. The feature selection block may be further broken down into three subunits as shown in Figure 2: the transducer, the feature selector, and the preprocessor.

The transducer has the function of transforming raw data into a form compatable with the language of the classification device. Typically the transducer is an Analog-to-Digital Converter to convert the analog data signal into a digital signal which can be utilized by the computer the classifier is using.

Feature selection deals with the decision of what measurements to take from the input data. Usually this decision is rather subjective and is dependent on practical matters such as cost, availability, etc. There is little general theory on what measurements to take, but the approach usually taken to determine the contribution a measurement makes to the accuracy of a classification is to leave it out and note the change in accuracy of the classification.

Interrelated with the feature selector is the preprocessing unit. Together they aid the classifier in making a classification. Preprocessing includes algebraic transformations such as squaring, extraction of roots, and taking of logs; and changes in variables through transforms such as the Fourier transform. Preprocessing is quite useful for the following two reasons: first, it can spread the clusters of data points of categories further apart thus making classification easier and, second, it can reduce the dimensionality of the data by discarding dimensions deemed expendable or by combining two or more dimensions in some way. The advantage of dimensionality reduction will be seen later when the calculation made by the classifier during a classification process is

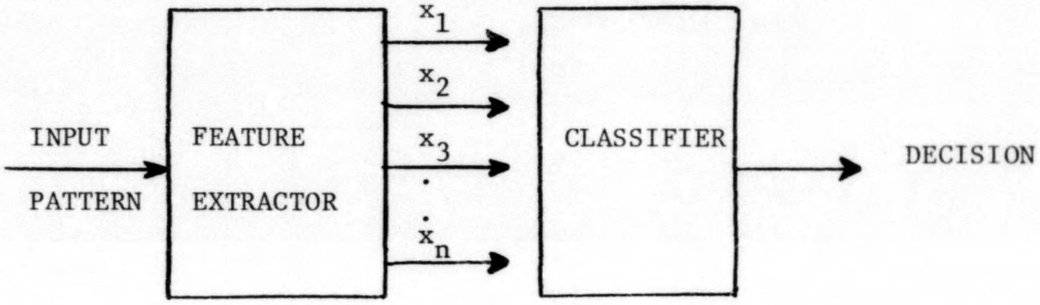

FIGURE 1. A PATTERN RECOGNITION SYSTEM

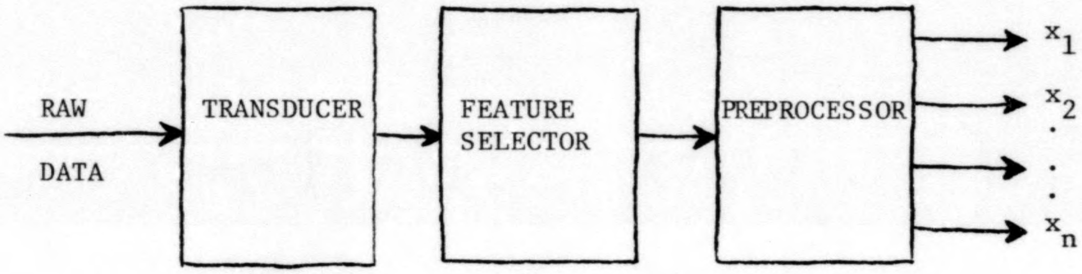

FIGURE 2. THE FEATURE EXTRACTOR

examined. The preprocessing of data has received considerable attention but as of yet no definite procedures for this operation have been defined since it is not completely independent of the classifier.

After the operation of the feature extractor on the raw data it may be represented as an N-dimensional vector

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

These N-dimensional "pattern" vectors can be represented as a point in an N-dimensional hyperspace. Figure 3 illustrates pattern vectors in a two-dimensional pattern space. Suppose that there are two or more categories of pattern vectors represented in the pattern space. The task of the classifier is to assign each pattern vector or point in the hyperspace to its proper class. This can be accomplished by the partition of the hyperspace into mutually exclusive regions with each region corresponding to a particular pattern class. The classifier defines the decision surface between the regions of the hyperspace. Numerous classifiers have been presented in the literature[(3)] but generally they are modifications or combinations of the following three: the Linear Learning Machine, the K-Nearest Neighbor classifier, and the Potential Function Classifier.

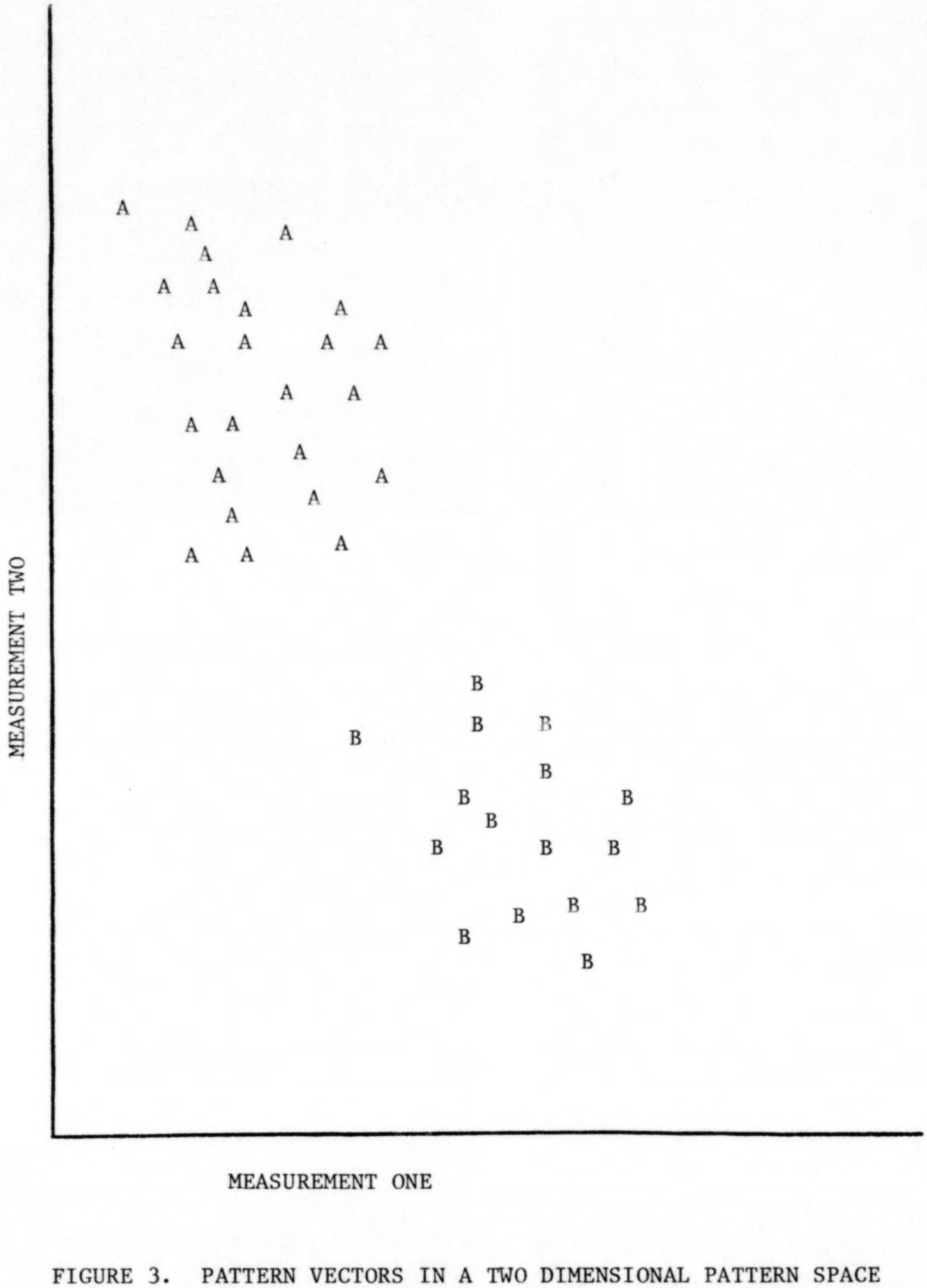

FIGURE 3. PATTERN VECTORS IN A TWO DIMENSIONAL PATTERN SPACE

THE LINEAR LEARNING MACHINE

The linear learning machine[24] is a binary pattern classifier which defines the decision surface between two pattern classes by repeatedly correcting the orientation of the decision surface until all patterns of a training set are classified correctly. The principal decision making of the linear learning machine is performed by the Threshold Logic Unit (TLU). The TLU's used are capable of placing a pattern in one of two classes. The original pattern vector is represented as X. The TLU implements a plane of the same dimensionality as the patterns which will separate the data into the two classes. Since it is very convenient to have the decision surface pass through the origin of the hyperspace (it becomes simplier mathematically), all of the original vectors are augmented with an N + 1 component to give a new vector. Hence,

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_n \end{bmatrix}$$

and

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ . \\ x_n \\ x_{n+1} \end{bmatrix}$$

Now an N + 1 dimensional hyperplane which passes through the origin may be used to separate the pattern sets. A convenient way to determine whether a point lies on one side of the hyperplane or the other is to use a vector normal to the plane at the origin. This vector, called the weight vector (W), defines the locus of points which make up the hyperplane separating the two classes. Because W is perpendicular to the plane, the dot product of W with any pattern vector (Y) will determine on which side of the plane the pattern vector lies.

$$s = W \cdot Y = |W|\ |Y| \cos\theta$$

Where θ is the angle between the two vectors. $|W|$ and $|Y|$ are always positive, so for

$$-90° < \theta < 90° \quad \cos\theta > 0$$

$$\text{and} \quad s > 0$$

$$90° > \theta > 270° \quad \cos\theta < 0$$

$$\text{and} \quad s < 0$$

Thus for pattern vectors less than 90° from W (and thereby on one side of the plane) the dot product is always positive, while for patterns on the other side of the plane the dot product is always negative. Another form of the dot product is

$$s = W \cdot Y = w_1 y_1 + w_2 y_2 + \cdot \cdot \cdot + w_n y_n + w_{n+1} y_{n+1}.$$

This type of computation is quite easy for the digital computer. One can arbitrarily assign the positive side of the hyperplane as category one and the negative side as category two. A block diagram illustration of the Threshold Logic Unit is shown in Figure 4.

To develop a decision maker for a given classification, a training set of pattern vectors, for which the correct categories are known, is presented to the classifier one at a time, and whenever a misclassification occurs, a correction process (negative feedback) is applied to the weight vector. This process continues until all patterns of the training set are correctly classified or it is determined that the

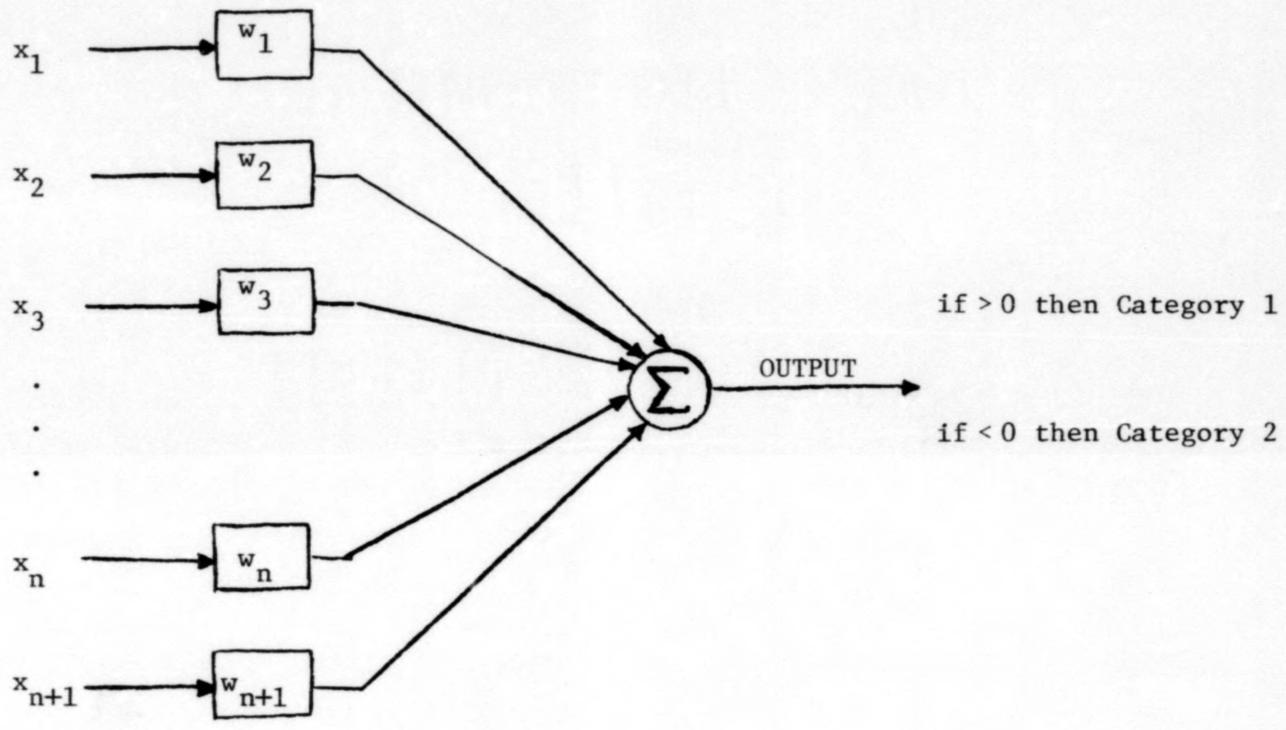

FIGURE 4. THRESHOLD LOGIC UNIT

patterns will not result in training in which case the computer terminates the training after a certain number of feedbacks. Figure 5 gives a flow diagram of this process. One of the simplest and most effective feedback methods to date is to move the decision hyperplane along the perpendicular axis between the misclassified point and the plane, so that after the correction it is the same distance on the correct side of the point as it was previously on the incorrect side. This movement is accomplished by adding an appropriate multiple of the pattern vector (Y) to the weight vector. Thus

$$W \cdot Y_i = s$$

where s has the incorrect sign for classifiying Y_i. Therefore, weight vector, W', is desired such that

$$W' \cdot Y_i = -s$$

by combining a fraction, c, of Y_i with W.

$$W' = W + c\ Y_i$$

By combining the above equation we can get the new weight vector, W', from the equation

$$W' = W - \frac{2\ s}{Y_i \cdot Y_i} \quad Y_i$$

This method of classification works well on a pattern space where the categories are easily linearly separable; but when they are not, the classifier begins to use excessive computer time with no results other than the assumption that the categories are linearly inseparable. (It cannot be proven that two categories are linearly inseparable until the computer performs an infinite number of feedbacks.)

Some of the problems associated with linear inseparability may be overcome by the addition of a "width" parameter(28) to the linear learning machine. This parameter creates a null region in the case of

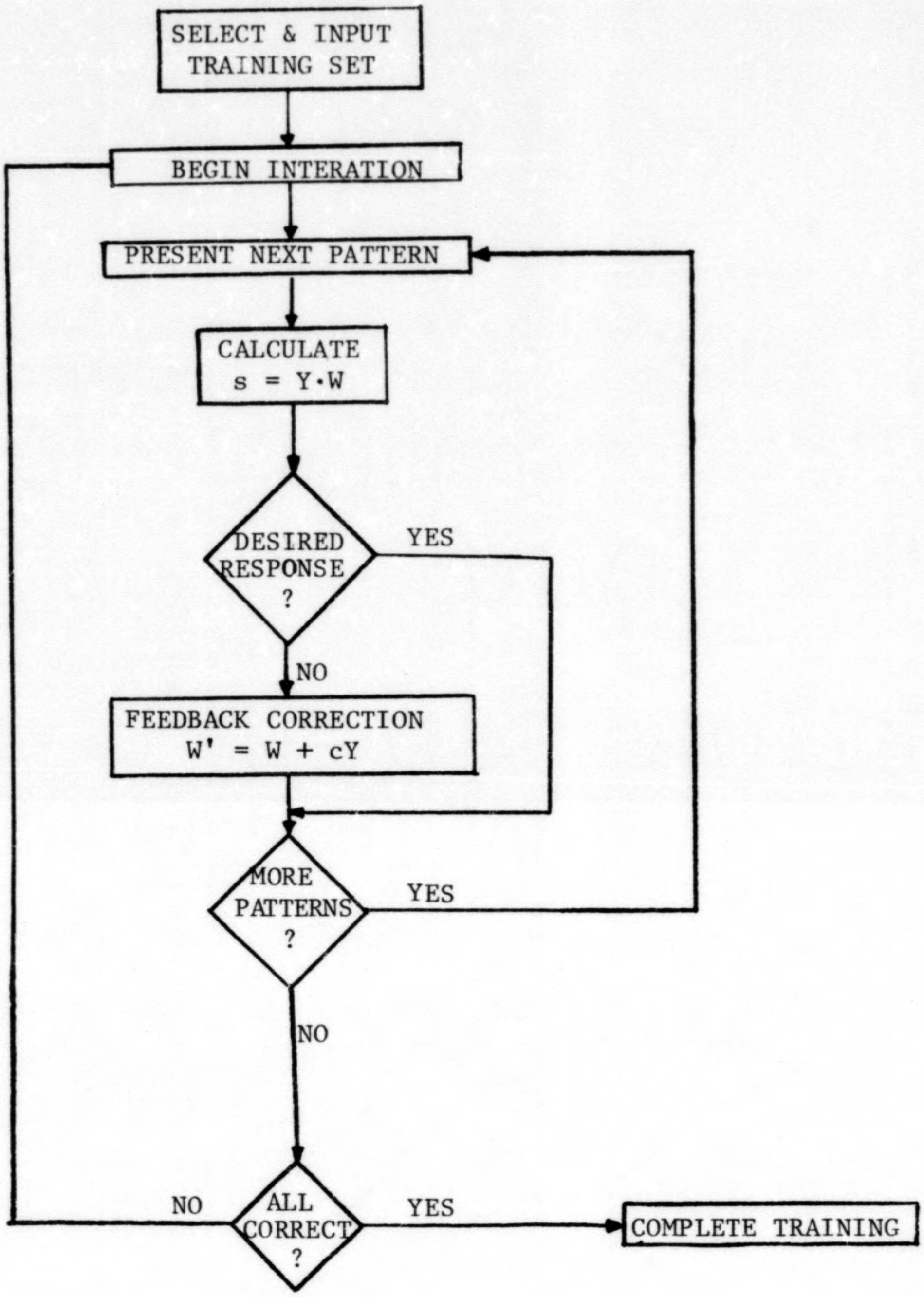

FIGURE 5. TRAINING PROGRAM

linearly separable categories and defines a region of overlap containing the subset of the inseparable data in the case of linearly inseparable categories. By changing the value of this parameter, a confidence value may be obtained for that particular classifier. This parameter is illustrated by Figure 6.

A major advantage of the linear learning machine is that once the values of the weight vector have been determined the task of classifying unknown patterns can be accomplished on a simple calculator.

THE K NEAREST NEIGHBOR CLASSIFIER

The K Nearest Neighbor (KNN) classifier(18) is quite simple both in conception and in computation which is one of its advantages. An unknown pattern is classified according to the majority vote of its K Nearest Neighbors in the N-dimensional space. Its nearest neighbors will be members of a training set whose categories are known. Computationally, all that is required is to compute the distances between the unknown pattern vector and all of the pattern vectors in the training set. The KNN classifier is a multicategory classifier. Any number of categories can be represented in the training set and the unknown pattern will still be classified according to the majority vote. The nearer neighbors are weighted to have a larger vote than those farther away. The KNN classifier is also nonlinear which allows it to classify patterns such as those represented in Figure 7. The reason for this is that the Euclidean distance in an N-dimensional space between point i and j,

$$D_{ij} = \left[\sum_{k=1}^{n} (x_{ik} - x_{jk})^2 \right]^{\frac{1}{2}}$$

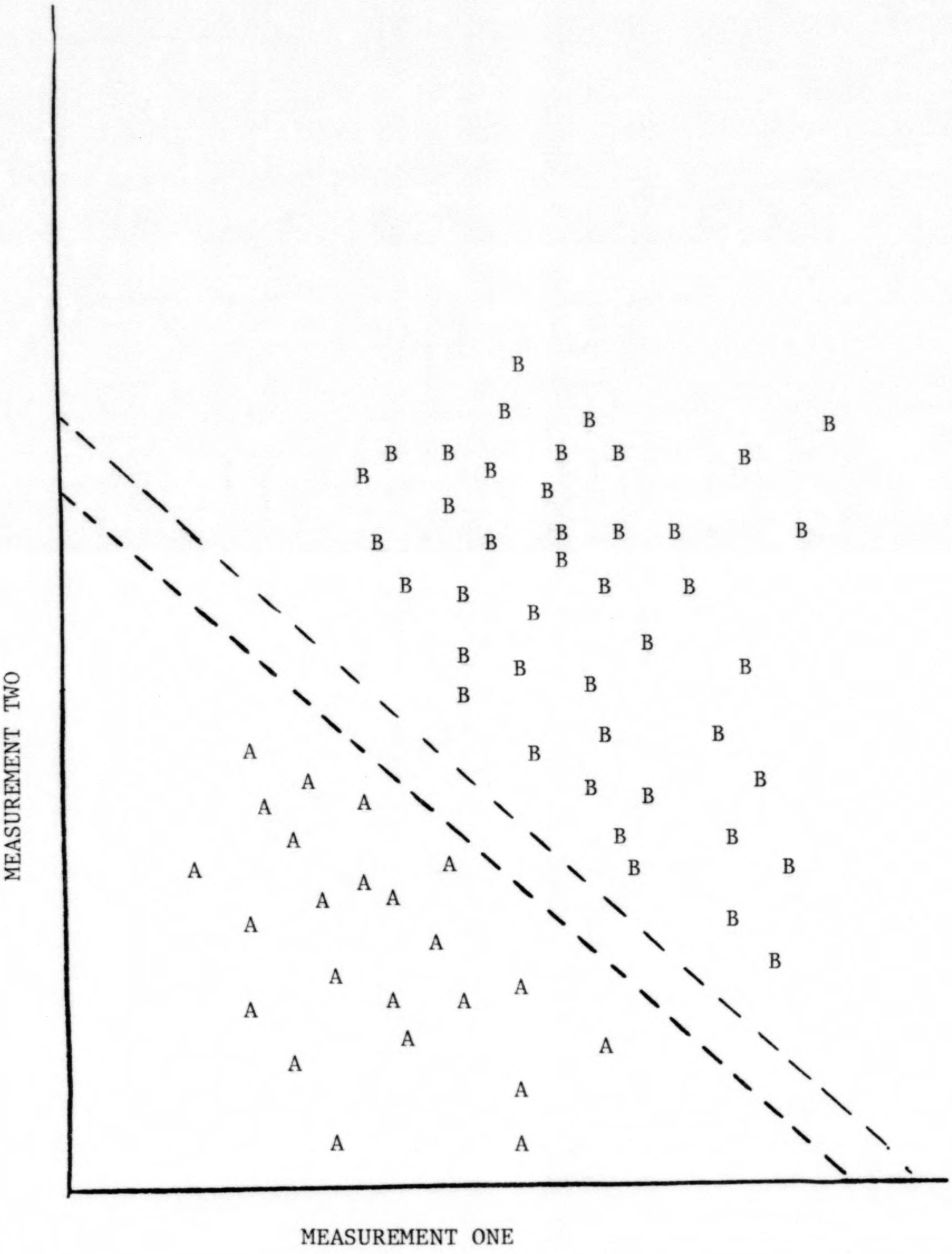

FIGURE 6. A NULL REGION IN THE PATTERN SPACE GENERATED BY USE OF THE WIDTH PARAMETER.

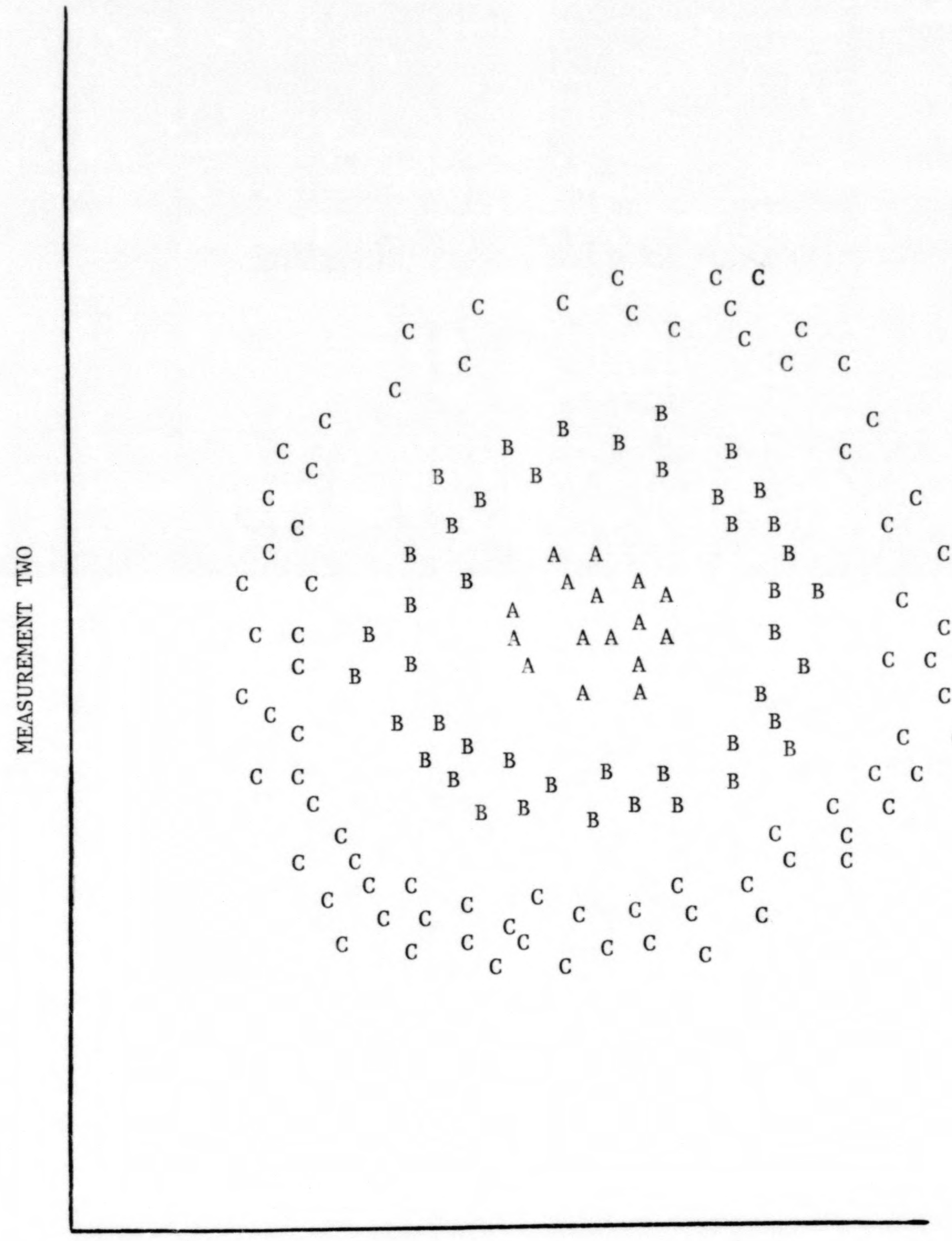

FIGURE 7. THREE CATEGORIES OF LINEARLY INSEPARABLE DATA

is a nonlinear function of the features. Many studies have been made on the KNN classifier to give it a firm statistical foundation and it is generally accepted as a standard by which other classifiers are judged. There is no training required for the KNN classifier but it sometimes uses a great amount of computer time to do the computations necessary to make a decision.

THE POTENTIAL FUNCTION CLASSIFIER

The Potential Function Classifier (PFC)(27) is also a quite simple classifier which, as the KNN classifier, requires no training. To visualize the PFC one can consider each point in a hyperspace as having a positive or negative charge with each point having its own potential field. One can find a zero potential surface in the hyperspace between the two categories of positive and negative charges. This zero potential surface is the decision surface of the classifier. Figure 8 shows a two-dimensional potential function decision surface. Computationally, the decision process involves the evaluation of the following equation:

$$M = \text{sgn}(\sum_{j=1}^{n+1} P(D_{+1,j}) \, W_{+1,j} - \sum_{j=1}^{n-1} P(D_{-1,j}) \, W_{-1,j} + T)$$

where $D_{+1,j}$ is the Euclidean distance from the unknown pattern vector to the j th pattern vector in the positive category of the training set.

$D_{-1,j}$ is the Euclidean distance from the unknown pattern vector to the j th pattern vector in the negative category of the training set.

$P(D)$ is some function of the distance.

$W_{+1,j}$ is a weight factor for the positive distances.

$W_{-1,j}$ is a weight factor for the negative distances.

T is a scalar quantity which can be either positive, negative, or zero.

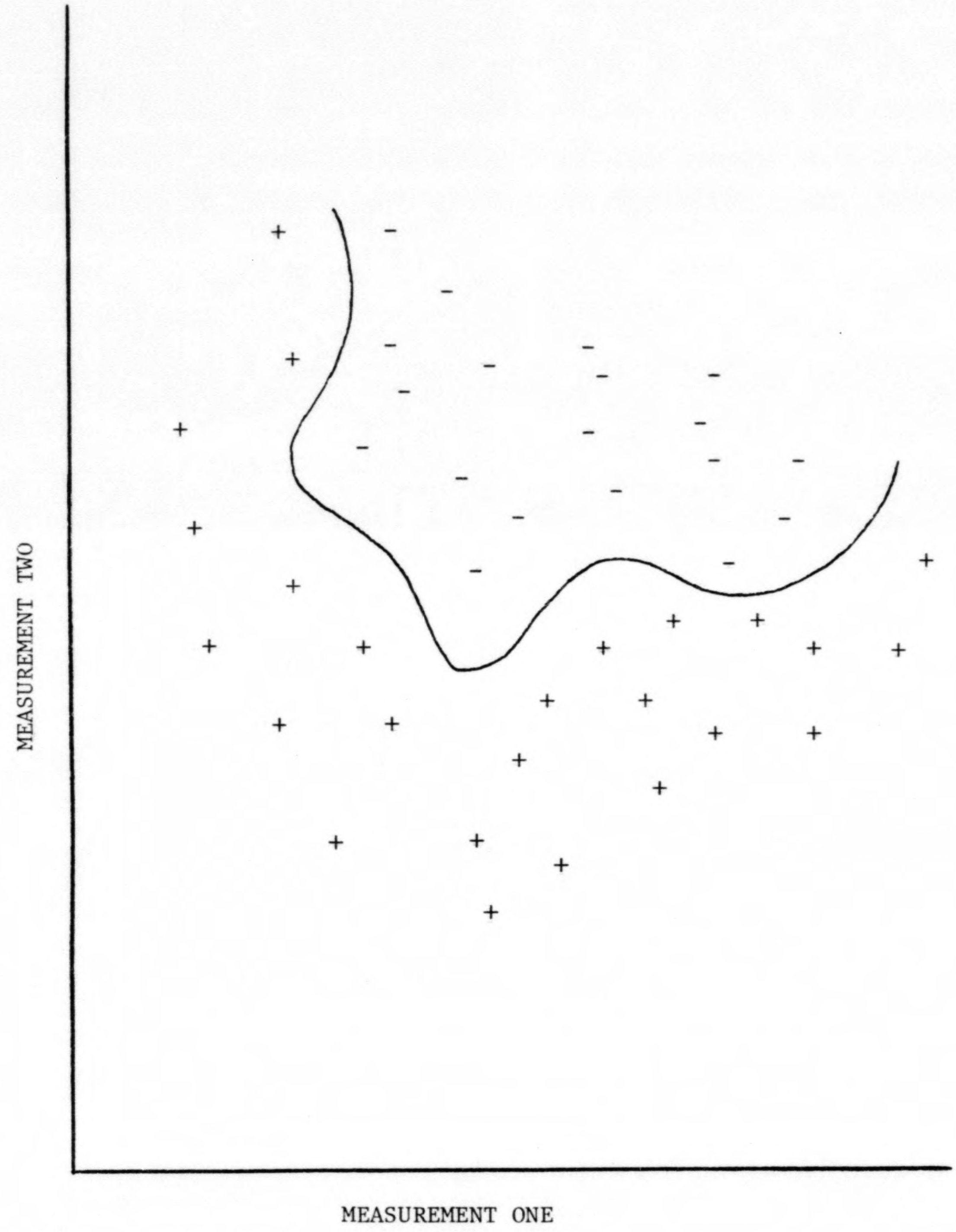

FIGURE 8. DECISION SURFACE OF A POTENTIAL FUNCTION CLASSIFIER IN A TWO DIMENSIONAL SPACE.

The PFC as presented here can only be applied to two category systems. It is somewhat more reliable than the KNN classifier and usually requires less computer usage time.

IV. EXPERIMENTAL

To test the different pattern recognition techniques a totally different kind of data was chosen than that utilized by other researchers to date. It has been desired for some time to find a better way to utalize the data available on students, i.e., ACT scores, Toledo scores, High School Grade Point Averages, etc., to advise them on taking Chemistry 120. Since other methods have met with only limited success it seemed likely that the emperical approach of pattern recognition should yield better results. Three hundred and forty-five sets of data were taken over a period from 1971 to 1975. The following information was taken on each student: ACT composite score, ACT English score, ACT math score, ACT natural sciences score, Toledo composite score, Toledo math score, Toledo chemistry score, High School Grade Point Average, and the number of hours taken during that semester. This information was obtained from the Chemistry Department and the Registrar's Office at Western Kentucky University.

The different classifiers tested were the Linear Learning Machine, the Nearest Neighbor Classifier, and the Potential Function Classifier. Both the Linear Learning Machine and the KNN Classifier have been used quite often in chemical applications but little has been done with the Potential Function Classifier. Programs of each of the classifiers are contained in the appendices of this paper.

A PDP-11/45 computer was utilized for this work.

V. DISCUSSION AND RESULTS

Several preprocessing techniques were tried with the raw data scores and the best was found to be that of scaling the scores from 0 to 1 and then squaring the scaled quantity. The effect of this is to spread the two categories farther apart to make classification easier.

Several attempts were made to classify the data using the linear learning machine classifier but in all attempts the training routine was unable to achieve convergence. A width parameter was included to allow the training routine to be completed but it was found that a majority of the data lay in the region of overlap making any classification attempts with the resulting weight vector useless. It was concluded that the linear learning machine classifier was not able to operate effectively on this kind of data.

The next attempt at classifying the data was made with the K Nearest Neighbor Classifier. Since the KNN Classifier does not require a training routine, this problem is eliminated. Again 300 patterns were taken as the training set and prediction was attempted on the remaining 45. Since the KNN Classifier can be a multicategory classifier just as easily as it can be a binary classifier, attempts were made at predicting both the letter grade and whether a student would pass or fail. The results of this were reasonable (averaging 80%) for the pass-fail predictions but were quite poor (about 35%) for predicting the letter grade.

With the KNN Classifier it is easy to vary the number of neighbors which participate in the voting for the category of the unknown pattern.

Therefore classifications were made while varying the value of k (the number of nearest neighbors) and keeping the training and prediction sets the same. The results of this are shown in Table 1 and Figure 9. The best k value was found to be in the range of 10 to 15. It is expected that the optimum k value will vary depending on the kind of data the classifier operates on.

To determine the dependence of the predictability on the training set size, a series of classifications were made while varying the training set size. The prediction set was the same for each classification attempt. The results of this are shown in Table 2 and Figure 10. The results show that the performance of the classifier improves with increasing training set size, thus if more data were available, the quality of the classifier would improve. However, a larger training set means longer calculations and more computer usage time which could lead to prohibitive costs.

The Potential Function Classifier was next tested with the data set. The training and prediction sets were set up as before. The positive and negative weight factors were obtained by considering the number of patterns in each category in the training set and assigning weight factors to give each category an equal weight. Tests were run to determine the best function of the Euclidean distance. The results of this are shown in Table 3 and Figure 11. This function will vary for each different application of the PFC but for this application the best results were obtained with P(D) equal to D^{-15}.

A determination of the effect of training set size on the PFC was also made. The KNN Classifier was also tested on the same training and prediction set to give a comparison of the two classifiers.

The results of these tests are given in Table 4 and Figure 12. In this test the predictability of the PFC was slightly better than that of the KNN Classifier.

Since all the available data was being used, the only other test that could be made along the line of changing the training set and prediction set was to reshuffle the data cards. Both the PFC's and KNN Classifier's predictability varied considerably with the reshuffling of the data. Results of both varied from 60 to 85-90% with the PFC generally doing slightly better than the KNN. This variation demonstrates the variation in the data which made it impossible to distinguish good from bad data in most cases.

TABLE 1. THE DEPENDENCE OF PREDICTABILITY ON THE NUMBER OF NEAREST NEIGHBORS(K).

VALUE OF K	% CORRECT PREDICTION OF PASS/FAIL	% CORRECT PREDICTION OF LETTER GRADE
1	84.45	35.56
5	82.23	35.56
10	86.67	40.00
15	84.45	48.89
20	84.45	42.23

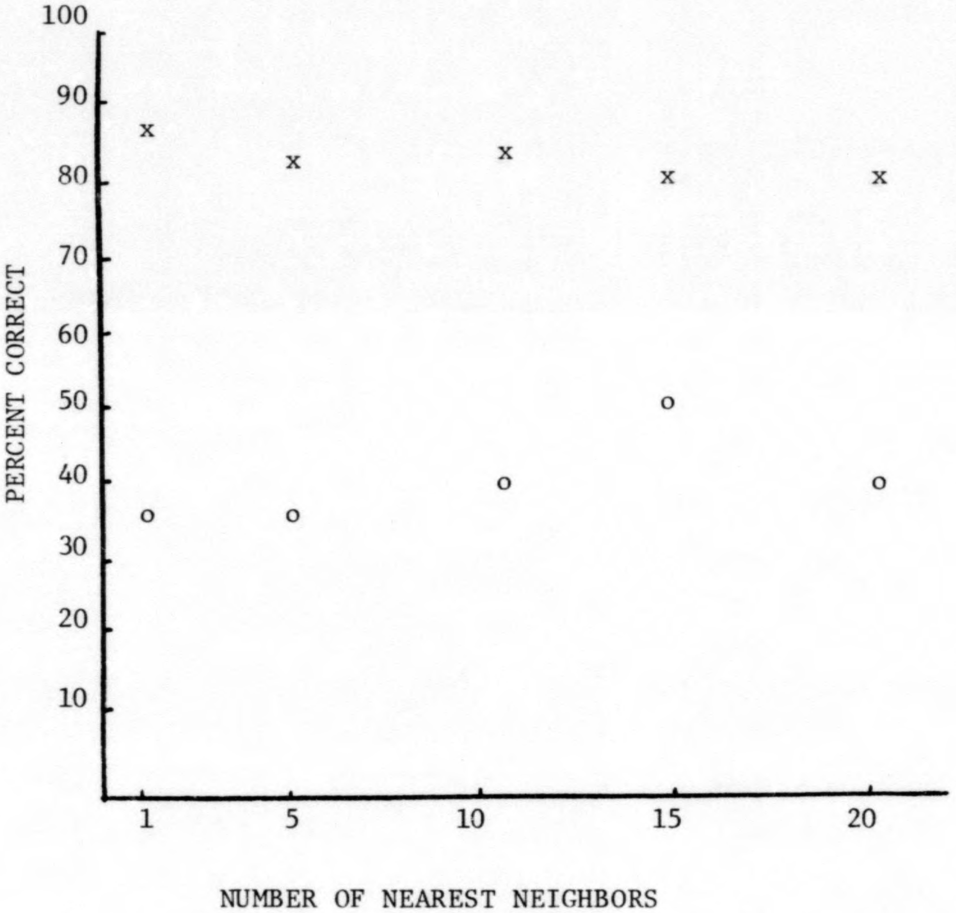

FIGURE 9. THE DEPENDENCE OF PREDICTABILITY ON THE NUMBER OF NEAREST NEIGHBORS (K).

Pass/Fail Correct Predictions = x
Letter Grade Correct Predictions = o

TABLE 2. THE DEPENDENCE OF PREDICTABILITY ON TRAINING SET SIZE.

NUMBER OF PATTERNS IN TRAINING SET	% CORRECT PREDICTION OF PASS/FAIL	% CORRECT PREDICTION OF LETTERGRADE
50	71.12	33.34
100	73.34	22.23
150	68.89	28.89
200	73.34	35.56
250	75.56	35.56
300	80.00	37.78

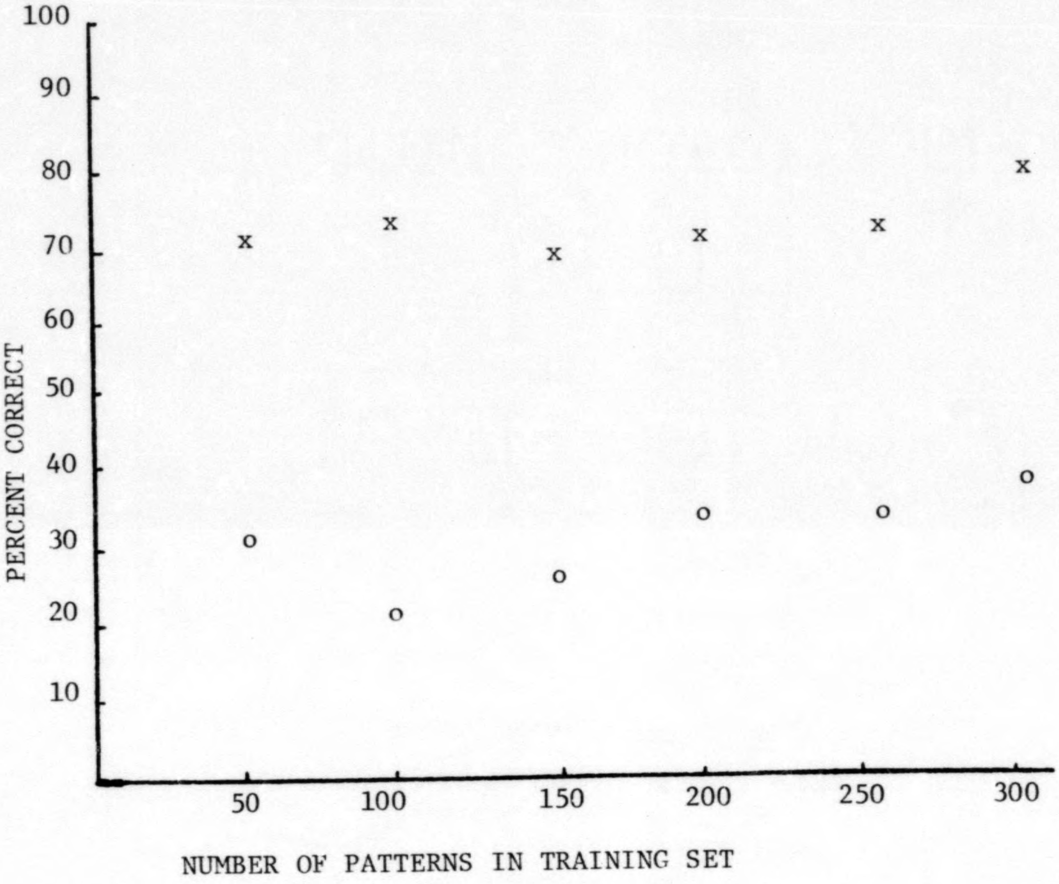

FIGURE 10. THE DEPENDENCE OF PREDICTABILITY ON TRAINING SET SIZE

Pass/Fail Correct Predictions = x
Letter Grade Correct Predictions = o

TABLE 3. DEPENDENCE OF PREDICTABILITY ON THE FUNCTION OF THE EUCLIDEAN DISTANCE.

P(D)=	% CORRECT PREDICTION OF PASS/FAIL
D^9	42.22
D^4	44.44
D	46.67
D^{-1}	60.00
D^{-4}	60.00
D^{-9}	71.11
D^{-12}	73.33
D^{-15}	77.78
D^{-20}	77.78

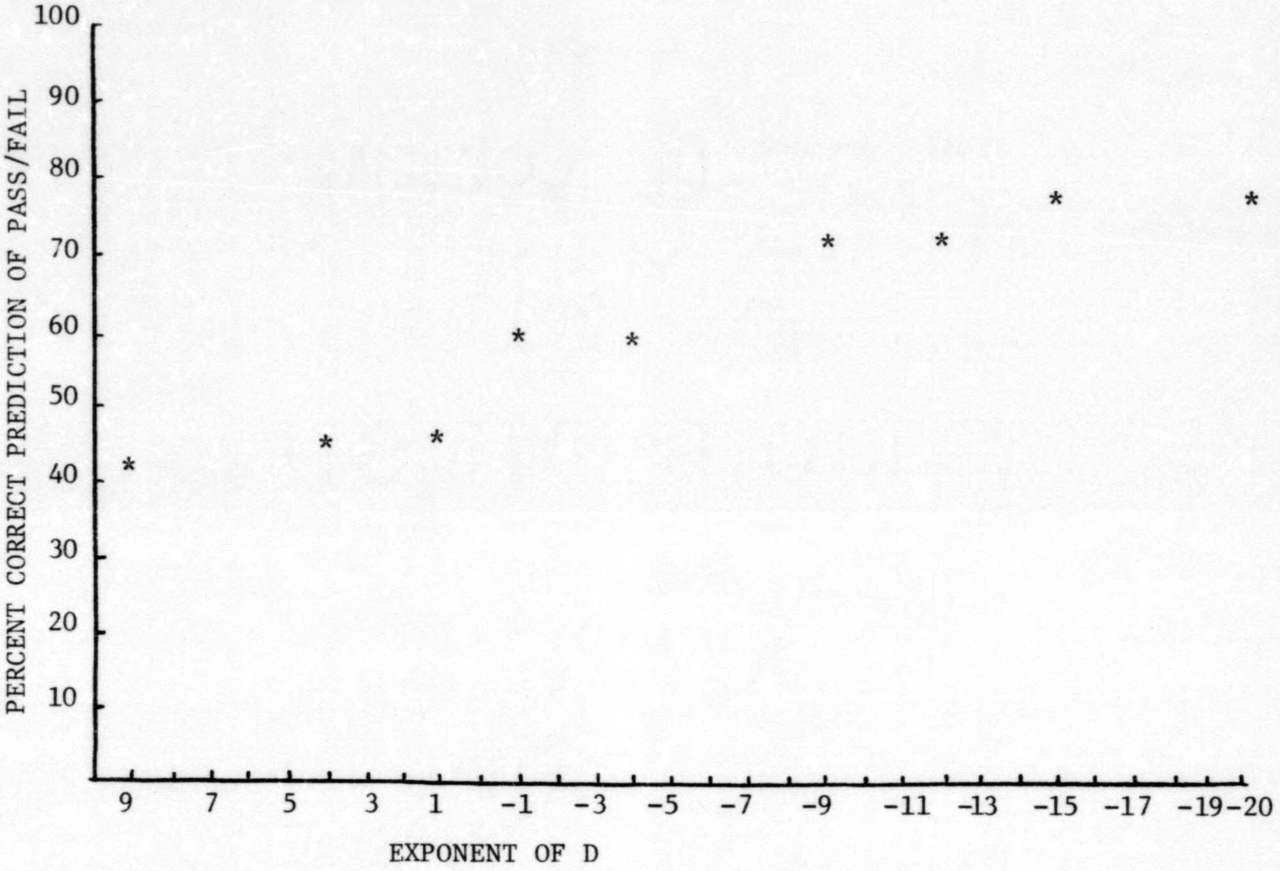

FIGURE 11. DEPENDENCE OF PREDICTABILITY ON THE FUNCTION OF THE EUCLIDEAN DISTANCE (D).

TABLE 4. THE DEPENDENCE OF PREDICTABILITY ON TRAINING SET SIZE

NUMBER OF PATTERNS IN TRAINING SET	% CORRECT PREDICTION OF PASS/FAIL	
	PFC	KNN
50	88.88	51.12
100	88.88	62.23
150	88.88	75.56
200	86.67	77.78
250	91.11	86.67
300	88.88	82.23

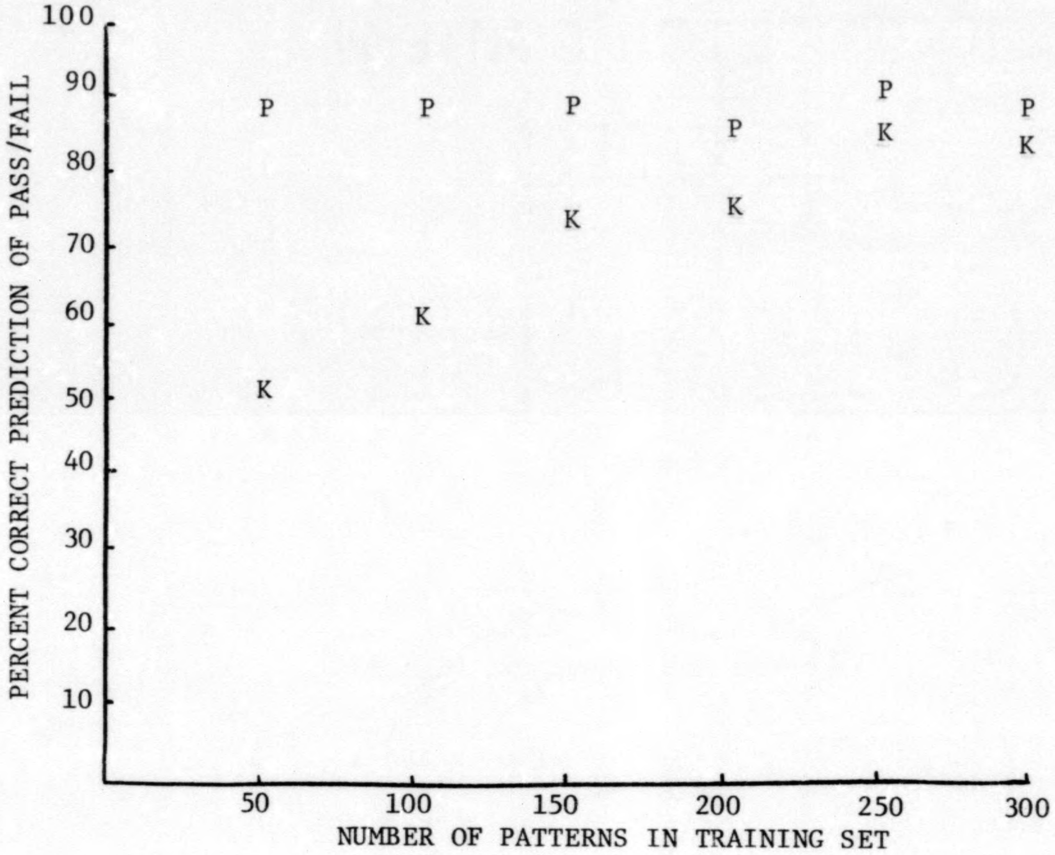

FIGURE 12. THE DEPENDENCE OF PREDICTABILITY ON TRAINING SET SIZE

Potential Function Classifier = P
K Nearest Neighbor Classifier = K

VI. CONCLUSION

Other research into the application of various pattern recognition techniques has been confined to the type of data that does not have a great deal of variation. Prior comparisons have been made on data that has been artificially generated or that has been shown to give acceptable results in all types of classifiers. The data used in this application is dependent on a great many factors and consequently there is a large amount of variation in the data in each category. It is impossible in most cases to distinguish what would be considered good data from that which would be considered bad.

On good data the KNN Classifier and the PFC have been shown to be about equal in reliability with both being much better than the Linear Learning Machine. In this application, a good example of acting on poor data, the Linear Learning Machine is not a workable classifier; the KNN and PFC Classifiers are acceptable with the PFC being somewhat better than the KNN Classifier.

As would be expected, the predictability of both the KNN and PFC Classifiers would improve with a larger training set. The value of k for the KNN Classifier optimizes because as k is increased there is first an increasing number of nearest neighbors of the correct category voting for the category of the unknown, then as k gets even larger, it begins encompassing a large portion of the training set, not just the nearby neighbors. It would be extremely difficult to find an optimum expression for P(D) for the PFC as there are a great many possibilities to consider. Also the predictability of the classifiers would be expected to improve if the number of data points in each pattern was increased.

The value of the pattern recognition methods lies in their ease of application to all kinds of data and the reliability which they can achieve. Further research must be directed in the area of feature extraction as there are many improvements to be made there, such as finding the best method of separating the categories in the pattern space. The next possible step in this direction would be the on-line use of pattern recognition to speed up analysis of data in all areas of science.

APPENDIX A

The Linear Learning Machine Program

The Linear Learning Machine program presented here was designed after a model presented by Jurs and Isenhour.[8] It employs a width parameter which can be set equal to a negative number to create a region of overlap. (D in line 22 of the program is the width parameter.) The data used by the program is taken from a previously created data file. After the training routine the computer prints out the calculated weight vector and the number of feedbacks required to achieve convergence. It then goes into the prediction routine where it prints out each pattern it misclassifies. At the end of the prediction routine the number of patterns misclassified are totaled. Should convergence not be achieved during the training routine the program will print out the word "nonconvergent" and print out the weight vector that was calculated to that point. The computer language used is BASIC PLUS.

```
8    Z=0
10   T%=345
12   DIM W(11),V(345)
14   OPEN 'DATA.ED1' AS FILE #4
16   DIM #4,P(345,11)
18   W1=.1
20   N%=9
22   D=-.001
24   R%=300
50   FOR I%=1 TO N% + 1
55   W(I%)=W1
57   IF I% +2=Z THEN W(I%)=0
60   NEXT I%
65   FOR I%=1 to R%
70   V(I%)=I%
75   NEXT I%
85   F%=0
```

```
90   U%=0
95   Y%=0
100  IF U% > 0 GOTO 110
105  U%=R%
107  H=1
110  FOR X%=1 TO U%
115  I%=V(X%)
120  P(I%,N%=3)=1
125  S=W(N%+1)
130  FOR J%=1 TO N%
132  IF J%+2=Z THEN J%=J%+1
135  S=S+P(I%,J%+2)*W(J%)
140  NEXT J%
145  IF P(I%,2) > 1 GOTO 155
150  IF S+D > 0 GOTO 170 ELSE 235
155  IF S-D > 0 GOTO 235
160  C=2*(D-S)
165  GOTO 175
170  C=2*(-D-S)
175  Q=1
180  FOR J%=1 TO N%
182  IF J%+2=Z THEN J%=J%+1
185  Q=Q+P(I%,J%+2)**2
190  NEXT J%
195  C=C/Q
200  FOR J%=1 TO N%
202  IF J%+2=Z THEN J%=J%+1
205  W(J%)=W(J%)+C*P(I%,J%+2)
210  NEXT J%
212  W(N%+1)=W(N%+1)+C
215  X3=1
216  H=0
220  IF X3=0 GOTO 235
225  Y%=Y%+1
230  V(Y%)=I%
235  NEXT X%
240  U%=Y%
242  IF U%=0 GOTO 260
245  F%=F%+1
250  IF F% > 1000 GOTO 510
255  GOTO 95
260  PRINT F%
265  F1=F1+F%
270  IF H=1 GOTO 280
275  GOTO 65
280  PRINT "TOTAL FEEDBACKS=", F1
285  FRINT "WEIGHT FACTORS"
290  MAT PRINT W
400  A%=0
405  FOR I%=1 TO T%
410  L=W(N%+1)
415  FORJ%=1 TO N%
417  IF J%+2=Z THEN J%=J%+1
420  L=L+W(J%*P(I%,J%+2)
425  NEXT J%
427  D=0
430  IF P(I%,2) > 1 GOTO 440
435  IF L+D > 0 GOTO 450 ELSE GOTO 470
440  IF L-D > 0 GOTO 470
450  PRINT P(I%,1), "MISSED",L,P(I%,2)
455  A%=A%+1
470  NEXT I%
475  PRINT "TOTAL MISCLASSIFIED=",A%
480  CLOSE #4
500 STOP
510  PRINT "NONCONVERGENT"
520  PRINT "WEIGHT FACTORS???"
530  MATPRINT W
550  STOP
600  END
```

APPENDIX B

The K Nearest Neighbor Classifier Program

This program for the K Nearest Neighbor Classifier also uses data from a previously created data file. The number of nearest neighbors is changed by changing the value of N in line 25 of the program. Whenever a pattern is misclassified its number is printed out along with how it was classified and its true class. At the end of the program a tally is given of the number of patterns missed.

```
10   E=1
15   P=300
20   Q=345
25   N=15
30   M=5
32   DIM W(345,11),S(25),T(25),C(5),D(5)
35   OPEN 'DATA.ED1' AS FILE #3
40   DIM #3, G(345,11)
41   FOR I=1 TO 345
42   FOR J=1 TO 11
43   W(I,J)=G(I,J)
44   NEXT J
45   NEXT I
46   CLOSE #3
50   U=0
51   A=0
53   FOR I=P+1 TO Q
55   FOR J=1 TO M
57   S(J)=10000
59   NEXT J
60   FOR K=E TO P
65   R=0
70   FOR J=3 TO 11
75   R=R+(W(I,J)-W(K,J))**2
80   NEXT J
83   H=K
85   FOR J=1 TO N
90   IF R > S(J) GOTO 115
92   V=S(J)
93   L=T(J)
95   S(J)=R
100  T(J)=H
```

```
105  R=V
110  H=L
115  NEXT J
120  NEXT K
121  FOR K=1 TO M
122  C(K)=0
123  NEXT K
125  FOR J=1 TO N
130  X=T(J)
135  FOR K=1 to M
140  IF W(X,2) <> K-1 GOTO 145
142  C(K)=C(K)+1/S(J)
143  D(K)=X
145  NEXT K
150  NEXT J
170  B=0
175  FOR K+1 TO M
180  IF C(K) < B GOTO 200
185  B=C(K)
190  Y=D(K)
200  NEXT K
210  IF W(Y,2)=W(I,2) GOTO 225
211  GOTO 215 IF W(Y,2) > 1 IF W(I,2) > 1
212  GOTO 215 IF W(Y,2) < 2 IF W(I,2) < 2
213  A=A+1
215  PRINT W(I,1), "MISSED",W(I,2),W(Y,2),W(Y,1)
220  U=U+1
225  NEXT I
230  PRINT "TOTAL MISSED",U,"% MISSED",U?(Q-P)
235  PRINT "# MISSED ON PASS/FAIL BASIS,A
236  PRINT "% MISSED ON PASS/FAIL BASIS, A/(Q-P)
300  STOP
400  END
```

APPENDIX C

The Potential Function Classifier Program

The Potential Function Classifier performs very similar to the way the KNN Classifier does. The scalar quantity is given in line 20 as T. The quantities W and V in lines 30 and 40 are the weight factors for the positive and negative distances respectively. The function of the distance is given in line 250 by F.

```
10    U%=300
12    X%=1
15    L%=345
20    T=0
30    W=1
40    V=2.6319
50    OPEN 'DATA.ED1' AS FILE #3
52    DIM #3, W(345,11)
55    DIM A(345,11),B(11)
60    FOR I=1 TO 345
63    FOR J=1 TO 11
65    A(I,J)=W(I,J)
67    NEXT J
70    NEXT I
75    CLOSE #3
90    E%=0
100   FOR K%=U%+1 TO L%
110   FOR I%=3 TO 11
120   B(I%)=A(K%,I%)
130   NEXT I%
170   P=0
180   M=0
190   FOR I%=X% TO U%
200   D=0
210   FOR J%=3 TO 11
220   D=D+(A(I%,J%)-B(J%))**2
230   NEXT J%
240   D=SQRT(D)
250   F=1/D**15
260   IF A(I%,2) > 1 GOTO 290
270   M=M+F*V
280   GOTO 300
290   P=P+F*W
```

```
300  NEXT I%
320  S=SGN(P-M+T)
400  IF A(K%,2) > 1 THEN R=1 ELSE R=-1
410  IF R=S GOTO 440
420  PRINT A(K%,1),R,S
430  E%=E%+1
440  NEXT K%
450  PRINT "TOTAL MISSED",E%
455  Y=L%*U%*E%
456  Z=(Y/(L%*U%))*100
460  PRINT "% CORRECT",Z
500  STOP
600  END
```

LIST OF REFERENCES

1. N.J. Nilsson, "Learning Machines," McGraw-Hill Book Co., New York, New York (1965).

2. G.S. Sebestyen, "Decision-Making Processes in Pattern Recognition," Macmillian, New York, New York (1962).

3. B.G. Batchelor, "Practical Approach to Pattern Classification," Plenum Publishing Corp., New York, New York (1974).

4. E.A. Patrick, "Fundamentals of Pattern Recognition," Prentice-Hall, Englewood Cliffs, New Jersey (1972).

5. H.C. Andrews, "Mathematical Techniques in Pattern Recognition," Wiley-Interscience, New York, New York (1972).

6. J.T. Tou and R.C. Gonzalez, "Pattern Recognition Principles," Addison-Wesley Publishing Company, Reading, Mass. (1974).

7. K.S. Fu, "Sequential Methods in Pattern Recognition and Machine Learning," Academic Press, Inc., New York, New York (1968).

8. P.C. Jurs and T.L. Isenhour, "Chemical Applications of Pattern Recognition," Wiley-Interscience, New York, New York (1975).

9. T.M. Cover and P.E. Hart, IEEE Trans. on Info. Theory, IT-13, 21 (1967).

10. C.A. Rosen, Science, 156, 38 (1967).

11. T.M. Cover, IEEE Trans. on Info. Theory, IT-14, 50 (1968).

12. D.W. Peterson, ibid., IT-16, 26 (1970).

13. P.C. Jurs, B.R. Kowalski, and T.L. Isenhour, Anal. Chem., 41(1), 21 (1969).

14. B.R. Kowalski, P.C. Jurs, T.L. Isenhour, and C.N. Reilley, Anal. Chem., 41(6), 695 (1969).

15. N.M. Frew, L.E. Wangen, and T.L. Isenhour, Pattern Recognition, 3, 281 (1971).

16. J.B. Justice and T.L. Isenhour, Anal. Chem., 46(2), 223 (1974).

17. B.R. Kowalski and C.A. Reilley, J. Phys. Chem., 75, 1402 (1971).

18. B.R. Kowalski and C.F. Bender, Anal. Chem., 44(8), 1405 (1972).

19. B.R. Kowalski, P.C. Jurs, T.L. Isenhour, and C.N. Reilley, Anal. Chem., 41(14), 1945 (1969).

20. H.B. Woodruff, S.R. Lowry, and T.L. Isenhour, Anal. Chem., 46(16), 2150 (1974).

21. P.S. Shoenfeld and J.R. DeVoe, Anal. Chem., 48(5), 403R (1976).

22. R.G. Casey and G. Nagy, Sci. Amer., 224, 56(1971).

23. L.A. Currie, J.J. Filliben, and J.R. DeVoe, Anal. Chem., 44(5), 497R (1972).

24. T.L. Isenhour and P.C. Jurs, Anal. Chem., 43(10), 20A (1971).

25. C.J.D.M. Verhagen, Pattern Recognition, 7, 109 (1975).

26. B.R. Kowalski, Anal. Chem., 47(13), 1152A (1975).

27. See reference 3, p. 43.

28. L.E. Wangen, N.M. Frew, and T.L. Isenhour, Anal. Chem., 43(7), 845 (1971).