Reprinted from

Symposium on Machine Processing of Remotely Sensed Data

June 3 - 5, 1975

The Laboratory for Applications of
Remote Sensing

Purdue University
West Lafayette
Indiana

IEEE Catalog No.
75CH1009-0 -C

CANONICAL ANALYSIS FOR INCREASED CLASSIFICATION SPEED AND CHANNEL SELECTION*

Walter Eppler

Lockheed Electronics Company
Aerospace Systems Division
Houston, Texas (77058)

I. ABSTRACT

The quadratic form can be expressed as a monotonically increasing sum of squares when the inverse covariance matrix is represented in canonical form. This formulation has the advantage that, in testing a particular class hypothesis, computations can be discontinued when the partial sum exceeds the smallest value obtained for other classes already tested. A method for channel selection is presented which arranges the original input measurements in that order which minimizes the expected number of computations. The classification algorithm was tested on data from LARS Flight Line C1 and found to reduce the sum-of-products operations by a factor of 6.7 compared to the conventional approach. In effect, the accuracy of a twelve-channel classification was achieved using only that CPU time required for a conventional four-channel classification.

II. INTRODUCTION

The well-known classification rule based on the maximum-likelihood criterion and assumed normal probability density functions involves evaluating quadratic forms in the case of M classes. Sometimes a linear transformation is performed on the original N measurements to form $\hat{N} < N$ measurements for use in evaluating the quadratic forms resulting in a reduction in computation time. The disadvantages of this approach are:

1. Additional computer time is required to perform the linear transformations.
2. It is inevitable that some (usually small but unknown) class separability is lost in the dimensionality-reduction.

The algorithm described in this paper has the advantage that it uses only as many channels as necessary to make the desired discriminations; if necessary, all channels are used and no information is sacrificed. Classification speed is achieved by using first those channels which are most important for discrimination. The paper describes how the proper order is determined; the derivation has obvious application in the general area of channel-selection. Empirical results are presented to show the optimum channel-order and resulting increase in classification speed in typical applications.

---

*This research was funded by NASA at the Johnson Space Center under Contract NAS-12200.

III. CLASSIFICATION USING THE QUADRATIC FORM

Computer-based classification systems are often based on the assumption that multivariate measurements from the classes of interest are normally distributed. In this case the conditional probability density function is given by Eq. (1)† where the mean vector

$$p_i(X) = \frac{1}{(2\pi)^{N/2}|K_i|^{1/2}} \exp\left[-\frac{1}{2}(X - M_i)^T K_i^{-1} (X - M_i)\right] \qquad (1)$$

and covariance matrix are computed from training samples. Because it plays such an important role in the derivations which follow, it is useful to define the quadratic form according to Eq. (2). The maximum likelihood decision rule given

$$Q_i(X) = (X - M_i)^T K_i^{-1}(X - M_i) \qquad (2)$$

by Eq. (3) assigns a sample to the most-likely class. Combining Eqs. (1) - (3)

$$\text{If:} \quad P_j p_j(X) > P_i p_i(X) \quad \text{all } j \neq i \quad \text{Then:} \quad X \to j \qquad (3)$$

results in the familiar decision rule given by Eq. (4) where the class-pair constant

$$\text{If:} \quad Q_j(X) < C_{ij} + Q_i(X) \quad \text{all } j \neq i \quad \text{Then:} \quad X \to j \qquad (4)$$

C_{ij} is defined by Eq. (5). In order to classify a sample into one of M classes, it is necessary to compute the quadratic form in Eq. (2) M times. The CPU

$$C_{ij} = \ell n\left[\frac{|K_i|P_j^{\,2}}{|K_j|P_i^{\,2}}\right] \qquad (5)$$

time required to compute each quadratic form is proportional to $N(N + 1)$ so that the time per sample varies as MN(N+1) for LARS-like classification algorithms. One of the objectives of this paper is to demonstrate a method which reduces significantly the CPU time without in any way changing the classification result.

IV. UNITARY CANONICAL FORM

It is well-known that a unitary matrix can be found which satisfies Eqs. (6) and (7) where σ_{ik}^2 is the *kth* eigenvalue of K_i. The term σ_{ik}^2 is necessarily positive

$$K_i = U_i^T \begin{bmatrix} \sigma_{i1}^2 & & & & \\ & \ddots & & \bigcirc & \\ & & \sigma_{ik}^2 & & \\ & \bigcirc & & \ddots & \\ & & & & \sigma_{iN}^2 \end{bmatrix} U_i \qquad (6)$$

$$U_i^T U_i = U_i U_i^T = I \qquad (7)$$

†Notation used in this abstract is defined in the Glossary of Symbols in Section X.

for all k because the covariance matrix is positive definite. This notation is convenient because it suggests that σ_{ik} is the standard deviation of the scatter along the kth principal axis. It is useful to consider the transformation defined by Eqs. (8) and (9) where V_{ik} is the kth column eigenvector of K_i. According to Eq. (9) the scalar elements y_{ik} are the projections

$$Y_i = \begin{bmatrix} y_{i1} \\ \vdots \\ y_{ik} \\ \vdots \\ y_{iN} \end{bmatrix} = U_i(X - M_i) = \begin{bmatrix} V_{i1}^T \\ \vdots \\ V_{ik}^T \\ \vdots \\ V_{iN}^T \end{bmatrix} (X - M_i) \qquad (8)$$

$$y_{ik} = V_{ik}^T (X - M_i) \qquad (9)$$

(i.e., dot products) of $(X - M_i)$ along the eigenvectors V_{ik}^T. By combining Eqs. (2), (6), (7), and (8) it is possible to express the value of the quadratic form according to Eq. (10).

$$Q_i = \sum_{k=1}^{N} y_{ik}^2/\sigma_{ik}^2 \qquad (10)$$

V. CLASSIFICATION USING THE UNITARY CANONICAL FORM

It is possible to restate the decision rule given by Eq. (4) in the form of Eq. (11) which specifies all classes i which X does <u>not</u> belong to.

$$\text{If: } Q_i(X) > Q_j(X) - C_{ij} \quad \text{Then : } X \text{ not Class } i \qquad (11)$$

Combining Eqs. (10) and (11) and regarding the summation in Eq. (10) to being carried out in two parts results in Eq. (12)

$$\text{If: } \sum_{k=1}^{n} y_{ik}^2/\sigma_{ik}^2 + \sum_{k=n+1}^{N} y_{ik}^2/\sigma_{ik}^2 > Q_j(X) - C_{ij} \quad \text{Then: } X \text{ not Class } i \qquad (12)$$

Because $\sum_{k=n+1}^{N} y_{ik}^2/\sigma_{ik}^2$ is always positive the maximum likelihood decision rule takes the following form:

$$\text{If: } \sum_{k=1}^{n} y_{ik}^2/\sigma_{ik}^2 \geq Q_j(X) - C_{ij} \quad \text{Then: } X \text{ not Class } i \qquad (13)$$

This formulation (Dye, 1974) is extremely useful because it states that computations for Class i may <u>cease</u> when the <u>partial</u> sum $\hat{Q}_i$ given by Eq. (14) exceeds the value

$$\hat{Q}_i = \sum_{k=1}^{n} y_{ik}^2 \Big/ \sigma_{ik}^2 \tag{14}$$

$Q_j(X)$, the value of the quadratic form for the correct class, minus C_{ij}. Because the values of j and $Q_j(X)$ for a given pixel are not known until all classes have been tested, $\hat{Q}_i$ is compared with $Q_\ell(X)$, the smallest value of quadratic form for <u>classes tested up to that point</u>. In other words, at any stage Class i is the candidate and Class ℓ is the current best estimate. The correct class is the value of ℓ after M classes have been tested.

This suggests the classification algorithm shown in Fig. 1. The first step is to make a class hypothesis. As the initial candidate, select that class assigned to the previous pixel; then test all other classes in order of decreasing *a priori* probabilities. For each candidate class accumulate the partial sum $\hat{Q}_i$ by successively adding $y^2_{ik} \big/ \sigma^2_{ik}$ until either:

1) After n cycles through the loop $\hat{Q}_i$ exceeds $(Q_\ell - C_{i\ell})$ in which case Class i is discarded as a candidate.

2) After N cycles through the loop $\hat{Q}_i$ is less than $(Q_\ell - C_{i\ell})$ in which case Class i replaces Class ℓ as the current best estimate.

After all classes have been tested the maximum-likelihood estimate is $j = \ell$. The advantage of the algorithm based on Eq. (13) and Fig. 1 is that it uses only as many channels as required to make the desired discriminations; if necessary, however, all N channels are used and no information is lost.

VI. SELECTING THE OPTIMUM SEQUENCE OF EIGENVECTORS

From Eqs. (13) and (14) and Fig. 1 it is apparent that CPU time is minimized by causing $\hat{Q}_i$ to increase by the largest possible increment, y_{ik}^2/σ_{ik}^2, for each value of $1 \le k \le N$. This results in discarding incorrect hypotheses at the minimum value of n.

The expected value of the k^{th} increment averaged over all X from Class j is given by Eq. (15).

$$\left\langle y_{ik}^2/\sigma_{ik}^2 \right\rangle_j = \frac{1}{\sigma_{ik}^2} \left\langle \left[V_{ik}^T (X - M_i) \right]^2 \right\rangle_j \tag{15a}$$

$$= \frac{V_{ik}^T}{\sigma_{ik}} \left\langle (X - M_i)(X - M_i)^T \right\rangle_j \frac{V_{ik}}{\sigma_{ik}} \tag{15b}$$

$$= \frac{V_{ik}^T}{\sigma_{ik}} \left[K_j + (M_j - M_i)(M_j - M_i)^T \right] \frac{V_{ik}}{\sigma_{ik}} \tag{15c}$$

For specified values of i and j it is possible to evaluate Eq. (15c) for $1 \le k \le N$ and to arrange the vectors V_{ik}^T/σ_{ik} in the order of decreasing values of $\left\langle y_{ik}^2/\sigma_{ik}^2 \right\rangle_j$. This is the order in which they should be read in Eq. (9) so that $\hat{Q}_i$, given by Eq. (14), always increases at the fastest rate.

Note that for a given <u>candidate</u> Class i, the optimum order for using the eigenvectors depends on j, the <u>correct</u> class. Because j is not known until all M classes have been tested, the value ℓ for the <u>current best estimate</u>, is used to select the prestored order in which the N eigenvectors are used.* According to Fig. 1, the M classes are tested in the following order:

1) The first class tested is the one assigned to the neighboring pixel; this hypothesis is likely to be correct due to the spatial correlation within typical scenes.
2) The remaining classes are tested in order of decreasing *a priori* probabilities.

With this approach it is likely that $\ell = j$ after only a few classes have been tested and therefore the resulting eigenvector order is best suited to discarding incorrect class hypothesis very early in the computation of $\hat{Q}_i$.

It is instructive to determine the increment added to $\hat{Q}_i$ when $i = j$; i.e., when the candidate class is the correct class. Equation (16) is

$$\left\langle y_{jk}^2/\sigma_{jk}^2 \right\rangle_j = \frac{V_{jk}^T}{\sigma_{jk}} K_j \frac{V_{jk}}{\sigma_{jk}} = 1 \text{ for all } 1 \leq k \leq N \tag{16}$$

obtained by combining Eqs. (6), (7), (8), and (15c). According to Eq. (16) $\hat{Q}_j$ increases linearly from zero to N† as shown in Fig. 2 For other candidate classes $\hat{Q}_i$ increases to $Q_i(X)$ for $k = N$. According to the decision rule given by Eq. (13), the computations for Class $i \neq j$ may cease after n terms when $\hat{Q}_i > (Q_j(X) - C_{ij})$ which for the <u>expected</u> case is when $\hat{Q}_i > (N - C_{ij})$ as shown in Fig. 2. From this figure it is apparent that the eigenvectors should be used in that specific order which causes $\hat{Q}_i$ to increase most rapidly so that it reaches $(N - C_{ij})$ for the minimum value of n.

VII. CLASSIFICATION BASED ON THE LOWER-TRIANGULAR CANONICAL FORM

It is well known (Forsythe, 1967 and Van Rooy, 1973) that the symmetric matrix K_i^{-1} can be represented in terms of the lower triangular matrix L_i according to Eq. (17).

$$K_i^{-1} = L_i^T L_i \tag{17a}$$

$$L_i = \begin{bmatrix} \ell_{i11} & & & \bigcirc \\ \ell_{i21} & \ell_{i22} & & \\ \vdots & & \ddots & \\ \ell_{iN1} & \ell_{iN2} & \cdots & \ell_{iNN} \end{bmatrix} \tag{17b}$$

Using the transformation given by Eqs. (18) and (19) makes it possible to express the quadratic form according to Eq. (20).

$$\tilde{Y}_i = \begin{bmatrix} \tilde{y}_{i1} \\ \tilde{y}_{ik} \\ \tilde{y}_{iN} \end{bmatrix} = L_i(X - M_i) = \begin{bmatrix} \tilde{V}_{i1}^T \\ \tilde{V}_{ik} \\ \tilde{V}_{iN}^T \end{bmatrix} (X - M_i) \tag{18}$$

$$\tilde{y}_{ik} = \tilde{V}_{ik}^T (X - M_i) \tag{19}$$

$$Q_i(X) = \sum_{k=1}^{N} \tilde{y}_{ik}^2 \tag{20}$$

---

*This approach is fundamentally different from the one used by Bendix (Dye, 1974) which uses only <u>one</u> eigenvector order for each candidate class.

†This agrees with the well-known result that $Q_j(X)$ has a chi-square distribution with an expected value equal to N.

These equations are exactly analogous to Eqs. (8)-(10) except that $\tilde{V}_{ik}^T$ are rows from the lower triangular matrix L_i rather than the unitary matrix U_i. As before, it is useful to define the partial sum $\tilde{Q}_i$ by Eq. (21); again it is compared with $(Q_\ell(X) - C_{i\ell})$ to determine whether Class i should be discarded as a hypothesis.

$$\tilde{Q}_i = \sum_{k=1}^{\tilde{n}} \tilde{y}_{ik}^2 \qquad (21)$$

By analogy with Eq. (15) the expected value of $\left\langle \tilde{y}_{ik}^2 \right\rangle_j$ is given by Eq. (22).

$$\left\langle \tilde{y}_{ik}^2 \right\rangle_j = \tilde{V}_{ik}^T \left[K_j + (M_j - M_i)(M_j - M_i)^T \right] \tilde{V}_{ik} \qquad (22)$$

In the case of the lower triangular canonical form it is doubly advantageous to minimize $\tilde{n}$:

1) As before, it reduces the number of terms in the summation for $\tilde{Q}_i$.

2) Because $\tilde{V}_{ik}^T$ is zero beyond the k^{th} element, the number of terms involved in computing each $\tilde{y}_{ik}$ is equal to k ; see Eq. (19). This means that less CPU time is required to compute the early elements.

By permuting the channel order in the measurement vector X (and thereby changing K_i, L_i, K_j, M_i, and M_j), it is possible to cause the values $\left\langle \tilde{y}_{ik}^2 \right\rangle_j$, given by Eq. (22), to be in descending order on k . The result is that the most important single channel is used first, the most important pair of channels is used second, etc. This causes $\tilde{Q}_i$ to increase at the fastest possible rate so that an incorrect hypothesis can be discarded after the fewest number of terms; see Fig. 2.

VIII. CHANNEL SELECTION AND CLASSIFICATION SPEED

Programs were written to accomplish channel selection and classification using the lower triangular canonical form according to Eqs. (17) - (22). These programs, called SELECT and CLASS, were written in FORTRAN and require less than 20,000 words of core storage so that they can be run on almost any computer. The system was tested using twelve-channel multispectral scanner data from LARS Flight Line C1; ground-truth data was available for nine classes with the designations given in Table 1.

The program SELECT takes card inputs giving the mean vector and covariance matrix for each class and puts out a tape giving a) the optimum channel order, b) the elements of L_i in the optimum order, c) the elements of M_i in the optimum order; these are the inputs required by CLASS. The order is optimized for each combination of i and j such that Q_i , given by Eqs. (21) and (22), increases as fast as possible with k. In addition to tape output, SELECT also produces listings giving statistical data described in the following paragraphs. To run SELECT in the case of LARS Flight Line C1 (N=12, M=9) required 30 sec. of CPU time on an IBM 360/67.

The lower-triangular canonical form is ideally suited for evaluating the usefulness of the various channels. It is apparent from Eq. (19) that $\tilde{y}_{i\tilde{n}}$, the last term included $\tilde{Q}_i$, involves only the first $\tilde{n}$ measurements in the permuted measurement vector. In other words, the process of permuting X to put $\tilde{y}_{ik}^2$ in decreasing order places the original measurements in order of decreasing usefulness for discriminating Class j from Class i. Table 2 gives the first four (out of 12) channels which are best for each of the 72 possible (i, j) - combinations, where $i \neq j$. Table 3 gives the number of (i, j) - combinations for which each channel is used at each place in the optimum order;

e.g., Channel 9 is used first for 20 (i, j) - combinations and second for another 10 combinations. Table 3 also gives the average placement for each channel. The channels in order of decreasing usefulness (averaged over all combinations) are 9, 12, 1, 10, 11, 6, 8, 2, 7, 4, 5, and 3.

Another output from the program SELECT is an estimate of $\tilde{n}$, the number of iterations (i.e., terms in $\tilde{Q}_i$) which must be used before the expected value of $\tilde{Q}_i$, $\overline{Q}_{ij}(\tilde{n})$ given by Eq. (23), exceeds $(N-C_{ij})$ as shown in Fig. 2.

$$\overline{Q}_{ij}(\tilde{n}) = \sum_{k=1}^{\tilde{n}} \left\langle \tilde{y}^2_{ik} \right\rangle_j \qquad (23)$$

Table 4 shows that for most (i,j) - combinations only one term is needed. However, for j = 6 and i = 5 (i.e., testing for Class 5 on a pixel belonging to Class 6) four terms must be used before $\overline{Q}_{ij}(\tilde{n})$, the expected value of $\tilde{Q}_i$, exceeds $(N-C_{ij})$ and the incorrect hypothesis can be discarded.

A more realistic estimate of $\tilde{n}$ is obtained by taking into account the fact that Q_j does not always equal N, but is itself a random variable having a Chi Square density function with N degrees of freedom (Eppler, 1972). The probability that exactly $\tilde{n}$ iterations are required before $\overline{Q}_{ij}(\tilde{n}) > (Q_j - C_{ij})$ is given by Eq. (24).

$$P_{ij}(\tilde{n}) = \int_{\overline{Q}_{ij}(\tilde{n}-1) + C_{ij}}^{\overline{Q}_{ij}(\tilde{n}) + C_{ij}} \chi_N(Q)dQ \qquad (24)$$

As an example, Table 5 gives $P_{97}(\tilde{n})$ from the SELECT output listing. The average number of iterations for a given (i, j) - combination is given by Eq. (25); Table 6 gives $\overline{N}_{ij}$ for all possible (i, j) - combinations.

$$\overline{N}_{ij} = \sum_{\tilde{n}=1}^{N} \tilde{n} \cdot P_{ij}(\tilde{n}) \qquad (25)$$

The number of sum-of-products which must be computed in the course of $\tilde{n}$ iterations is $[0.5\tilde{n}\,(\tilde{n} + 1) + \tilde{n}]$ so that the average number of sum-of-product operations is given by Eq. (26). Table 7 gives the average

$$\overline{O}_{ij} = \sum_{\tilde{n}=1}^{N} [0.5\tilde{n}\,(\tilde{n} + 1) + \tilde{n}]\; P_{ij}(\tilde{n}) \qquad (26)$$

number of operations for all possible (i, j) - combinations.

Tables 4 - 7 are based on theory and are derived from statistical parameters of the data and application (i.e., mean vector and covariance matrix for each class of interest). Tables 6 and 7 show that almost all (i, j) - combinations can be discriminated using only one or two terms (i.e., channels of data) in the partial sum given by Eq. (21) and this requires five or fewer sum-of-product operations per class. This is a substantial savings compared with the conventional approach which always uses all twelve channels and performs 90 sum-of-product operations per class.

The program CLASS takes as input the multispectral scanner data tape and the output tape from SELECT and processes the data according to Eqs. (13), (17) - (21), and Fig. 1 to classify specified segments of data. The program was checked out using data from LARS Flight Line C1 with results described in the following paragraphs.

The first tests were performed on data within the 23 training fields used to derive the mean vector and covariance matrix for all classes. Table 8 gives the empirical probability that, for the given training field, Hypothesis i can be discarded after $\tilde{n}$ terms in the partial sum Eq. (21). For example, the incorrect Class 4 can be discriminated from the true Class 2 on the basis of only one channel* nearly 89% of the time; use of two channels* made discrimination possible 100% of the time. Table 8 also gives the $\bar{n}_{12}$, the average number of terms required to discard Hypothesis i when the pixel belongs to Class 2. These values are in good agreement with the corresponding column (i.e., $J = 2$) of Table 6.

After classifying all of the training fields it was possible to derive the average number of iterations required in the case of certain selected (i, j) - combinations. Table 9 shows the empirical results for those cases (in which $\bar{n}_{ij} \geq 1.5$) circled in Tables 6 and 7. Comparison shows that the empirical results in Table 9 are in general agreement with, although slightly higher than, the theoretical values given in Table 6.

Next the program CLASS was used to classify 20,000 pixels of 12-channel data from LARS Flight Line C1. It was found that the probability of requiring $\tilde{n}$ terms varied according to Table 10. It shows that an incorrect hypothesis was discarded by using only one channel 54% of the time and by using two channels 75% of the time. By using the data in Table 10 it was determined that, averaged over these 20,000 pixels, the average number of channels (i.e., terms in Eq. (21)) per class was 2.8 and the average number of sum-of-product operations per class was 13.6. The CPU time on an IBM 360/67 was approximately 2 min.

Classification algorithms currently in use evaluate the quadratic form to completion for all classes; for twelve channels 90 sum-of-products operations are required. The program CLASS based on Eqs. (13), (17)-(21), reduced the computations for the 20,000 pixel test case by a factor of $90.0/13.6 = 6.7$. Equation (27) gives R_{max}, the maximum ratio of improvement for sum-of-product operations. It results when the (M-1) incorrect classes are discarded on the first iteration and all N channels are used for the correct class.

$$R_{max} = \frac{M[0.5N(N + 1) + N]}{2(M - 1) + [0.5N(N + 1) + N]} = \frac{M}{1 + \frac{4(M - 1)}{N(N + 3)}} \qquad (27)$$

For the test case ($M = 9$ and $N = 12$) the maximum ratio of improvement is 7.6, only 15% greater than the value 6.7 actually experienced for the 20,000-pixel test segment. It is apparent from Eq. (27) that the program CLASS offers the greatest improvement over the conventional approach for those applications with large M and N. These are exactly the applications for which the conventional approach requires large amounts of computer time.

Another approach employed by some investigators (Decell, 1973) is to form $\hat{N}$ measurements which are weighted linear sums of the original N channels. In order to limit the number of sum-of-product operations per class to 14 (the value obtained using the program CLASS on the test segment), the original 12 channels would have to be reduced to $\hat{N} = 4$. The disadvantages of this approach compared with using CLASS are:

1. Additional computer time is required to perform the linear transformations.

2. It is inevitable that some (usually small but unknown) class separability is lost in the dimensionality-reduction.

---

*From Table 2 it is seen that Channel 1 is used first and a combination of Channels 1 and 9 is used second.

IX. SUMMARY AND CONCLUSIONS

It has been shown that the quadratic form can be expressed as a monotonically increasing sum of squares when the inverse covariance matrix is represented in the canonical form of Eq. (6) or Eq. (17a). This formulation has the advantage that, in testing a particular class hypothesis, computations can be discontinued when the partial sum in Eq. (14) or Eq. (21) exceeds the minimum value obtained for other classes already tested. The lower-triangular canonical form given by Eq. (17a) was selected for detailed investigation over the unitary form in Eq. (6) because a) it requires less computation and b) retains the identity of the original channels.

Using Eq. (22) it is possible to arrange the original input measurements in that order which minimizes the expected number of computations; results for a particular test case are presented in Tables 2 and 3. Also using Eq. (22) it is possible to compute the expected number of computations required to discriminate one class from another with results presented in Tables 4-7. The classification algorithm was tested on a 20,000-pixel segment of LARS Flight Line C1 and found to reduce the sum-of-products operations by a factor of 6.7 compared with the conventional approach. In effect, the accuracy of twelve channel classification was achieved using only that CPU time required for a conventional four-channel classification.

The author is pleased to acknowledge the assistance of K. Baker and A. H. Feiveson of NASA/JSC, R. H. Dye and R. H. Johnson of the Bendix Corporation, and E. L. Wilson and C. A. Helmke of Lockheed Electronics Co. who contributed in various ways to the formulation and preparation of this paper.

X. GLOSSARY OF SYMBOLS

Symbol	Meaning
N	Number of observations (i.e., channels of multispectral scanner data) available for each pixel.
X	N-dimensional column vector of observed values.
p_i	Conditional probability density function for Class i .
M_i	Mean vector for Class i computed from training samples.
K_i	Covariance matrix for Class i computed from training samples.
Q_i	Quadratic form variable defined by Eq. (2).
P_i	*A priori* probability for Class i .
C_{ij}	Class-pair constant defined by Eq. (5).
M	Number of classes for a given application.
U_i	Unitary transformation matrix defined by Eqs. (6) and (7).
σ_{ik}	Square root of the *kth* eigenvalue of K_i . It is interpreted to be the standard deviation in the direction parallel to the *kth* eigenvector.
I	Identity matrix.
Y_i	N-dimensional column vector defined by Eq. (8).
y_{ik}	The *kth* element of Y_i .
V_{ik}	The *kth* column eigenvector of K_i .
$\hat{Q}_i$	Partial sum obtained using the unitary canonical form; it is defined by Eq. (14).

Symbol	Meaning
n	The number of terms which must be included in $\hat{Q}_i$ before Class i can be discarded as a candidate.
j	Designation of the correct class.
ℓ	Designation of the current best estimate of the correct class.
i	Designation of the candidate class being tested.
L_i	Lower triangular matrix defined by Eq. (17).
$\tilde{Y}_i$	N-dimensional column vector defined by Eq. (18).
$\tilde{y}_{ik}$	The *kth* element of $\tilde{Y}_i$.
V_{ik}	The *kth* column of L_i^T .
$\tilde{Q}_i$	Partial sum obtained using the lower triangular canonical form; it is defined by Eq. (21).
$\tilde{n}$	The number of terms which must be included in $\tilde{Q}_i$ before Class i can be discarded as a candidate.
$\hat{N}$	The number of linear combinations (of the original N measurements) used for classification with the dimensionality-reduction approach.
T	Threshold value used to establish the null class.
$P_{ij}(\tilde{n})$	Probability that $\tilde{n}$ iterations are required discard Hypothesis i when the pixel belongs to Class j .
χ_N	Chi Square density functions for N degrees of freedom.
$\overline{Q}_{ij}(\tilde{n})$	Expected value of $\tilde{Q}_i$ when pixel is from Class j and $\tilde{n}$ terms are included; see Eqs. (21) - (23).
$\overline{N}_{ij}$	Average number of terms in Eq. (21) required to discard Hypothesis i when pixel belongs to Class j .
$\overline{O}_{ij}$	Average number of sum-of-products which must be computed to discard Hypothesis i when pixel belongs to Class j .

XI. REFERENCES

H. P. Decell, Jr., J. A. Quirein, "An Iterative Approach to The Feature Selection Problem," Machine Processing of Remotely Sensed Data, October 1973, Laboratory for Applications of Remote Sensing, Purdue University.

R. H. Dye, "Multivariate Categorical Analysis - Bendix Style," BSR 4149, June 1974, Bendix Corporation, Ann Arbor, Michigan.

W. G. Eppler, "Empirical Distribution of Quadratic Form Used for Thresholding," LEC/HASD No. FSD-0001, Nov. 1972, Lockheed Electronics Co., Houston, Texas.

G. Forsythe, C. B. Moler, Computer Solution of Linear Algebraic Systems, Prentice Hall, Englewood Cliffs, N.J. 1967.

D. L. Van Rooy, M. S. Lynn, C. H. Snyder, "The Use of The Modified Cholesky Decomposition in Divergence And Classification Calculations," Machine Processing of Remotely Sensed Data, October 1973, Laboratory for Applications of Remote Sensing, Purdue University.

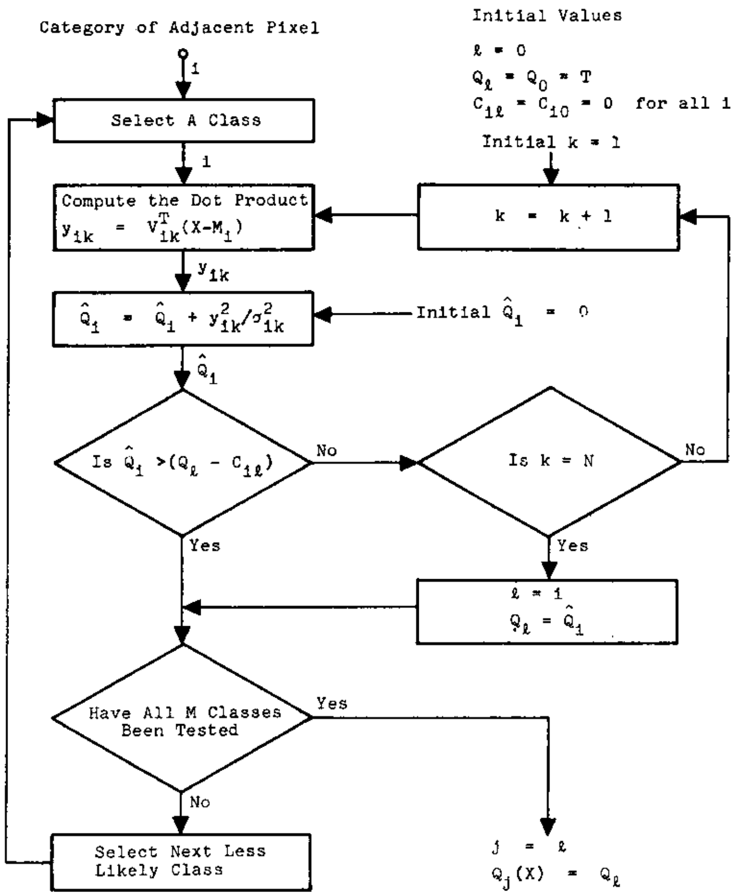

Fig. 1: Classification Algorithm Based on Eq. (13).

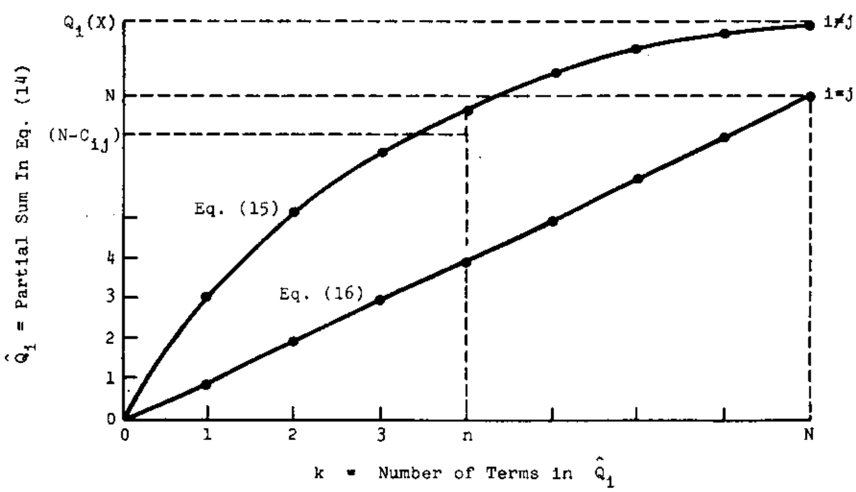

Fig. 2: Dependence of the Expected Value of $\hat{Q}_i$ On The Number of Terms, k .

Table 1: Numerical Designation for Ground Truth Classes on LARS Flight Line C1.

Class Number	1	2	3	4	5	6	7	8	9
Class Type	Soybean	Corn	Oats	Wheat I	Red Clover	Alfalfa	Rye	Bare Soil	Wheat II

Table 2: First Four Channels in Order Which Minimizes the Number of Iterations Required to Discard an Incorrect Hypothesis.

	N=12								
I \ J	1	2	3	4	5	6	7	8	9
1	------	11,10,1,12	1,10,12,6	1,9,2,10	12,6,11,9	12,6,11,2	10,1,8,2	9,12,6,1	10,1,8,2
2	10,6,9,7	---------	10,6,12,8	10,6,9,7	12,6,10,7	12,10,9,6	10,1,8,2	9,6,8,7	10,1,8,2
3	1,7,11,8	10,4,12,1	-------	11,10,12,2	11,12,10,8	11,12,10,8	8,10,1,7	12,1,7,11	8,12,1,11
4	1,9,6,8	1,9,6,11	11,9,6,12	-------	11,8,12,9	11,9,12,8	6,10,7,9	1,12,10,6	6,12,10,7
5	9,1,12,2	1,10,6,2	9,2,10,6	9,2,10,6	-------	1,9,6,10	9,2,8,1	9,6,1,7	9,2,10,6
6	12,6,1,9	1,10,12,6	9,6,2,10	9,2,6,10	6,11,1,10	---------	9,2,11,8	9,6,1,7	10,2,9,11
7	8,1,12,7	10,1,8,2	6,12,9,7	6,9,7,10	12,11,7,2	12,11,7,1	--------	12,7,1,8	10,2,12,6
8	9,12,11,8	9,12,11,10	12,9,11,1	1,10,5,12	12,9,11,8	12,9,11,8	12,11,1,9	-------	10,1,8,11
9	9,1,11,8	9,11,1,7	9,12,11,8	1,3,12,6	9,12,11,8	9,11,7,12	9,12,4,8	1,5,2,7	-------

Table 3: Number of Cases (Out of 72) Each Channel is Used on the K^{th} Iteration; Also Given is the Average Placement.

Channel Number	PLACEMENT												Average Placement
	1	2	3	4	5	6	7	8	9	10	11	12	
1	12	10	10	5	5	3	6	6	5	2	4	4	5.17
2	0	8	3	10	8	11	7	4	9	2	3	7	6.53
3	[illegible]	1	0	0	2	2	4	3	9	10	24	17	10.07
4	0	1	1	0	6	12	8	13	7	9	12	3	8.17
5	[illegible]	1	1	0	4	6	9	19	12	11	5	4	8.29
6	5	11	7	9	6	4	7	6	3	7	2	5	5.75
7		2	6	12	7	9	7	5	4	5	6	9	7.06
8	3	1	8	15	10	11	7	2	4	6	1	4	5.90
9	20	10	5	5	4	3	3	3	6	6	4	3	4.89
10	[illegible]	9	8	7	6	4	4	3	5	6	5	3	5.33
11	7	6	12	5	10	3	7	4	5	5	3	5	5.72
12	13	12	11	4	4	4	3	4	3	3	3	8	5.13

Table 4: The Number of Iterations Required to Discard an Incorrect Hypothesis. This Estimate is Based on Approach Defined by Fig 2.

	N=12								
I \ J	1	2	3	4	5	6	7	8	9
1	--	2	1	1	1	1	1	1	1
2	2	--	1	1	1	1	1	1	1
3	2	1	--	1	1	1	1	1	1
4	1	1	2	--	1	1	1	1	1
5	1	1	1	1	--	4	1	1	1
6	1	1	1	1	2	--	1	1	1
7	1	1	1	1	1	1	--	1	1
8	1	1	1	1	1	1	1	--	1
9	1	1	1	1	1	1	3	1	--

Table 5: Probability That $\tilde{n}$ Terms are Required to Discard Hypothesis 9 When Pixel Belongs to Class 7.

$\tilde{n}$	1	2	3	4	5	6	7	8	9	10	11	12
P_{97}	0.13	0.40	0.26	0.07	0.05	0.02	0.02	0.02	0.01	0.01	0.01	0.00

Table 6: The Expected Number of <u>Iterations</u> Required to Discard an Incorrect Hypothesis. This Estimate Based on the Assumption that Q_j is Distributed According to Chi Square.

	N=12								
I \ J	1	2	3	4	5	6	7	8	9
1	---	2.14	1.43	1.01	1.00	1.00	1.03	1.19	1.00
2	1.65	---	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3	1.65	1.29	---	1.40	1.01	1.00	1.17	1.00	1.00
4	1.05	1.11	1.48	---	1.00	1.00	1.13	1.00	1.00
5	1.00	1.00	1.00	1.00	---	3.36	1.00	1.00	1.00
6	1.01	1.17	1.02	1.00	1.86	---	1.00	1.00	1.00
7	1.01	1.00	1.01	1.00	1.00	1.00	---	1.00	1.03
8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	---	1.00
9	1.22	1.03	1.33	1.79	1.00	1.00	2.88	1.08	---

Table 7: The Expected Number of Sum-of-Products Required to Discard an Incorrect Hypothesis. This Estimate is Based on the Assumption that Q_j is Distributed According to Chi Square.

	N=12								
I \ J	1	2	3	4	5	6	7	8	9
1	---	5.87	3.30	2.04	2.00	2.00	2.08	2.57	2.00
2	4.35	---	2.00	2.00	2.00	2.00	2.00	2.00	2.00
3	3.98	2.87	---	3.36	2.04	2.00	2.58	2.00	2.00
4	2.15	2.33	3.45	---	2.00	2.00	2.38	2.00	2.00
5	2.00	2.00	2.00	2.00	---	12.33	2.00	2.00	2.00
6	2.03	2.50	2.06	2.00	4.71	---	2.00	2.00	2.00
7	2.04	2.00	2.02	2.00	2.00	2.00	---	2.00	2.10
8	2.00	2.00	2.00	2.00	2.00	2.00	2.00	---	2.00
9	2.67	2.10	2.99	5.88	2.00	2.00	10.23	2.23	---

Table 8: Experimental Probability That $\tilde{n}$ Iterations are Required to Discard Hypothesis i for Training Samples Belonging to Class 2. Also Given is $\bar{n}$, the Average Number of Iterations.

Correct Class: J = 2 Number of Channels: N = 12 Number of Pixels = 483													
Class Hypothesis i	Number of Iterations, $\tilde{n}$												Average
	1	2	3	4	5	6	7	8	9	10	11	12	
1	0.164	0.493	0.149	0.052	0.037	0.014	0.008	0.014	0.010	0.021	0.004	0.033	2.994
2	0.0	0.0	0.0	0.0	0.002	0.002	0.002	0.006	0.006	0.012	0.010	0.959	11.884
3	0.712	0.280	0.006	0.002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.298
4	0.888	0.112	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.112
5	0.971	0.029	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.029
6	0.812	0.176	0.010	0.002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.203
7	0.996	0.004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.004
8	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000
9	0.934	0.062	0.004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.070

Table 9: Average Number of Iterations Required for Selected Cases Circled in Tables 6 and 7.

J	1	1	2	4	5	6	7
I	2	3	1	9	6	5	9
Number of Training Samples	1524	1524	1520	1534	1539	891	1247
Average Number of Iterations	2.40	1.80	2.88	2.41	3.11	3.84	3.70

Table 10: Probability That $\tilde{n}$ Terms are Used in Eq. (21) for the Case of a 20,000-Pixel Segment of LARS Flight Line C1.

$\tilde{n}$	1	2	3	4	5	6	7	8	9	10	11	12
$P(\tilde{n})$	0.543	0.204	0.070	0.025	0.019	0.014	0.012	0.009	0.008	0.006	0.006	0.085