

*File with
N76-27639*

NASA CR#
147790

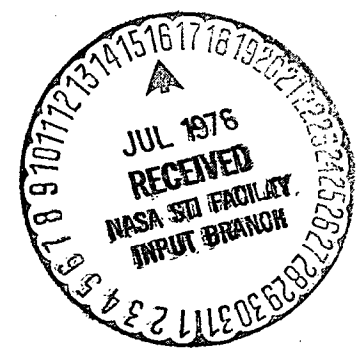
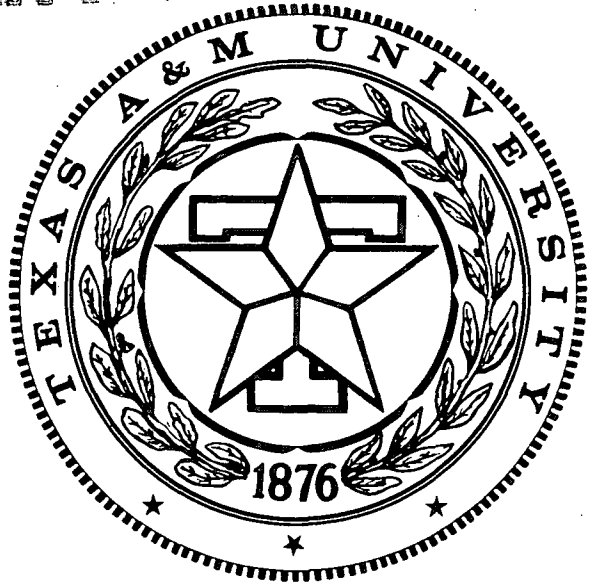
FINAL REPORT

APPLICATIONS OF FEATURE
SELECTION

Contract NAS-9-14689-4S

June 1, 1975 - May 31, 1976

REPRODUCIBLE COPY
(FACILITY CASSETTE COPY)



DEPARTMENT OF MATHEMATICS

TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS

FINAL REPORT
APPLICATIONS OF FEATURE
SELECTION
Contract NAS-9-14689-4S
June 1, 1975 - May 31, 1976

Prepared For:

Earth Observations Division
NASA/Johnson Space Center
Houston, Texas 77058

by

L.F. Guseman, Jr.
Principal Investigator
Department of Mathematics
Texas A&M University
College Station, Texas 77843

ACKNOWLEDGMENTS

The investigations discussed herein were carried out for the Earth Observations Division, NASA/Johnson Space Center, Houston, Texas, under the Contract NAS-9-14689 to the Texas A&M Research Foundation, College Station, Texas 77843 during the period June 1, 1975 to May 31, 1976. The work was coordinated through several seminar-type discussions with key personnel from NASA/EOD, especially Kenneth Baker, Dr. Alan Feiveson and Dr. B. Charles Peters, Jr. The various tasks were pursued in conjunction with Professors Jack D. Bryant and Jay R. Walton of Texas A&M University. Graduate Assistants Bruce P. Marion, David Nobles, and Manot Swasdee provided the necessary programming support for the tasks.

L. F. Guseman, Jr.

L.F. Guseman, Jr.

Principal Investigator

APPLICATIONS OF FEATURE SELECTION

1. INTRODUCTION

A practical application of remote sensing which is of considerable interest is the use of satellite-acquired (LANDSAT) multispectral scanner (MSS) data to conduct an inventory of some crop of economic interest such as wheat over a large geographical area. Any such inventory requires the development of accurate and efficient algorithms for data classification. The use of multitemporal measurements (several registered passes during the growing season) increases the dimension of the measurement space and thereby increases the computational load for a classification algorithm. When statistical pattern recognition techniques are used in classification algorithms, one method for reducing the dimensionality of the problem is by the use of feature selection/combination techniques.

Theoretical results pertaining to minimizing the probability of misclassification for linear feature selection were initially obtained by Guseman and Walker [1], [2]. A computational procedure was developed (see [1], [3]), for the case of two n -dimensional multivariate normal populations with equal a priori probabilities and a one-dimensional feature space. Theoretical results for the general case of m n -dimensional multivariate normal populations with arbitrary a priori probabilities and a k -dimensional feature space ($k < n$) appear in [6]. Development of a computational procedure for the special case $k = 1$, based on the results obtained in [6], was initiated in December

of 1973, and completed January 1975 under a previous contract (see [4]). During this same period, a preliminary investigation was initiated which involved the application of feature selection to estimation of proportions (see [5]).

Additional feature selection algorithms based on other criteria (e.g. divergence, Bhattacharyya, etc.) allowing transformation from n to k dimensions ($k > 1$) have been developed at the University of Houston.

Investigations carried out under this contract were concerned with

Extending the feature selection procedure of [4] to the case where the density function for each population is a convex combination of multivariate normals.

Application of the extended feature selection procedure to the problem of estimating the proportions of "WHEAT" and "NON-WHEAT" in a given sample segment.

Application of feature selection to color display of multi-channel images.

Each of these investigations is discussed in turn in the sequel.

2. EXTENDING FEATURE SELECTION PROCEDURE

The feature selection program (LFSPMC) was extended to treat the case where the density function for each population is a convex combination of multivariate normals. Additional modifications were made to the program to allow for simpler data set-up and program operation. The feature selection method formulation, associated computational technique, and users guide for the program are contained in

L.F. Guseman, Jr., and Bruce P. Marion, LFSPMC (Version 2): Linear Feature Selection Program Using the Probability of Misclassification, NASA Contract NAS-9-14689-4S, Texas A&M University, Department of Mathematics, Report #6, March 1976.

The new version of LFSPMC is capable of working with several populations, each of whose density functions is a convex combination of multivariate normals.

Additions to the program include internal computation of the program parameters which set limits on the number of iterations through the optimization algorithm and provide the initial guess for the minimum of the objective function.

The computational procedure for providing the starting vectors to the optimization algorithm does not appear to work well when several component classes are combined into one single convex combination. This situation is still being investigated.

3. ESTIMATION OF PROPORTIONS

Investigations were carried out which led to the formulation and implementation of computational procedures which treat the following problem:

Given LANDSAT data (multitemporal) over a LACIE sample segment, and representative samples of the classes present in the segment, estimate what proportion of the sample segment is in WHEAT.

The approach to the above problem taken in the investigations can be summarized as follows:

- (a) The density function for each of the classes WHEAT and NON-WHEAT is expressed as an appropriate convex combination of multivariate normal densities with known means and covariance matrices (usually obtained from a training sample).
- (b) Feature selection is performed (using LFSPMC (Version 2)) to produce a corresponding optimal one-dimensional Bayes classifier for WHEAT VS. NON-WHEAT whose associated confusion matrix is computable.
- (c) Using the one-dimensional classifier and associated confusion matrix, an estimate of the true proportion of WHEAT in the sample segment is made.

The mathematical formulation of the estimation of proportions procedures, along with numerical results appear in

L.F. Guseman, Jr. and Jay R. Walton, Methods for estimating proportions of convex combinations of normals, NASA Contract NAS-9-14689-4S, Texas A&M University, Department of Mathematics, Report #7, April, 1976.

The methods presented in the above report have the advantage that classification of multitemporal data over a sample segment is performed optimally in one-dimensional space thereby reducing significantly the computer time spent in classification. Another advantage is that the confusion matrix, used in obtaining unbiased estimates of the WHEAT proportions, can be readily computed without the costly requirement of obtaining a labeled sample independent of the training sample.

The estimation of proportions procedures were implemented (Program ESTPRO), and tested using MSS measurements (four registered passes) from Hill County Montana. The preliminary results appear to be quite good. In one case (ESTIMATOR 4), two unbiased estimates are available depending on the choice of sign. At present, there is no automatic rule which dictates the choice of sign. Instructions for use of ESTPRO appear in

L.F. Guseman, Jr., Bruce P. Marion and Manot Swasdee, Users Guide - ESTPRO: Estimation of proportions program using feature selection, NASA Contract NAS-9-14689-4S, Texas A&M University, Department of Mathematics, Report #8, May, 1976.

An additional investigation into the problem of formulating minimum variance unbiased estimators was performed. Preliminary results were obtained for the special case of two multivariate normal populations.

Results of this investigation appear in

Jay.R. Walton, Observations on minimum variance proportion estimation, NASA Contract NAS-9-14689-1S, Texas A&M University, Department of Mathematics, Report #5, February, 1976.

Preliminary numerical results based on results in this report indicate a sensitivity to the choice of variance-like function being minimized, as well as a dependence on the sample being used in the proportion estimation portion. In addition, no results are yet available which indicate how one might extend the above work to the "WHEAT" vs "NON-WHEAT" problem.

4. COLOR DISPLAY OF MULTICHANNEL IMAGES

A common problem for an image analyst-photo interpreter (AI) is the analysis of multichannel images of high dimensionality. For example, satellite acquired multispectral scanner data from several temporal passes may be twelve or more dimensional. Current color display techniques allow split-screen displays of up to twelve channels (for up to 16-dimensional four pass LANDSAT data). However, most AI work is restricted to one pass data owing to interpretation, display and data management problems.

For many applications, one pass is enough. However, recognition of crops (for example, for the ultimate purpose of a large area crop inventory) by statistical pattern recognition techniques requires more than one pass data to achieve acceptable performance. If these techniques are applied in the original space (of, say, twelve dimensions), computational problems become severe. Feature selection techniques furnish methods of reducing the dimensionality of feature space which preserve (in some sense) data separation, making computationally feasible refined pattern recognition techniques.

Of course, an AI is not a computer. The AI can

- (i) make use of subtle spatial relationships to recognize fields which a computer would "see" as all boundary (for instance, the long narrow fields of crops alternating with fallow in dry areas)
- (ii) pass over large areas (urban, forest) at a glance
- (iii) adjust subjectively for differing soil types, agricultural practices, sun angle (signature extension)

All these tasks are difficult for the computer. By making use of his special abilities, an AI using one pass data will compete favorably with

the best automatic pattern recognition techniques. Even so, it seems that if multitemporal data can be reduced in size and displayed like one pass data in a predictable form, then the accuracy of the AI product must increase.

The reports

Jack D. Bryant, David Nobles and Manot Swasdee, Computer Program Documentation: Program NONLN1 and NONLN2 - Nonlinear Color Display Program, NASA Contract NAS-9-14689-4S, Texas A&M University, Department of Mathematics, Report #9, May, 1976

Jack D. Bryant, David Nobles and Manot Swasdee, Computer Program Documentation: Program ROTAT1 and ROTAT2 - Rotation To Produce Color Displays, NASA Contract NAS-9-14689-4S, Texas A&M University, Department of Mathematics, Report # 10, May, 1976

discuss first efforts to accomplish a reduction of 12 dimensional data to 4 dimensional data in a form which can be displayed in the same manner as one pass data is currently being displayed. The detailed description of our methods appear there. Here we discuss in a general way the idea of generating one pass LANDSAT-like data from the output of a linear feature selection program mapping multitemporal LANDSAT data feature selection program to 4 dimensions.

Several problems exist which make transformed (by a linear feature selection procedure) data unlike one pass LANDSAT data.

1. The transformed data consists of a four vector of real numbers. How can the transformed data be displayed at all on a device which expects input data to be 6 or 7 bit (unsigned) integers (that is, fixed point numbers in the range 0-63 or 0-127)?

2. The transformed data has a much larger range than agricultural information occupies. How can the data be quantized without losing agricultural information?
3. After transformation, the data has no intrinsic meaning. That is, feature selection programs produce a transformation which optimizes some measure of separation. Obviously the transformation is not unique. If problems 1. and 2. are solved somehow, how can colors be assigned to the scaled data so that comparable colors result when different feature selection transformations are applied?

Problems 1. and 2. are easily solved. The idea is to find what happens to the mean vectors of the classes the feature selection program was trained on. We allow some variation, and define two 4-hyperrectangles: one, which is used to define a map into a unit 4-cube, has its corners defined by the maximum over all classes of means plus standard deviation σ (square root of diagonal element in covariance matrix for that class and transformed channel) and minimum of means minus σ ; and another, which is used to test for data with probably no agricultural significance by corners maximum over all classes of mean plus 3 σ and minimum of mean minus 3 σ . Figure 1 (see following page) plots the image in a unit 3-cube of transformed means we found in test data generated from Hill County North data using the UH Feature Selection Program. (Of course, there are four dimensions in actual transformed data.)

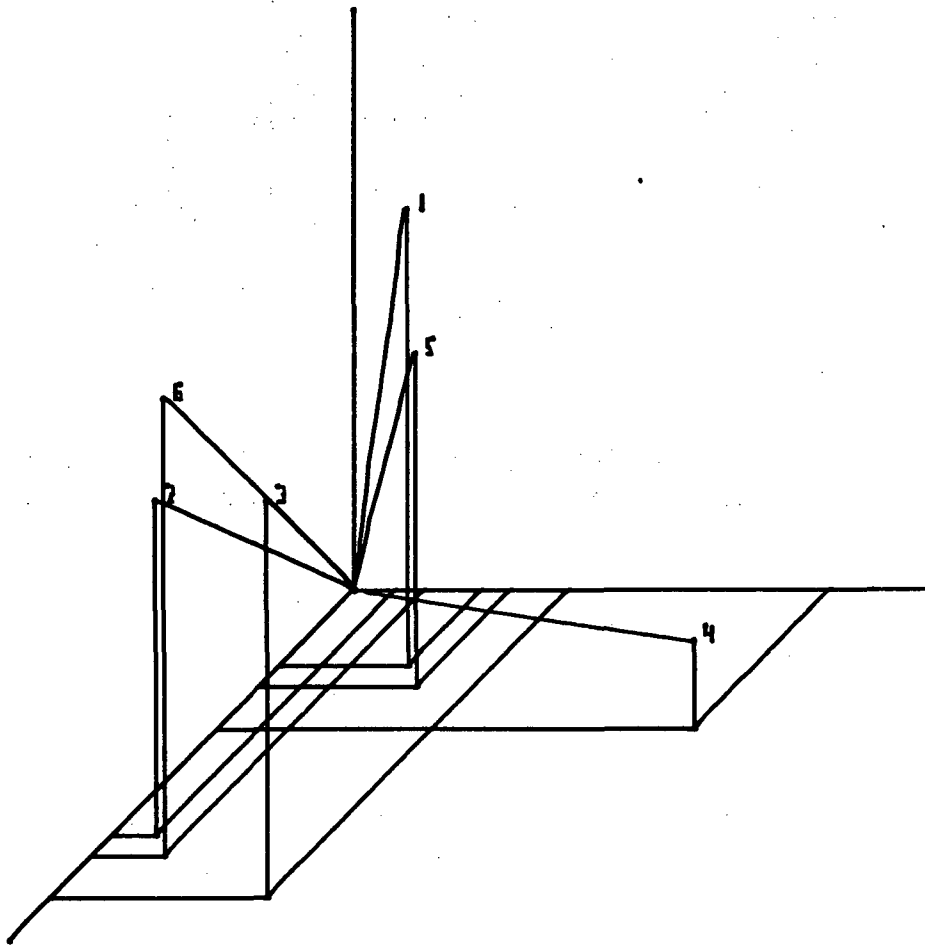


Figure 1 Six transformed means in channels 1, 2, 4

This data could now be scaled and displayed. However, it would be little help to an AI unless there were some way to predict which color represented which class. (In the current example, 1 is GRASS and 5 is WHEAT.) This is the point of problem 3: how can colors be assigned consistently? We motivate our choices as follows:

- a. In all scenes which we have viewed, at least one class which we have viewed is GRASS.
- b. There will usually be one class in which there is more interest than in others: for instance, WHEAT.
- c. The colors red and green seem to carry more subjective information than blue. Hence we want to make the class we have most interest in (WHEAT) red and another class which is nearly always present (GRASS) green.

(The statement in c. is misleading. We are not trying to produce a classification map; we only produce an enhanced display.)

The two methods we investigate (and which are described in detail in the above reports) are quite different but accomplish somewhat the same thing. One starts out being a composition of rotations which ends up with WHEAT along the "red" axis and GRASS in the "red-green" plane; some nonlinear scaling is then applied to improve this result. The other starts out being highly nonlinear (and somewhat noise-susceptible), but is much more efficient in computer time usage. Both do well making WHEAT red; we think ROTAT1-2 (the rotation-based method) is best.

REFERENCES

1. L.F. Guseman, Jr. and Homer F. Walker, On minimizing the probability of misclassification for linear feature selection, JSC Internal Technical Note, JSC-08412, NASA/Johnson Space Center, Houston, Texas, August, 1973.
2. L.F. Guseman, Jr. and Homer F. Walker, The differentiability of the probability of misclassification as a function of a linear feature selection matrix, Report #28, NAS-9-12777, Department of Mathematics, University of Houston, August, 1973.
3. L.F. Guseman, Jr. and Homer F. Walker, On minimizing the probability of misclassification for linear feature selection: A computational procedure, The Search For Oil: Some Statistical Methods and Techniques, D.B. Owen Ed., Marcel Dekker, Inc. New York, 1975.
4. L.F. Guseman, Jr. and Bruce P. Marion, LFSPMC: Linear feature selection program using the probability of misclassification, NASA Contract NAS-9-13894-1S, Texas A&M University, Department of Mathematics, Report #3, January, 1975.
5. L.F. Guseman, Jr. and Bruce P. Marion, A method for estimating proportions, NASA Contract NAS-9-13894-1S, Texas A&M University, Department of Mathematics, Report #4, April, 1975.
6. L.F. Guseman, Jr., B. Charles Peters, Jr., and Homer F. Walker, On minimizing the probability of misclassification for linear feature selection, *Ann. Statist.* 3(1975), 661-668.

LFSPMC (VERSION 2)
LINEAR FEATURE SELECTION PROGRAM USING
THE PROBABILITY OF MISCLASSIFICATION

By

L. F. Guseman, Jr. and Bruce P. Marion

Department of Mathematics
Texas A&M University

Report #6

Prepared For

Earth Observations Division
NASA/Johnson Space Center
Houston, Texas
Contract NAS-9-14689-4S

March, 1976

TABLE OF CONTENTS

Section		Page
1.0	INTRODUCTION	1
2.0	MATHEMATICAL PRELIMINARIES	3
3.0	COMPUTATIONAL PROCEDURE	8
	3.1 Parameter Initialization	8
	3.2 Initial Vector Determination	11
	3.3 Optimization Algorithm	14
	3.4 Computation of $g(B)$ and $\frac{\partial g}{\partial B}$	15
4.0	OPERATING PROCEDURES	22
5.0	EXAMPLE INPUT AND OUTPUT	25
	REFERENCES	47

1.0 INTRODUCTION

The problem of classification arises when an observer must determine the class of an object by investigating a set of measurements or features taken from the object. It is assumed that the object belongs to one of a finite number of classes (e.g. crops) and that each class is described by a probability distribution of its measurement vectors. It is also assumed that combinations of classes can be defined as convex combinations of the associated density functions. When the dimension of the measurement vector is high and a large number of objects are to be classified the computational load increases significantly. As a result, one employs feature selection techniques which allow classification in spaces of lower dimension while preserving as much as possible the discriminatory power inherently available in the original measurements.

In the sequel we discuss the computational procedure and associated computer program for a linear feature selection technique. The technique assumes:

1. A finite number, m , of convex combinations of classes
2. Each class is described by an n -dimensional multivariate normal density function of its measurement vectors.
3. The mean vector and covariance matrix for each density function are known (or can be estimated).
4. The a priori probability for each class and for each convex combination of classes is known.

The technique produces a single linear combination of the original measurements which minimizes the one-dimensional probability of misclassification defined by the transformed densities. The procedure for two classes with equal a priori probabilities was developed in [3]. Subsequent theoretical results from [4] and the original version of LFSPMC discussed in [6] form the basis for the procedure described herein. The computational procedure and a description of the associated computer program appear in Section 3.0. Procedures for using the program appear in Section 4.0. Section 5.0 contains example input and output.

2.0 MATHEMATICAL PRELIMINARIES

Let $\Pi_1, \Pi_2, \dots, \Pi_L$ be distinct classes (e.g. crops of interest) with known a priori probabilities $\alpha_1, \alpha_2, \dots, \alpha_L$, respectively. Let $x = (x_1, x_2, \dots, x_n)^T \in E^n$ denote a vector of measurements (e.g. LANDSAT multispectral scanner data from either a single pass or several registered passes) taken from an arbitrary element of $\bigcup_{i=1}^L \Pi_i$. Suppose that the measurement vectors for class Π_i are distributed according to the n -dimensional multivariate normal density function

$$p_i(x) = (2\pi)^{-n/2} |\Sigma_i|^{-1/2} \exp \left[-\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right], \quad 1 \leq i \leq L.$$

We assume that the $n \times 1$ mean vector μ_i and the $n \times n$ covariance matrix Σ_i for each class Π_i are known with Σ_i positive definite, $1 \leq i \leq L$. The symbol $|A|$ is used to denote the determinant of the matrix A .

Given the partition*

$$\{1, 2, \dots, k_1, k_1 + 1, \dots, k_2, k_2 + 1, \dots, k_i, k_i + 1, \dots, k_m\}$$

of $\{1, 2, \dots, L\}$ into m subsets, let

$$\Gamma_i = \bigcup_{j=k_{i-1}+1}^{k_i} \Pi_j, \quad i = 1, 2, \dots, m,$$

where $k_0 = 0$, $k_m = L$, and let $m_i = k_i - k_{i-1}$.

Then Γ_i has a priori probability

*The program allows an arbitrary partition of the L classes into convex combinations. We assume the ordered partition defined herein only for purposes of simplifying notation in the discussion. See Example 2, Section 5.0.

$$\gamma_i = \sum_{j=k_{i-1}+1}^{k_i} \alpha_j, \quad i = 1, \dots, m.$$

Letting

$$h_i = \sum_{j=k_{i-1}+1}^{k_i} \frac{\alpha_j}{\gamma_i} p_j, \quad 1 \leq i \leq m$$

we see that each h_i is a convex combination of multivariate normals.

The mixture density for $\bigcup_{j=1}^L \Pi_j$ in terms of the combinations of classes

Γ_i , $i = 1, 2, \dots, m$ is easily seen to be

$$h = \sum_{i=1}^m \gamma_i h_i.$$

Since, in general, the α_j 's are unknown (and consequently so are the

γ_i 's), we allow for the specification of the a priori probabilities

β_1, \dots, β_m for $\Gamma_1, \dots, \Gamma_m$, respectively; that is, we assume the

expression for h given by

$$h = \sum_{i=1}^m \beta_i h_i.$$

The n -dimensional probability of misclassification, denoted by PMC,

of objects from $\bigcup_{i=1}^m \Gamma_i$ is given (see [1]) by

$$\begin{aligned} \text{PMC} &= 1 - \int_{E^n} \max_{1 \leq i \leq m} \beta_i h_i(x) dx \\ &= 1 - \sum_{i=1}^m \beta_i \int_{R_i} h_i(x) dx, \end{aligned}$$

where the sets R_i , $1 \leq i \leq m$, called the Bayes' decision regions, are defined by

$$R_i = \{x \in E^n \mid \beta_i h_i(x) = \max_{1 \leq j \leq m} \beta_j h_j(x)\}, \quad 1 \leq i \leq m.$$

The resulting classification procedure, called the Bayes' optimal classifier, is defined as follows:

Assign an element to Γ_i if its vector x of measurements belongs to R_i .

If $B = (b_1, \dots, b_n)$ is a nonzero $1 \times n$ vector and $x \in E^n$, then $y = Bx \in E^1$ and the transformed measurement vectors $y = Bx$ for class Π_i are distributed according to the univariate normal density function (see [1]) given by

$$p_i(y, B) = (2\pi)^{-1/2} (B\Sigma_i B^T)^{-1/2} \exp \left[-\frac{(y - B\mu_i)^2}{2B\Sigma_i B^T} \right], \quad 1 \leq i \leq L.$$

The transformed density for Γ_i is given by

$$h_i(y, B) = \sum_{j=k_{i-1}+1}^{k_i} \frac{\alpha_j}{\gamma_i} p_j(y, B), \quad i \leq i \leq m$$

The probability of misclassification g of an object from $\bigcup_{i=1}^m \Gamma_i$ in terms of the transformed measurements $y = Bx \in E^1$, as a function of nonzero B , is given by

$$\begin{aligned}
 g(B) &= 1 - \int_{E^1} \max_{1 \leq i \leq m} \beta_i h_i(y, B) dy \\
 &= 1 - \sum_{i=1}^m \beta_i \int_{R_i(B)} h_i(y, B) dy,
 \end{aligned}$$

where the transformed Bayes' decision regions are given by

$$R_i(B) = \left\{ y \in E^1 \mid \beta_i h_i(y, B) = \max_{1 \leq j \leq m} \beta_j h_j(y, B) \right\}, \quad 1 \leq i \leq m.$$

We use $G(B)$ to denote the probability of correct classification for B .

The computational procedure and associated computer program described in the sequel present a method for determining a nonzero $1 \times n$ vector B which minimizes g , or equivalently, which maximizes G . The method yields a linear feature selection procedure in that classification is ultimately performed in E^1 using only a single feature; namely, an optimal linear combination of the original measurements. The classification procedure in E^1 is described as follows:

If B is a nonzero $1 \times n$ vector which minimizes g , then assign an object to Γ_i if, for its measurement vector x , $Bx \in R_i(B)$.

Following arguments similar to those presented in [4], we obtain the expression for the Gateaux differential (see [7], [4]) of G (when it exists) given by

$$\delta G(B; C) = - \sum_{i=1}^m \beta_i \sum_{j=k_{i-1}+1}^{k_i} \frac{\alpha_j}{\gamma_i} p_j(y, B) \left[\frac{C \Sigma_i B^T}{B \Sigma_i B^T} (y - B \mu_i) + C \mu_i \right] \Bigg|_{R_i(B)}$$

where the notation $\left| \begin{array}{c} \\ R_i(B) \end{array} \right.$ denotes the sum of the values of the function at the right endpoints of the intervals comprising $R_i(B)$ minus the sum of its values at the left endpoint.

If B is a nonzero $1 \times n$ vector which minimizes $g(B) = 1 - G(B)$, then B must satisfy the vector equation

$$\frac{\partial g}{\partial B} = \begin{pmatrix} \delta g(B; C_1) \\ \vdots \\ \delta g(B; C_n) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix},$$

where C_j , $1 \leq j \leq n$, is a $1 \times n$ vector with a one in the j^{th} slot and zeros elsewhere. Using the formula for $\frac{\partial G}{\partial B}$ resulting from the above expression, and using the fact that $\frac{\partial g}{\partial B} = -\frac{\partial G}{\partial B}$, we obtain a numerically tractable expression for the variation in the probability of misclassification g with respect to B . The use of this expression in a computational procedure for obtaining a nonzero B which minimizes g is discussed in subsequent sections.

3.0 COMPUTATIONAL PROCEDURE

The computational procedure for determining the nonzero $1 \times n$ vector B which minimizes the probability of misclassification g with respect to the one-dimensional transformed density functions is embodied in the FORTRAN program LFSPMC (VERSION 2).

Apart from the various program parameters and command cards (discussed in the sequel), the basic input data to the program consists of the class names, mean vectors and covariance matrices which comprise the class statistics deck. All input data to the program is from unit reference 5 (usually punched cards). All output from the program is printed on unit reference 6. Several additional options are built into the program which provide the user with the capability of making successive runs using designated subsets of the original classes or features already provided by the class statistics deck.

The program is divided into the following four subsections which are discussed in turn in the sequel:

Parameter Initialization

Initial Vector Determination

Optimization Algorithm

Computation of $g(B)$ and $\frac{\partial g}{\partial B}$

3.1 Parameter Initialization

All input variables to the program are of a fixed format and must be entered as shown in Section 4.0 and as illustrated in the examples in Section 5.0. These variables are:

- MC : Number of convex combinations of classes, \leq MTOT.
- N : Dimension of feature space, \leq NFPC.
- CLS : Class names, 12 characters, double subscripted array.
- MTOT : Number of classes in the class statistics deck.
- NFPC : Number of features per class in the class statistics deck.
- KCLS : Numeric labels of the designated classes from the MTOT classes in the class statistics deck, single subscripted array.
- IFEA : Numeric labels of the N designated features from the NFPC features in the class statistics deck, single subscripted array.
- COVARB . . . : Input covariance matrices, triple subscripted array.
- XMEANB . . . : Input mean vectors, double subscripted array.
- APROB. . . . : A priori probabilities for the component classes, single subscripted array.
- BETA : A priori probabilities for the MC convex combinations, single subscripted array.
- ICMB : Numeric labels of component classes as defined in vector KCLS used in defining convex combinations (Need not be in ascending order, see Example 2, Section 5), double subscripted array.

Four command codes select program options as follows:

STAT - definition and entry of a given statistics deck,

FEAT - definition of a desired subset of features from the current statistics deck,

COMB - definition of convex combinations of a subset of class of the current statistics deck, along with associated a priori probabilities for the convex combinations and component classes.

FSEL - computation of the $1 \times n$ transformation vector which minimizes the PMC expression.

When the STAT command is used, values of MTOT and NFPC for the new statistics deck are entered. The names for the respective classes in the statistics deck are defined on succeeding cards. The class statistics deck, comprised of the MTOT mean vectors in the order of ascending class numbers followed by the MTOT covariance matrices in the order of ascending class numbers, is entered. The entries of each mean vector in the order of ascending feature number are entered according to the format (5X, 5D15.8). The $NFPC(NFPC + 1)/2$ elements on and above the diagonal of each covariance matrix are entered by column in the format (5X, 5D15.8). It is assumed that the diagonal elements of each covariance matrix are in order of ascending feature number. The first entry of each new mean vector or covariance matrix starts on a new card. The entire statistics deck with appropriate class names is printed.

If the FEAT command is selected, a new value for N and the numeric labels of the desired features (IFEA) are entered and printed.

The COMB command requires entry of the numeric labels and a priori probabilities of the desired component classes defining each convex combination. After each combination is defined, the a priori probability for the combination is entered. The names of the classes defining the combinations and the a priori probabilities are output. Parameters initialized using the STAT, FEAT, and COMB commands remain in effect until the respective command is again used.

The FSEL command requires the input of

IZ : initial B-vector flag
 = 0 compute the initial vector B_0 for the
 optimization algorithm.
 = 1 input the initial vector B_0

The FSEL command must be preceded by the STAT and COMB commands.

3.2 Initial Vector Determination

A nonzero $1 \times n$ vector B_0 which minimizes g cannot, in general, be obtained in closed form, and a numerical optimization (minimization) procedure is required. Any such optimization algorithm must be given an initial vector B_0 .

When the initial vector B_0 is to be computed, SUBROUTINE BCOMP is called. For the special case of two multivariate normal classes with equal a priori probabilities, B_0 is computed in SUBROUTINE BC2CP using the formula (see [3], [4])

$$B_0 = (\mu_1 - \mu_2)^T (\Sigma_1 + \Sigma_2)^{-1}$$

In all other cases the initial vector is computed using the procedure described below (see [5], [9]).

Given α_i , μ_i , and Σ_i , $1 \leq i \leq L$, let

$$\Sigma = \frac{1}{L} \sum_{i=1}^L \alpha_i \Sigma_i$$

and determine (using SUBROUTINE EIGEN see [10]) an $n \times n$ matrix A such that $A \Sigma A^T = I$. Letting $\eta_i = A \mu_i$, $1 \leq i \leq L$, the problem can then be reduced to finding a fixed point of the function H defined as follows:

For a given $1 \times n$ vector C , choose indices i_j , $1 \leq j \leq L$, for the η_i 's and α_i 's such that ($<$ indicates ordering of intervals)

$$R_{i_1}(C) < R_{i_2}(C) < \dots < R_{i_L}(C),$$

where

$$R_{i_j}(C) = \left\{ y \mid \alpha_{i_j} p_{i_j}(y, C) = \max_{\substack{1 \leq k \leq L \\ k \neq j}} \alpha_{i_k} p_{i_k}(y, C) \right\}$$

To determine the regions $R_{i_j}(C)$, the roots between transformed

densities considered pairwise are computed as

$$a_{i_j}(C) = \frac{\ln(\alpha_i/\alpha_j)}{C(\eta_j - \eta_i)} + \frac{C(\eta_j + \eta_i)}{2}, \quad i \neq j, \quad i, j = 1, \dots, L.$$

Once all the roots $a_{i_j}(C)$ have been found, the regions $R_{i_j}(C)$,

$j = 1, 2, \dots, L$ are defined by the following ordering:

a) Choose i_1 such that

$$C_{n_{i_1}} = \min_{1 \leq j \leq L} C_{n_j}$$

b) Choose i_2 such that

$$a_{i_2 i_1} = \min_{1 \leq j \leq L} a_{j i_1}$$

c) Given i_1 and i_2 , choose i_3 such that

$$a_{i_3 i_2} = \min_{1 \leq j \leq L} \left\{ a_{j i_2} \mid a_{j i_2} > a_{i_2 i_1} \right\}$$

d) In general choose i_{k+1} such that

$$a_{i_{k+1} i_k} = \min_{1 \leq j \leq L} \left\{ a_{j i_k} \mid a_{j i_k} > a_{i_k i_{k-1}} \right\}$$

provided

$$\left\{ a_{j i_k} \mid a_{j i_k} > a_{i_k i_{k-1}} \right\} \neq \emptyset$$

If the above set is empty, the procedure is terminated.

For

$$a_j = \frac{\ln(\alpha_{i_j} / \alpha_{i_{j+1}})}{C \left(\mu_{i_{j+1}} - \mu_{i_j} \right)} + \frac{C \left(n_{i_{j+1}} + n_{i_j} \right)}{2}, \quad 1 \leq j \leq L-1$$

we let

$$F(C) = \sum_{j=1}^{L-1} \alpha_{i_j} p_{i_j}(a_j, C) (n_{i_{j+1}} - n_{i_j}).$$

Then

$$H(C) = \frac{F(C)^T}{\|F(C)^T\|_2} .$$

To find a fixed point of H ($C = H(C)$), we let $C_0 = \eta_i - \eta_j$,

where

$$\|\eta_i - \eta_j\|_2 = \max_{r \neq s} \|\eta_r - \eta_s\|_2,$$

and compute successive vectors C_k using the mean iteration formula (see [8])

$$C_{k+1} = \frac{k}{k+1} C_k + \frac{1}{k+1} H(C_k), \quad k = 0, 1, 2, \dots .$$

The number of iterations is specified by the internal parameter ITER (25 is a reasonable value). Upon completion of the iterations, the final C_k , say C , is used to compute an initial vector B_0 from the formula

$$B_0 = CA .$$

3.3 Optimization Algorithm

The numerical minimization algorithm used to find a local minimum of g is SUBROUTINE DFMFP from the IBM Scientific Subroutine Package [10]. The procedure is based on the method of Fletcher and Powell [2]. Computation of the minimizing B is controlled by SUBROUTINE BVECT which initializes the following parameters used by DFMFP:

- EST : An estimate of the minimum value of $g(B)$.
- EPS : Tolerance for the expected absolute error of the optimization algorithm. Experience has shown 10^{-4} to be a reasonable value.
- LIMIT . . . : Maximum number of iterations for the optimization algorithm.

Values for the parameters are computed using empirically determined formulae. On return from SUBROUTINE DFMFP, the error parameter IER is checked and a message is printed if convergence is not achieved in LIMIT iteration, if the gradient calculations are in error, or if no minimum is found. Calculation of $g(B)$ and $\frac{\partial g}{\partial B}$ is discussed below.

3.4 Computation of $g(B)$ and $\frac{\partial g}{\partial B}$

The computation of $g(B)$ and $\frac{\partial g}{\partial B}$ using the expressions given in Section 2.0 is performed in SUBROUTINE FUNCT. The function subprogram DPHIX computes

$$\phi(a, \mu, \sigma^2) = \frac{1}{2} + \frac{1}{2} \operatorname{ERF}\left(\frac{a-\mu}{\sqrt{2}\sigma}\right)$$

used in the computation of $g(B)$, where ERF is a library function subprogram given by

$$\operatorname{ERF}(a) = 2(2\pi)^{-\frac{1}{2}} \int_0^a \exp\left[-\frac{1}{2} t^2\right] dt .$$

The transformed density functions $p_i(y, B)$, $1 \leq i \leq L$ are evaluated in the function subprogram XNDF. The function subprogram FUNVAL

computes

$$h_i = \sum_{j=k_{i-1}+1}^{k_i} \frac{\alpha}{\gamma_i} p_j(y, B), \quad 1 \leq i \leq m.$$

In order to evaluate $g(B)$ and $\frac{\partial g}{\partial B}$, it is necessary to determine the regions $R_i(B)$, $1 \leq i \leq m$, defined in Section 2.0.

To determine the regions $R_i(B)$, $1 \leq i \leq m$, the roots, $y_{ij}(B)$, of the equations

$$F_{ij}(B) = \beta_i h_i - \beta_j h_j = 0, \quad i \neq j, \quad i, j = 1, 2, \dots, m$$

are determined. If $m_i = m_j = 1$ (i.e. h_i and h_j each consist of a single normal class), then the roots of $F_{ij}(B)$ are determined in SUBROUTINE ROOTSS by evaluating the quadratic equation

$$F_{ij}(B) = n_{ij}(B)y^2 + 2\rho_{ij}(B)y + v_{ij}(B),$$

where

$$n_{ij}(B) = B \Sigma_{k_i} B^T - B \Sigma_{k_j} B^T$$

$$\rho_{ij}(B) = (B \Sigma_{k_j} B^T) B \mu_{k_i} - (B \Sigma_{k_i} B^T) B \mu_{k_j}$$

and

$$v_{ij}(B) = (B \Sigma_{k_i} B^T) (B \mu_{k_j})^2 - (B \Sigma_{k_j} B^T) (B \mu_{k_i})^2 \\ + (B \Sigma_{k_i} B^T) (B \Sigma_{k_j} B^T) \ln \left(\frac{\beta_i^2 B \Sigma_{k_j} B^T}{\beta_j^2 B \Sigma_{k_i} B^T} \right).$$

For the case where $\eta_{ij}(B) = 0$, a single root

$$y_{ij}(B) = \frac{(B\mu_{k_j})^2 - (B\mu_{k_i})^2 + 2(\ln \frac{\beta_i}{\beta_j})(B\Sigma_{k_i} B^T)}{2(B\mu_{k_j} - B\mu_{k_i})}$$

is obtained.

When either $m_i \neq 1$ or $m_j \neq 1$, the roots of the equation

$F_{ij}(B) = 0$ are determined using Newton's method in SUBROUTINE ROOTSM.

A maximum of $2(m_i + m_j)$ roots are searched for in the interval

(x_1, x_2) , where

$$x_1 = \min_{k \in K} \left\{ B\mu_k - 3\alpha_k (B\Sigma_k B^T)^{1/2} \right\}$$

$$x_2 = \max_{k \in K} \left\{ B\mu_k + 3\alpha_k (B\Sigma_k B^T)^{1/2} \right\}$$

with

$$K = \left\{ k_{i-1}+1, \dots, k_i \right\} \cup \left\{ k_{j-1}+1, \dots, k_j \right\} .$$

The starting values

$$x_0 = B\mu_k + \left(\frac{1}{2} B\Sigma_k B^T \right)^{1/2}$$

and

$$x_0 = B\mu_k - \left(\frac{1}{2} B\Sigma_k B^T \right)^{1/2}$$

are needed for each component class, k , in each of the two convex combinations considered. Updating of the argument, x , the function value, y , and the derivative, d , are continued until one of the

following conditions is met:

$$(a) \quad |y| < 10^{-4} .$$

$$(b) \quad |d| < \frac{|y|}{x_2 - x_1} .$$

(c) The number of iterations of Newton's method exceeds 10 .

$$(d) \quad x \notin (x_1, x_2) .$$

Only when condition (a) is met is a root, $y_{ij}(B)$, of $F_{ij}(B) = 0$ defined.

After $2(m_i + m_j)$ starting values are used, the roots thus determined are arranged in ascending order. Roots having approximately equal values are combined into a single root.

Once the roots, $y_{ij}(B)$, of all the equations $F_{ij}(B) = 0$ have been found, the regions $R_i(B)$, $1 \leq i \leq m$, are determined by the following ordering defined on a (possibly proper) subset of the roots:

(a) Choose i_1 , such that

$$\beta_{i_1} \sum_{k=k_{i_1-1}+1}^{k_{i_1}} \frac{\alpha_k}{\gamma_{i_1}} p_k(y_{i_2 i_1}, B) = \max_{1 \leq j \leq m} \beta_j \sum_{k=k_{j-1}+1}^{k_j} \frac{\alpha_k}{\gamma_j} p_k(y_{i_2 i_1}, B) ,$$

where

$$y_{i_2 i_1} = \min_{\substack{1 \leq j \leq m \\ 1 \leq r \leq m}} [y_{jr}(B) - B \Sigma_1 B^T]$$

considering only the subset of roots at which at least one of the convex combinations of densities has a value other than machine zero.

(b) Choose i_2 such that

$$\beta_{i_2} \sum_{k=k_{i_2-1}+1}^{k_{i_2}} \frac{\alpha_k}{\gamma_{i_2}} p_k(y_0, B) = \max_{1 \leq j \leq m} \beta_j \sum_{k=k_{j-1}+1}^{k_j} \frac{\alpha_k}{\gamma_j} p_k(y_0, B),$$

where

$$y_0 = \min_{\substack{1 \leq j \leq m \\ 1 \leq r \leq m}} \{y_{jr} \mid y_{jr} > y_{i_2 i_1}\}$$

(c) Let

$$y_{i_3 i_2} = \min_{1 \leq j \leq m} \{y_{ji_2} \mid y_{ji_2} > y_{i_2 i_1}\}.$$

(d) In general, choose i_q such that

$$\beta_{i_q} \sum_{k=k_{i_q-1}+1}^{k_{i_q}} \frac{\alpha_k}{\gamma_{i_q}} p_k(y_0, B) = \max_{1 \leq j \leq m} \beta_j \sum_{k=k_{j-1}+1}^{k_j} \frac{\alpha_k}{\gamma_j} p_k(y_0, B)$$

where

$$y_0 = \min_{\substack{1 \leq j \leq m \\ 1 \leq r \leq m}} \{y_{jr} \mid y_{jr} > y_{i_q i_{q-1}}\},$$

If $\{y_{jr} \mid y_{jr} > y_{i_q i_{q-1}}\} = \emptyset$, let

$$y_0 = y_{i_q i_{q-1}} + B \Sigma_1 B^T .$$

(e) Choose

$$y_{i_{q+1} i_q} = \min_{1 \leq j \leq m} \{ y_{j i_q} \mid y_{j i_q} > y_{i_q i_{q-1}} \}$$

if

$$\{ y_{j i_q} \mid y_{j i_q} > y_{i_q i_{q-k}} \} \neq \emptyset .$$

When the above set is empty the procedure is terminated. The regions $R_j(B)$ are given for $1 \leq j \leq m$, by

$$R_0(B) = \bigcup_{I_j} \{ y \mid y_{i_q i_{q-k}} < y \leq y_{i_{q+1} i_q} \} ,$$

where

$$I_j = \{ i_q : i_q = j \} .$$

The above procedure for determining the regions may detect unnecessary roots and assign the intervals to the left and right of such a root to the same region $R_i(B)$. The lack of machine precision may prohibit the evaluation of densities sufficiently accurate so that the same combination is defined to dominate at $+\infty$ and $-\infty$.

For both the initial B vector and the final normalized B vector, SUBROUTINE FUNCT outputs $R_i(B)$, $B \Sigma_i B^T$, $B \mu_i$, $g(B)$, $\frac{\partial g}{\partial B}$, and B . In outputting the transformed means and covariances, the classes are numbered in the order entered under the COMB command. For

the final normalized B , the entries q_{ij} of the confusion matrix are computed using the formulas

$$q_{ij} = \int_{R_i(B)} h_j(y, B) dy$$

and output.

The parameter IOUT is an internal output control flag provided to SUBROUTINE FUNCT.

IOUT : Control flag
= -1 first pass (B_0) printed.
= 0 intermediate iterations of DFMP printed.
= 1 last pass (final B) printed.

4.0 OPERATING PROCEDURE

In order to simulate object time dimensioning, the user must provide a calling routine of the following form:

```

DIMENSION ALRGE(IDIM)
DOUBLE PRECISION BLRGE(IDIM2)
COMMON MX,NX
MX =
NX =
CALL PRDIM(ALRGE,BLRGE)
STOP
END

```

The values of MX, NX, IDIM and IDIM2 are determined as follows:

MX = maximum value of MTOT for the program run.

NX = maximum value of NFPC for the program run.

$$IDIM = MX(23+4MX+NX(\frac{NX+1}{2}))+NX(\frac{5NX+5}{2})+12$$

$$IDIM2 = MX(7+3MX+\frac{MX^2}{3}+3NX+2NX^2)+NX(\frac{NX}{2} + \frac{11}{2})+7$$

If available storage is not a problem, the user can incorporate maximum fixed dimensions into the program.

The program is suitable for interactive operation with the inclusion of parameter request messages. The program was written in IBM Fortran G with development on the Texas A&M University IBM 360-65.

Input parameters are of a fixed format and must be in a specified order. Shown below are the variable names as described in Sections 3.1 - 3.4 and the card formats for the command code sequences.

Statistics Definition:

```
"STAT",IO,NFPC
class names (one per card) [FORMAT(3A4)]
mean vectors                [FORMAT(5X,5D15.8)]
covariance matrices        [FORMAT(5X,5D15.8)]
```

Feature Definition:

```
"FEAT",N                    [FORMAT(A4,I2)]
IFEA                        [FORMAT(24(I2,1X))]
```

Several cards may be used to define IFEA if $N > 24$.

Combination Definition:

```
"COMB",MC                    [FORMAT(A4,I2)]
KCLS(1)-APROB(1), . . . . . , KCLS(M1)-APROB(M1),99-BETA(1)
[FORMAT(7(I2,1X,F7.6,1X),2X,A1)]
```

Each card defines the component class numbers (from the current statistics deck) along with associated a priori probabilities for each convex combination.

After all classes of a combination are defined, 99 is entered as a class number followed by the a priori probability for the combination. If the a priori probability for the combination is to be the sum of the a prioris of its component classes, the class number 99 and associated

a priori value can be omitted. If more than seven classes compose a combination, a slash (/) in column 80 of the current card indicates continued definition of the combination on succeeding cards.

Feature Selection:

"FSEL",IZ [FORMAT(A4,IZ)]

BVECT* [FORMAT(5X,5D15.8)]

*BVECT is entered only if IZ = 1.

5.0 EXAMPLE INPUT AND OUTPUT

Example 1.

A 5 class, 12 dimensional statistics deck from MSS measurements of Hill County, Montana, is entered. A subset of 8 features (5-12) is considered. Combination 1 is defined as class 1 and combination 2 is the combination of classes 2-5. Feature selection is performed.

Col. 1

STAT0512

WHEAT

FALLOW

BARLEY

GRASS

STUBBLE

statistics deck

FEAT08

05,06,07,08,09,10,11,12

COMB02

01-.347000,99-.5

02-.243000,03-.121000,04-.056000,05-.233000,99-.5

FSEL

Example 2.

A 6 class, 12 dimensional statistics deck from MSS measurements of Hill County, Montana, is entered. Each class is defined as a separate combination. Feature selection is performed. Next, combinations of classes 4 and 5, 1, and 2, 3, and 6 are defined and feature selection

performed.

Col. 1

STAT0612

BARLEY

STUBBLE

GRASS

WINTER WHEAT

SPRING WHEAT

FALLOW

statistics deck

COMB06

01-.166666

02-.166666

03-.166666

04-.166666

05-.166666

06-.166666

FSEL

COMB03

04-.166666,05-.166666,99-.333333

01-.166666,99-.333333

02-.166666,03-.166666,06-.166666,99-.333333

FSEL

COVARIANCE MATRIX FOR WHEAT

6.144	7.186	6.333	3.116	1.482	2.100	0.937	-0.186
3.123	5.532	1.062	-0.632	2.053	1.988	1.635	0.348
7.186	14.840	12.071	5.588	2.060	1.763	1.670	-0.047
5.702	9.277	2.307	-0.422	1.299	1.585	1.224	0.163
6.333	12.071	13.436	5.866	7.853	10.557	2.522	-1.697
5.280	8.483	1.975	-0.325	10.557	17.949	3.370	-3.018
3.116	5.588	5.869	3.502	2.522	3.370	5.448	2.084
2.856	4.488	1.366	0.055	10.557	17.949	3.370	-3.018
1.482	2.053	2.060	1.299	7.853	10.557	2.522	-1.697
4.170	7.479	2.684	0.314	10.557	17.949	3.370	-3.018
2.100	1.988	1.763	1.585	2.060	1.763	1.670	-0.047
5.542	10.201	3.409	0.419	1.299	1.585	1.224	0.163
0.937	1.635	1.670	1.224	7.853	10.557	2.522	-1.697
1.262	2.002	2.276	1.475	10.557	17.949	3.370	-3.018
-0.186	0.348	-0.047	0.163	-1.697	-3.018	2.084	3.399
-0.703	-1.754	1.175	1.227	4.170	5.542	1.262	-0.703
3.323	5.702	5.280	2.856	7.479	10.301	2.002	-1.754
7.195	11.036	2.837	-0.875	4.488	3.409	2.276	1.175
5.532	9.277	9.483	4.488	2.684	2.684	2.276	1.175
11.036	21.787	4.008	-2.529	0.314	0.439	1.475	1.227
1.062	2.307	1.975	1.366	0.055	0.439	1.475	1.227
2.837	4.608	5.635	1.996	0.055	0.439	1.475	1.227
-0.632	-0.422	-0.325	0.055	0.314	0.439	1.475	1.227
-0.875	-2.529	1.996	2.566				

MEAN VECTOR FOR WHEAT

48.115	58.212	56.478	25.965	33.611	30.274	52.106	30.611
37.876	36.177	52.389	27.221				

COVARIANCE MATRIX FOR FALLOW

4.789	5.355	6.533	3.590	1.560	2.767	2.694	1.536
0.927	1.532	2.730	1.151				
5.355	10.520	11.781	6.431	2.631	4.321	4.613	2.251
1.254	1.518	2.545	0.969				
6.533	11.781	16.566	8.509	3.002	5.119	5.432	2.649
1.247	3.269	2.632	1.005				
3.590	6.431	8.509	5.328	1.642	3.006	3.065	1.520
1.386	2.289	2.175	1.009				
1.560	2.631	3.002	1.642	4.158	4.266	3.434	1.472
0.963	1.665	1.169	0.318				
2.767	4.321	5.119	3.006	4.266	6.346	4.958	1.991
0.902	1.873	2.325	0.715				
2.694	4.613	5.432	3.065	3.434	4.958	6.247	2.686
1.524	2.247	2.125	0.904				
1.936	2.251	2.649	1.520	1.472	1.991	2.686	1.802
1.109	1.371	1.715	0.823				
0.927	1.254	1.247	1.386	0.963	0.902	1.524	1.109
5.358	5.821	3.676	1.370				
1.632	1.918	3.269	2.289	1.665	1.873	2.247	1.371
5.831	11.992	2.817	0.162				
2.730	2.545	2.632	2.175	1.189	2.325	2.125	1.715
3.676	2.817	10.166	5.132				
1.151	0.969	1.005	1.009	0.318	0.715	0.904	0.823
1.370	0.162	5.132	3.755				

MEAN VECTOR FOR FALLOW

43.429	47.839	44.625	19.670	39.938	43.536	43.759	20.804
40.634	44.446	43.920	20.116				

COVARIANCE MATRIX FOR EARLEY

5.367	9.473	8.169	4.204	5.227	2.862	0.366
2.636	3.292	1.555	-0.008			
9.473	29.456	25.608	13.049	16.466	7.470	-0.739
10.787	14.043	7.283	1.768			
8.169	25.608	29.702	13.845	19.003	6.839	-2.537
10.941	14.732	6.315	1.991			
4.204	13.049	13.845	8.315	10.554	4.503	-1.265
6.132	8.414	3.684	0.956			
4.068	16.466	19.003	10.554	25.172	6.211	-2.837
12.739	16.215	8.625	3.057			
5.227	24.218	29.184	16.858	39.515	12.727	-6.227
19.866	26.307	12.936	3.914			
2.862	7.470	6.839	4.503	8.211	12.727	7.591
4.428	5.391	3.280	0.467			0.819
0.366	-0.739	-2.537	-1.265	-2.837	-6.227	0.819
-0.971	-1.579	0.288	0.016			2.743
2.636	10.787	10.941	6.132	12.739	15.866	4.428
11.764	12.602	8.933	3.787			-0.571
3.292	14.043	14.732	8.414	16.215	26.307	5.391
12.602	18.565	9.657	3.566			-1.579
1.555	7.283	6.315	3.684	8.625	12.936	3.280
8.408	9.657	10.573	4.408			0.288
-0.008	1.768	1.991	0.956	3.057	3.934	0.467
3.787	3.566	4.408	2.762			0.016

MEAN VECTOR FOR EARLEY

48.600	60.240	60.380	28.520	40.400	57.520	31.060
43.540	48.380	49.500	23.080			

EXAMPLE 1: SAMPLE OUTPUT

COVARIANCE MATRIX FOR GRASS

3.196	3.800	2.463	1.393	1.906	3.194	1.713	0.861
1.204	1.718	2.331	0.982	3.386	5.933	3.181	1.621
3.900	6.831	4.007	2.130	2.023	4.067	3.083	1.756
1.957	3.236	3.963	1.613	1.429	2.666	1.376	1.369
2.463	4.007	5.600	2.089	4.159	5.406	2.665	1.204
2.001	3.233	3.890	1.715	5.406	11.951	5.456	2.549
1.393	2.130	2.089	1.734	2.666	2.666	2.666	2.592
1.223	1.757	2.296	1.158	2.665	6.052	6.052	2.592
1.906	3.386	2.023	1.429	2.665	5.496	6.052	2.592
1.360	2.430	2.407	1.360	1.204	2.549	2.592	1.888
3.194	5.933	4.067	2.666	2.665	1.579	1.579	1.888
2.731	4.240	5.044	2.597	2.665	5.496	6.052	2.592
1.713	3.181	3.083	1.579	2.665	5.496	6.052	2.592
1.649	2.462	3.338	1.591	1.204	2.549	2.592	1.888
0.861	1.621	1.796	1.169	1.360	2.731	1.649	1.062
1.062	1.588	1.918	1.014	1.360	2.731	1.649	1.062
1.204	1.557	2.001	1.223	2.430	4.240	2.462	1.588
3.417	3.799	3.288	1.364	2.430	4.240	2.462	1.588
1.718	3.236	3.233	1.757	2.407	5.044	3.338	1.918
3.799	7.609	4.756	2.153	2.407	5.044	3.338	1.918
2.331	3.920	3.920	2.296	1.300	2.597	1.591	1.014
3.288	4.756	7.636	3.288	1.300	2.597	1.591	1.014
0.982	1.613	1.715	1.158	38.881	37.505	46.614	24.445
1.364	2.153	3.256	1.595	38.881	37.505	46.614	24.445

MEAN VECTOR FOR GRASS

42.723	44.792	43.574	19.911	38.881	37.505	46.614	24.445
39.693	38.485	45.228	22.218	38.881	37.505	46.614	24.445

COVARIANCE MATRIX FOR STUBBLE

1.791	1.668	1.416	0.892	0.437	0.621	0.859	0.424
-0.171	-0.005	-0.287	-0.291				
1.608	4.777	2.579	1.667	1.236	1.660	2.474	1.510
0.296	1.267	0.439	-0.099				
1.416	2.579	3.527	1.844	1.315	1.525	2.688	1.610
0.469	1.662	1.161	0.244				
0.892	1.667	1.844	1.844	1.263	1.470	1.822	1.151
0.363	1.064	0.638	0.113				
0.437	1.236	1.315	1.263	2.727	2.344	2.237	1.556
0.648	1.468	1.009	0.314				
0.621	1.660	1.525	1.470	2.344	4.757	2.675	1.754
0.599	1.832	1.379	0.664				
0.859	2.474	2.688	1.822	2.237	2.675	7.398	3.625
2.090	3.504	2.895	1.172				
0.424	1.610	1.610	1.151	1.596	1.794	3.625	2.562
1.537	2.229	2.130	0.837				
-0.171	0.256	0.459	0.363	0.648	0.599	2.090	1.537
3.778	3.316	3.855	1.622				
-0.005	1.267	1.662	1.064	1.468	1.932	3.504	2.229
3.316	5.016	4.288	1.856				
-0.287	0.439	1.161	0.638	1.009	1.379	2.895	2.130
3.855	4.288	6.578	3.054				
-0.291	-0.699	0.244	0.118	0.314	0.664	1.172	0.837
1.622	1.896	3.054	2.355				

MEAN VECTOR FOR STUBBLE

43.688	47.661	44.459	19.743	39.917	44.184	44.055	21.468
42.449	46.220	47.716	22.615				

EXAMPLE 1: SAMPLE OUTPUT

FEATURES 8, 15, 18, 9, 10, 11, 12.

CLASS 1 (BETA=.50000) IS THE CONVEX COMBINATION OF:
WHEAT (ALPHA=.34700)

CLASS 2 (BETA=.50000) IS THE CONVEX COMBINATION OF:
FALLOW (ALPHA=.24300)
BARLEY (ALPHA=.12100)
STUBBLE (ALPHA=.05600)

ITERATION ROUTINE PARAMETERS

ESTIMATED MINIMUM=0.10
ABSOLUTE ERROR=1.0E-04
MAXIMUM NUMBER OF ITERATIONS= 60

EXAMPLE 1: SAMPLE OUTPUT

RESULTS FOR INITIAL B VECTOR

REGIONS
 -0.10000000 31 <Y< 0.35242436D 02 R 2(B)
 0.35242436D C2 <Y< 0.40890274D C2 R 2(B)
 0.40890274D 02 <Y< 0.40895037D 02 R 1(B)
 0.40895037D 02 <Y< 0.10000000D 31 R 1(B)

CLASS	TRANSFORMED COVARIANCES	TRANSFORMED MEANS
1	0.56478381D 01	0.453357C9D 02
2	0.52313452D 01	0.30179453D 02
3	0.58676217D 01	0.40087329D 02
4	0.27121558D 01	0.35951255D 02
5	0.33934265D 01	0.32896736D 02

PROB. OF MISCLASS. FOR CURRENT B= 0.49707552D-01

VECTOR OF GRADIENTS

FNTX(1) = 0.66544317D-01 FNTX(2) = 0.86961566D-01 FNTX(3) = 0.10265477D 00
 FNTX(4) = 0.51586735D-01 FNTX(5) = 0.66305966D-01 FNTX(6) = 0.12137666D 00
 FNTX(7) = -0.88839876D-02 FNTX(8) = -0.21225043D-01

B VECTOR

B(1) = 0.88118732D-01 B(2) = -0.25328571D 00 B(3) = 0.103444052D 00
 B(4) = 0.45757276D 00 B(5) = 0.21713215D 00 B(6) = -0.12775761D 00
 B(7) = 0.98430657D-01 B(8) = 0.80357569D 00

EXAMPLE 1: SAMPLE OUTPUT

NO CONVERGENCE IN LIMIT ITERATIONS

RESULTS FOR FINAL B VECTOR

REGIONS					
-0.10000000	31	<YK	0.57692522D 01	R 2 (B)	
0.57692522D	01	<YK	0.13492958D 02	R 2 (B)	
0.13492958D	02	<YK	0.13491728D 02	R 2 (B)	
0.13491728D	02	<YK	0.13496510D 02	R 1 (B)	
0.13496610D	02	<YK	0.10000000D 31	R 1 (B)	

CONFUSION MATRIX

0.98900 0.01107
0.01100 0.58853

CLASS TRANSFORMED COVARIANCES TRANSFORMED MEANS

1	C.77137058D 01	0.19852816D 02
2	0.83362239D 01	0.52910362D 01
3	C.75604461D 01	C.87654870C 01
4	0.30757874D 01	0.58329101D 01
5	0.32091904D 01	0.82313570D 01

PROB. OF MISCLASS. FOR CURRENT B = 0.11035557D-01

VECTOR OF GRADIENTS

FNTX(1) = 0.14847173D-03 FNTX(2) = -0.20636982D-04 FNTX(3) = -0.37869692D-04
FNTX(4) = 0.17573345D-03 FNTX(5) = -0.19357910D-04 FNTX(6) = 0.37368907D-04
FNTX(7) = 0.10476962D-03 FNTX(8) = 0.36413124D-04

B VECTOR

B(1) = -0.11152256D 00 B(2) = -0.16769531D 00 B(3) = 0.46053265D-01
B(4) = -0.13729909D 00 B(5) = -0.32156049D 00 B(6) = -0.18688114D 00
B(7) = 0.62641947D 00 B(8) = 0.63275332D 00

EXAMPLE 1: SAMPLE OUTPUT

COVARIANCE MATRIX FOR EARLEY

5.367	5.473	8.159	4.204	4.068	5.227	2.862	0.366
2.636	3.202	1.555	-0.008				
9.473	25.456	25.608	13.049	16.466	24.218	7.470	-0.739
10.787	14.043	7.283	1.763				
8.159	25.608	28.702	13.845	19.003	29.184	6.839	-2.537
10.941	14.732	6.315	1.991				
4.204	13.049	13.245	8.315	10.554	16.858	4.503	-1.269
6.132	8.414	3.694	0.956				
4.068	16.466	19.003	10.554	25.172	39.515	8.211	-2.837
12.739	16.215	8.625	3.057				
5.227	24.218	29.184	16.858	39.515	67.365	12.727	-6.227
19.866	26.307	12.936	3.934				
2.862	7.470	6.839	4.503	8.211	12.727	7.591	0.819
4.428	5.391	3.280	0.467				
0.366	-0.739	-2.537	-1.269	-2.837	-6.227	0.819	2.743
-0.971	-1.579	0.288	0.016				
2.636	10.787	10.941	6.132	12.739	19.866	4.428	-0.971
11.764	12.602	8.933	3.787				
3.292	14.043	14.732	8.414	16.215	26.307	5.391	-1.579
12.602	18.565	9.657	3.566				
1.555	7.283	6.315	3.684	8.625	12.936	3.280	0.288
8.933	5.657	10.573	4.408				
-0.008	1.768	1.991	0.956	3.057	3.934	0.467	0.016
3.787	3.506	4.408	2.762				

MEAN VECTOR FOR EARLEY

48.600	60.240	60.380	28.520	40.400	40.820	57.520	31.060
43.540	48.330	49.500	23.080				

COVARIANCE MATRIX FOR STUBBLE

1.791	1.608	1.416	0.892	0.437	0.621	0.859	0.424
-0.171	-0.005	-0.287	-0.291				
1.608	4.777	2.579	1.667	1.236	1.660	2.474	1.510
0.296	1.267	0.439	-0.059				
1.416	2.579	3.527	1.844	1.315	1.525	2.688	1.610
0.469	1.662	1.161	0.244				
0.892	1.667	1.844	1.844	1.263	1.470	1.822	1.151
0.363	1.064	0.538	0.118				
0.437	1.236	1.315	1.267	2.727	2.344	2.237	1.596
0.648	1.468	1.009	0.314				
0.621	1.660	1.525	1.470	2.344	4.757	2.675	1.794
0.599	1.832	1.370	0.664				
0.859	2.474	2.688	1.822	2.237	2.675	7.398	3.625
2.090	3.504	2.895	1.172				
0.424	1.510	1.610	1.151	1.596	1.794	3.625	2.562
1.537	2.229	2.130	0.837				
-0.171	0.296	0.469	0.263	0.648	0.599	2.090	1.537
3.778	3.316	3.955	1.622				
-0.005	1.267	1.662	1.064	1.468	1.832	3.504	2.229
3.316	6.015	4.288	1.895				
-0.287	0.439	1.161	0.638	1.009	1.379	2.895	2.130
3.855	4.288	6.978	3.054				
-0.291	-0.099	0.244	0.118	0.314	0.564	1.172	0.837
1.622	1.896	3.054	2.085				

MEAN VECTOR FOR STUBBLE

43.688	47.661	44.459	19.743	39.917	44.184	44.055	21.468
42.449	46.220	47.716	22.615				

EXAMPLE 2: SAMPLE OUTPUT

COVARIANCE MATRIX FOR GRASS

3.196	3.800	2.463	1.393	1.906	3.194	1.713	0.861
1.204	1.718	2.331	0.982				
3.800	6.831	4.007	2.430	3.386	5.933	3.181	1.621
1.957	3.236	3.962	1.611				
2.463	4.007	5.600	2.089	2.023	4.067	3.083	1.756
2.001	3.233	3.890	1.715				
1.393	2.430	2.089	1.714	1.429	2.666	1.978	1.189
1.223	1.757	2.226	1.158				
1.906	3.386	2.023	1.429	4.159	5.406	2.665	1.204
1.360	2.430	2.407	1.300				
3.194	5.933	4.067	2.666	5.406	11.951	5.456	2.549
2.731	4.240	5.044	2.597				
1.713	3.191	3.083	1.978	2.665	5.496	6.052	2.592
1.649	2.462	3.338	1.591				
0.861	1.621	1.796	1.189	1.204	2.549	2.592	1.888
1.062	1.588	1.918	1.014				
1.204	1.957	2.001	1.223	1.360	2.731	1.649	1.062
3.417	3.799	3.288	1.364				
1.718	3.236	3.233	1.757	2.430	4.240	2.462	1.528
3.799	7.609	4.756	2.153				
2.331	3.563	3.890	2.296	2.407	5.044	3.338	1.918
3.288	4.756	7.636	3.256				
0.861	1.618	1.715	1.158	1.360	2.597	1.591	1.014
1.364	2.153	3.256	1.995				

MEAN VECTOR FOR GRASS

42.723	44.792	43.574	19.911	38.881	37.505	46.614	24.445
39.693	38.485	45.228	22.218				

COVARIANCE MATRIX FOR WINTER WHEAT

6.144	7.186	6.333	3.116	1.482	2.100	0.937	-0.186
5.323	5.552	1.062	-0.632	2.053	1.988	1.635	0.348
7.186	14.846	12.071	5.588	2.060	1.763	1.670	-0.047
5.702	5.277	2.307	-0.422	1.299	1.585	1.224	0.163
6.333	12.071	13.436	5.569	7.853	10.557	2.522	-1.657
5.280	8.483	1.975	-0.325	10.557	17.949	3.370	-3.018
3.116	5.588	5.869	3.502	2.522	3.370	5.448	2.084
2.856	4.488	1.366	0.055	2.522	3.370	5.448	2.084
1.482	2.053	2.060	1.299	7.853	10.557	2.522	-1.657
4.170	7.479	2.684	0.314	10.557	17.949	3.370	-3.018
2.100	1.588	1.763	1.585	2.060	1.763	1.670	-0.047
5.542	10.557	3.409	0.479	1.299	1.585	1.224	0.163
0.937	1.635	1.570	1.224	2.053	1.975	1.670	-0.047
1.262	2.002	2.276	1.475	2.522	3.370	5.448	2.084
-0.186	0.348	-0.047	0.163	-1.657	-3.018	2.084	3.399
-0.703	-1.754	1.175	1.227	10.557	17.949	3.370	-3.018
3.323	5.702	5.280	2.955	4.170	5.542	1.262	-0.703
7.186	11.636	2.837	-0.875	7.479	10.301	2.002	-1.754
5.532	9.277	8.483	4.488	2.684	3.409	2.276	1.175
11.036	21.787	4.009	-2.529	2.684	3.409	2.276	1.175
1.062	2.307	1.975	1.366	0.314	0.439	1.475	1.227
2.837	4.008	5.635	1.996	0.314	0.439	1.475	1.227
-0.632	-0.422	-0.325	0.055	2.053	1.975	1.670	-0.047
-0.875	-2.529	1.996	2.566	10.557	17.949	3.370	-3.018

MEAN VECTOR FOR WINTER WHEAT

48.115	58.212	56.478	25.965	33.611	30.274	52.106	30.611
37.876	36.177	52.389	27.221				

COVARIANCE MATRIX FOR SPRING WHEAT

5.189	5.108	2.732	4.586	10.115	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.017	2.857	2.480	1.103
1.504	1.584	0.322	1.689	3.092	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
5.108	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
2.413	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
4.586	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
2.108	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
2.732	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
1.843	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
-0.061	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
3.001	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
-0.697	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
4.164	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
-0.830	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
1.737	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
0.592	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
1.103	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
1.504	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
5.826	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
1.584	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
6.695	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
1.699	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
2.984	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683
0.322	10.758	2.732	10.115	14.021	7.500	1.451	1.315	1.531	10.148	3.105	2.117	2.857	2.480	1.103	1.683
0.939	2.953	0.322	3.092	3.935	3.613	3.753	5.532	2.164	6.229	3.105	2.117	2.857	2.480	1.103	1.683

MEAN VECTOR FOR SPRING WHEAT

48.600	59.467	28.178	39.133	38.644	57.555	31.622
43.911	47.822	51.556	24.644			

EXAMPLE 2: SAMPLE OUTPUT

COVARIANCE MATRIX FOR FALLOW

4.789	5.155	6.533	3.590	1.560	2.767	2.694	1.536
0.927	1.632	2.730	1.151				
5.355	10.520	11.781	6.431	2.631	4.321	4.613	2.251
1.254	1.918	2.545	0.969				
6.533	11.781	16.566	8.509	3.002	5.119	5.432	2.649
1.247	3.269	2.632	1.005				
3.590	6.431	8.509	5.328	1.642	3.006	3.065	1.520
1.186	2.280	2.175	1.009				
1.560	2.631	3.002	1.642	4.198	4.266	3.434	1.472
0.963	1.666	1.189	0.318				
2.767	4.321	5.119	3.006	4.266	8.346	4.958	1.091
0.902	1.873	2.325	0.715				
2.694	4.613	5.432	3.065	3.434	4.958	6.247	2.686
1.524	2.247	2.125	0.904				
1.536	2.251	2.649	1.520	1.472	1.991	2.686	1.802
1.109	1.371	1.715	0.523				
0.927	1.254	1.247	1.386	0.963	0.902	1.524	1.109
5.358	5.831	3.676	1.370				
1.632	1.518	3.269	2.289	1.665	1.873	2.247	1.371
5.831	11.992	2.817	0.162				
2.730	2.545	2.632	2.175	1.189	2.325	2.125	1.715
3.676	2.817	10.166	5.132				
1.151	0.969	1.005	1.009	0.318	0.715	0.904	0.820
1.370	0.162	5.132	3.755				

MEAN VECTOR FOR FALLOW

43.429	47.839	44.625	15.670	39.938	42.536	42.759	20.804
40.634	44.446	43.929	20.115				

EXAMPLE 2: SAMPLE OUTPUT

CLASS 1 (BETA=.16667) IS THE CONVEX COMBINATION OF:
CAREFY (ALPHA=.16667)

CLASS 2 (BETA=.16667) IS THE CONVEX COMBINATION OF:
STURBLE (ALPHA=.16667)

CLASS 3 (BETA=.16667) IS THE CONVEX COMBINATION OF:
GRASS (ALPHA=.16667)

CLASS 4 (BETA=.16667) IS THE CONVEX COMBINATION OF:
WINTER WHEAT (ALPHA=.16667)

CLASS 5 (BETA=.16667) IS THE CONVEX COMBINATION OF:
SPRING WHEAT (ALPHA=.16667)

CLASS 6 (BETA=.16667) IS THE CONVEX COMBINATION OF:
FALLOW (ALPHA=.16667)

ITERATION ROUTINE PARAMETERS

ESTIMATED MINIMUM= 0.30

ABSOLUTE ERROR= 1.0E-04

MAXIMUM NUMBER OF ITERATIONS=120

RESULTS FOR INITIAL B VECTOR

REGIONS
 -0.10800000D 31 <X> 0.15789930D 02 R 6(B)
 -0.16769939D 02 <Y> 0.31059086D 02 R 6(B)
 0.31059086D 02 <X> 0.34871944D 02 R 2(B)
 0.34871744D 02 <Y> 0.39174690D 02 R 3(B)
 0.39174680D 02 <X> 0.42576508D 02 R 1(B)
 0.42576508D 02 <Y> 0.46824740D 02 R 5(B)
 0.46824740D 02 <X> 0.10000000D 31 R 4(B)

CLASS TRANSFORMED COVARIANCES
 1 0.80151729D 01
 2 0.43127917D 01
 3 0.33681665D 01
 4 0.84073905D 01
 5 0.43905911D 01
 6 0.75067169D 01

TRANSFORMED MEANS
 0.4125850D 02
 0.32840881D 02
 0.36885123D 02
 0.49273239D 02
 0.44376685D 02
 0.25890481D 02

PROB. OF MISCLASS. FOR CURRENT B= 0.33513452D -00

VECTOR OF GRADIENTS

FNTX(1) = -0.21272580D -01 FNTX(2) = -0.89454502D -01 FNTX(3) = -0.73577369D -01
 FNTX(4) = -0.21580605D -01 FNTX(5) = 0.69311244D -01 FNTX(6) = 0.14035326D 00
 FNTX(7) = 0.64253965D -03 FNTX(8) = -0.25946563D -01 FNTX(9) = 0.13044153D 00
 FNTX(10) = 0.18656786D 00 FNTX(11) = 0.14677089D 00 FNTX(12) = 0.55028796D -01

B VECTOR

B(1) = 0.8450952D -01 B(2) = -0.61457250D -01 B(3) = 0.72522521D -01
 B(4) = 0.10225815D -00 B(5) = -0.30643836D -01 B(6) = 0.30991387D 00
 B(7) = 0.21287441D 00 B(8) = 0.27948874D 00 B(9) = 0.21736044D 00
 B(10) = -0.23874378D 00 B(11) = 0.14233649D 00 B(12) = 0.94580573D 00

RESULTS FOR FINAL R VECTOR

REGIONS

-0.10000000	31	<Y<	-0.16921787D 02	R 6(B)
-0.16921787D 02	<Y<	0.15065957D 02	R 6(B)	
0.15465957D 02	<Y<	0.18954399D 02	R 2(B)	
0.18954399D 02	<Y<	0.23169737D 02	R 3(B)	
0.23169737D 02	<Y<	0.26456050D 02	R 1(B)	
0.26456050D 02	<Y<	0.30761700D 02	R 5(B)	
0.30761700D 02	<Y<	0.10000000D 31	R 4(B)	

CONFUSION MATRIX

0.42326	0.00000	0.066746	0.00448	0.19259	0.00013
0.00383	0.79390	0.04936	0.00000	0.00000	0.41027
0.13266	0.05048	0.89717	0.00011	0.00455	0.04284
0.03027	0.00000	0.00000	0.86392	0.07368	0.00000
0.39854	0.00600	0.00002	0.13099	0.72918	0.00000
0.00003	0.15562	0.00001	0.00000	0.00000	0.54576

CLASS TRANSFORMED CC VARIANCES TRANSFORMED MEANS

1	0.70491653D 01	0.26027031D 02
2	0.17291263D 01	0.16757522D 02
3	0.16686674D 01	0.21087201D 02
4	0.85582699D 01	0.33974144D 02
5	0.35020589D 01	0.28050218D 02
6	0.47551349D 01	0.15209765D 02

PROB. OF MISCLASS. FOR CURRENT BE 0.29096567D 00

VECTOR OF GRADIENTS

FNTX(1)	= 0.17169653D 05	FNTX(2)	= 0.38087695D 06	FNTX(3)	= 0.19266560D 05
FNTX(4)	= 0.21001056D 05	FNTX(5)	= 0.11770099D 05	FNTX(6)	= 0.47552923D 06
FNTX(7)	= 0.12912892D 05	FNTX(8)	= 0.16973904D 05	FNTX(9)	= 0.60730688D 06
FNTX(10)	= -0.2649648D 05	FNTX(11)	= 0.29093131D 05	FNTX(12)	= -0.17567875D 05

VECTOR

B(1)	= 0.15755546D 02	B(2)	= 0.15667465D 00	B(3)	= 0.80487594D 01
B(4)	= 0.76725571D 01	B(5)	= -0.77757683D 01	B(6)	= -0.25460085D 00
B(7)	= 0.56779286D 01	B(8)	= 0.35320330D 00	B(9)	= 0.65529241D 01
B(10)	= -0.27429676D 00	B(11)	= -0.43785646D 01	B(12)	= 0.82265704D 00

CLASS 1 (BETA=.3333) IS THE CONVEX COMBINATION OF:
WINTER WHEAT (ALPHA=.16667)
SPRING WHEAT (ALPHA=.16667)

CLASS 2 (BETA=.3333) IS THE CONVEX COMBINATION OF:
BARLEY (ALPHA=.16667)

CLASS 3 (BETA=.3333) IS THE CONVEX COMBINATION OF:
STUBBLE (ALPHA=.16667)
GRASS (ALPHA=.16667)
FALLOW (ALPHA=.16667)

ITERATION ROUTINE PARAMETERS

ESTIMATED MINIMUM= 0.15
ABSOLUTE ERROR= 1.0E-04
MAXIMUM NUMBER OF ITERATIONS=120

RESULTS FOR INITIAL B VECTOR

REGIONS
 -0.10000000 31 <Y< 0.29575426D 02 R 3(B)
 0.29575426D 32 <Y< 0.37922830D 02 R 3(B)
 0.37922830D 02 <Y< 0.43770935D 02 R 2(B)
 0.43770935D 02 <Y< 0.43778099D 02 R 2(B)
 0.43778099D 02 <Y< 0.10000000D 31 R 1(B)

CLASS	TRANSFORMED	COVARIANCES	TRANSFORMED	MEANS
1	0.84074033D 01		0.49273145D 02	
2	0.43905507D 01		0.44378579D 02	
3	0.40161799D 01		0.41526243D 02	
4	0.43127939D 01		0.32340761D 02	
5	0.72681024D 01		0.36385004D 02	
6	0.75047057D 01		0.29940366D 02	

PROB. OF MISCLASS. FOR CURRENT R= 0.20699933D 00

VECTOR OF GRADIENTS
 FNTX(1) = -0.11935191D 00 FNTX(2) = -0.34146536D 00 FNTX(3) = -0.37735146D 00
 FNTX(4) = -0.19702452D 00 FNTX(5) = -0.11203286D 00 FNTX(6) = -0.24832357D 00
 FNTX(7) = -0.21045761D 00 FNTX(8) = -0.77081239D -01 FNTX(9) = -0.73156751D -01
 FNTX(10) = -0.21323372D 00 FNTX(11) = -0.77872775D -01 FNTX(12) = -0.61331184D -02

B VECTOR
 B(1) = 0.84503293D -01 B(3) = 0.72522598D -01
 B(4) = 0.10225540D 00 B(6) = -0.30991679D 00
 B(7) = 0.21267709D 00 B(9) = 0.21735668D 00
 B(10) = -0.23874207D 00 B(11) = 0.14233905D 00 B(12) = 0.94580454D 00

EXAMPLE 2: SAMPLE OUTPUT

NO CONVERGENCE IN LIMIT ITERATIONS

RESULTS FOR FINAL B VECTOR

REGIONS
 -0.10000000 31 <YK 0.16806767D 02 R 3(B)
 0.16806767D 02 <YK 0.22583828D 02 R 3(B)
 0.22583828D 02 <YK 0.22587138D 02 R 2(B)
 0.22587138D 02 <YK 0.27585809D 02 R 2(B)
 0.27585809D 02 <YK 0.10000000D 31 R 1(B)

CONFUSION MATRIX

0.70325 0.07457 C.00000
 0.28480 C.91463 0.02026
 0.01194 0.01180 C.97977

CLASS	TRANSFORMED COVARIANCES	TRANSFORMED MEANS
1	C.577706130 01	0.31607103D 02
2	0.56671685D 01	0.27308244D 02
3	C.18144293D 01	0.25633086D 02
4	0.23850040D 01	0.18324583D 02
5	0.35073301D 01	0.19492336D 02
6	0.64346347D 01	0.16255430D 02

PROB. OF MISCLASS. FOR CURRENT 9= 0.13412749D C0

VECTOR OF GRADIENTS

FNTX(1) = 0.14537053D-04 FNTX(2) = 0.94215900D-06 FNTX(3) = 0.21373493D-04
 FNTX(4) = 0.19106154D-04 FNTX(5) = -0.83569562D-05 FNTX(6) = 0.17969361D-04
 FNTX(7) = 0.22199217D-04 FNTX(8) = 0.86391871D-C5 FNTX(9) = 0.36057012D-05
 FNTX(10) = C.56999312D-05 FNTX(11) = 0.11901647D-04 FNTX(12) = 0.25177997D-04

B VECTOR

B(1) = 0.33664692D-01 B(2) = -0.25145311D-01 B(3) = 0.15466674D 00
 B(4) = 0.20115621D C0 B(5) = -0.44658552D 00 B(6) = 0.17222076D 00
 B(7) = 0.11698609D 00 B(8) = 0.24272545D 00 B(9) = -0.32537082D 00
 B(10) = -0.10429556D C0 B(11) = 0.21622407D 00 B(12) = 0.68327226D C0

REFERENCES

1. T. W. Anderson, An Introduction to Multivariate Statistical Analysis, John Wiley and Sons, New York, 1958.
2. R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization, *Comput. J.* 6(1963), 163-168.
3. L. F. Guseman, Jr., and Homer F. Walker, On minimizing the probability of misclassification for linear feature selection, JSC Internal Technical Note, JSC-08412, Johnson Space Center, Houston, Texas, August, 1973.
4. L. F. Guseman, Jr., B. Charles Peters, Jr. and Homer F. Walker, On minimizing the probability of misclassification for linear feature selection, *Ann. Statist.* 3(1975), 661-668.
5. L. F. Guseman, Jr., and Bruce P. Marion, Obtaining initial vectors for minimizing the probability of misclassification, NASA Contract NAS-9-13894, Texas A&M University, Department of Mathematics, Report #1, Sept. 1974.
6. L. F. Guseman, Jr. and Bruce P. Marion, LFSPMC: Linear Feature Selection Program Using the Probability of Misclassification, NASA Contract NAS-9-13894, Texas A&M University, Department of Mathematics, Report #3, January, 1975.
7. D. G. Luenberger, Optimization by Vector Space Methods, John Wiley and Sons, New York, 1969.
8. W. Robert Mann, Mean value methods in iteration, *Proc. Amer. Math. Soc.* 4(1953), 506-510.

9. J. L. Solomon and B. Charles Peters, Jr., A simplified version of locating stationary points of the transformed probability of correct classification, Preprint.
10. System/360 Scientific Subroutine Package, Version III, 5th Edition, Form GH20-0205-4, International Business Machines, August, 1970.

METHODS FOR ESTIMATING PROPORTIONS
OF CONVEX COMBINATIONS OF NORMALS

By

L. F. Guseman, Jr. and Jay R. Walton

Department of Mathematics
Texas A&M University

Report #7

Prepared For

Earth Observations Division
NASA/Johnson Space Center
Houston, Texas
Contract NAS-9-14689-4S

April, 1976

METHODS FOR ESTIMATING PROPORTIONS OF
CONVEX COMBINATIONS OF NORMALS

L. F. Guseman, Jr. and Jay R. Walton

1. INTRODUCTION

The techniques in this report were developed to treat the following problem:

Given LANDSAT data (multitemporal) over a LACIE sample segment, and representative samples of the classes present in the segment, estimate what proportion of the sample segment is in WHEAT.

The approach to the above problem taken in this report can be summarized as follows:

- (a) The density function for each of the classes WHEAT and NON-WHEAT is expressed as an appropriate convex combination of multivariate normal densities with known means and covariance matrices (usually obtained from a training sample).
- (b) Feature selection is performed to produce a corresponding optimal one-dimensional Bayes classifier for WHEAT VS. NON-WHEAT whose associated confusion matrix is known.
- (c) Using the one-dimensional classifier and associated confusion matrix, an estimate of the true proportion

of WHEAT in the sample segment is made. In some cases, the resulting estimate is unbiased.

The methods presented herein have the advantage that classification of multitemporal data over the sample segment is performed optimally in one-dimensional space thereby reducing significantly the computer time spent in classification. Another advantage is that the confusion matrix, used in obtaining unbiased estimates of the wheat proportions, can be readily computed without the costly requirement of obtaining a labeled sample independent of the training sample.

The general mathematical framework of proportion estimation procedures is discussed in Sections 2 and 3. In Section 4 we define four particular estimators based on the previous mathematical discussion. Section 5 contains preliminary numerical results of the four estimators presented in Section 4.

2. PRELIMINARY MATHEMATICAL DISCUSSION

Let Π_1, \dots, Π_m be distinct classes with true (but unknown)

a priori probabilities $\alpha_1, \dots, \alpha_m$, respectively. Let $\Omega = \bigcup_{i=1}^m \Pi_i$

and let $X : \Omega \rightarrow R^n$ be a random vector with mixture density

$$f = f_X = \sum_{i=1}^m \alpha_i f_i ,$$

where each class conditional density function $f_i = f_{X/\Pi_i}$ is

assumed to be $N(\mu_i, \Sigma_i)$ with μ_i, Σ_i known, $1 \leq i \leq m$.

A method of estimating all m a priori probabilities $\alpha_1, \dots, \alpha_m$ in the mixture f from a given sample was discussed in [1]. The estimation problem discussed in [1] resulted in solving the problem:

$$\text{minimize } \|P\alpha - \hat{e}\| \quad (\text{Euclidean norm})$$

$$\text{subject to } \sum_{i=1}^m \alpha_i = 1, \quad \alpha_i \geq 0, \quad 1 \leq i \leq m$$

where \hat{e} is the m -dimensional vector of proportions obtained by classifying a random sample of size N , and P is the $m \times m$ error matrix associated with the classifier used to obtain \hat{e} ; that is,

$P = (p_{ij})$, where

$$p_{ij} = \int_{R_j} f_j(x) dx \quad i, j = 1, 2, \dots, m$$

and R_1, \dots, R_m are given classification regions. If P is invertible, then $\hat{\alpha} = P^{-1}\hat{e}$ is an unbiased estimate of α (which, in general, may not satisfy the nonnegativity constraints).

The following discussion forms the basis for the estimation procedures presented in the sequel.

Let $\Gamma_1 = \bigcup_{i=1}^k \Pi_i$, $\Gamma_2 = \bigcup_{i=k+1}^m \Pi_i$. Then Γ_1 and Γ_2 have a

priori probabilities $\gamma_1 = \alpha_1 + \dots + \alpha_k$ and $\gamma_2 = \alpha_{k+1} + \dots + \alpha_m$,

respectively. Letting

$$h_1 = \sum_{i=1}^k \frac{\alpha_i}{\gamma_1} f_i$$

and

$$h_2 = \sum_{i=k+1}^m \frac{\alpha_i}{\gamma_2} f_i,$$

we have

$$\begin{aligned} f &= \sum_{i=1}^m \alpha_i f_i \\ &= \sum_{i=1}^k \alpha_i f_i + \sum_{i=k+1}^m \alpha_i f_i \\ &= \gamma_1 \sum_{i=1}^k \frac{\alpha_i}{\gamma_1} f_i + \gamma_2 \sum_{i=k+1}^m \frac{\alpha_i}{\gamma_2} f_i \\ &= \gamma_1 h_1 + \gamma_2 h_2. \end{aligned}$$

We note that the density functions h_1 and h_2 are convex combinations of normals. Henceforth it will be convenient to denote the above expression for f as a convex combinations of convex combinations by h ; that is $h = \gamma_1 h_1 + \gamma_2 h_2$. Suppose we are given decision regions S_1 and S_2 and associated decision function $C : \Omega \rightarrow \{1,2\}$ defined for each $\omega \in \Omega$ by

$$C(\omega) = i \text{ iff } X(\omega) \in S_i, \quad i = 1, 2.$$

Then the probability that $\omega \in \Omega$ is classified as belonging to Γ_i is given by

$$\begin{aligned} \Pr([X \in S_i]) &= \Pr([X \in S_i] \cap (\Gamma_1 \cup \Gamma_2)) \\ &= \Pr\left(\bigcup_{j=1}^2 ([X \in S_i] \cap \Gamma_j)\right) \\ &= \sum_{j=1}^2 \Pr([X \in S_i] \cap \Gamma_j) \\ &= \sum_{j=1}^2 \gamma_j \Pr([X \in S_i] | \Gamma_j) \\ &= \gamma_1 \Pr([X \in S_i] | \Gamma_1) + \gamma_2 \Pr([X \in S_i] | \Gamma_2) \end{aligned}$$

Let $Z = (Z_1, Z_2)^T$, where $Z_i = \chi_{S_i} \circ X$ and χ_{S_i} is the characteristic function of $S_i \subseteq R^n$. For a fixed i ,

$$\begin{aligned}
 E(Z_i) &= E(\chi_{S_i}(X)) \\
 &= \int_{\mathbb{R}^n} \chi_{S_i} h(x) dx \\
 &= \int_{S_i} h(x) dx \\
 &= \int_{S_i} (\gamma_1 h_1(x) + \gamma_2 h_2(x)) dx \\
 &= \gamma_1 \int_{S_i} h_1(x) dx + \gamma_2 \int_{S_i} h_2(x) dx \\
 &= \gamma_1 P([X \in S_i] | \Gamma_1) + \gamma_2 P([X \in S_i] | \Gamma_2)
 \end{aligned}$$

Let $\omega^N = (\omega_1, \omega_2, \dots, \omega^N)$ be a random sample of size N from Ω . For each fixed i , $i = 1, 2$, let

$$Z_{ir}(\omega^N) = Z_i(\omega_r) \quad , \quad 1 \leq r \leq N$$

Then for each fixed i , Z_{i1}, \dots, Z_{iN} are independent random variables and each has the same distribution as Z_i (see [4]). Thus $E(Z_{ir}) = E(Z_i)$,

$1 \leq r \leq N$. Letting $\hat{d}_i = \frac{1}{N} \sum_{r=1}^N Z_{ir}$, we have $\hat{d}_i(\omega^N) = \frac{N_i}{N}$, where

N_i is the number of elements in ω^N that are classified as being from

Γ_i . Letting $d_i = E(\hat{d}_i)$, we have

$$\begin{aligned}
 d_i &= E(\hat{d}_i) = E\left(\frac{1}{N} \sum_{r=1}^N Z_{ir}\right) \\
 &= \frac{1}{N} \sum_{r=1}^N E(Z_{ir}) = \frac{1}{N} \sum_{r=1}^N E(Z_i) \\
 &= E(Z_i) = \gamma_1 P([X \in S_i] | \Gamma_1) + \gamma_2 P([X \in S_i] | \Gamma_2) .
 \end{aligned}$$

Then, if

$$\gamma = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}, \quad d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \quad \text{and} \quad \hat{d} = \begin{pmatrix} \hat{d}_1 \\ \hat{d}_2 \end{pmatrix}$$

we obtain

$$d = E(\hat{d}) = Q\gamma,$$

where Q is the 2×2 matrix whose entry q_{ij} , in the i^{th} row and j^{th} column, is given by

$$q_{ij} = \int_{S_i} h_j(x) dx \quad i, j = 1, 2.$$

The error matrix Q is determined by the classification regions S_1 and S_2 and the true class conditional density functions h_1 and h_2 for Γ_1 and Γ_2 , respectively; that is, h_1 and h_2 in terms of the true a priori probabilities, $\alpha_1, \alpha_2, \dots, \alpha_m$. If, in addition, the regions S_1 and S_2 are to be determined by h_1 and h_2 , as in Bayesian classification, then S_1 and S_2 are also functions of $\alpha_1, \dots, \alpha_n$.

3. GENERAL PROCEDURE FOR DEFINING ESTIMATORS

Each of the estimators presented in Section 4 results from using the following general procedure to determine appropriate decision regions and associated error matrices.

Begin by specifying values of $\beta = (\beta_1, \beta_2)^T$, $\beta_1 \geq 0$, $\beta_2 \geq 0$, $\beta_1 + \beta_2 = 1$, and $\eta = (\eta_1, \dots, \eta_m)^T$, $\eta_i \geq 0$, $1 \leq i \leq m$, $\sum_{i=1}^m \eta_i = 1$,

to produce approximating conditional densities

$$h_1(\eta : x) = \sum_{i=1}^k \frac{\eta_i}{v_1} f_i(x) ; v_1 = \sum_{i=1}^k \eta_i$$

$$h_2(\eta : x) = \sum_{i=k+1}^m \frac{\eta_i}{v_2} f_i(x) ; v_2 = \sum_{i=k+1}^m \eta_i$$

and approximating mixture density

$$h(\beta, \eta : x) = \beta_1 h_1(\eta : x) + \beta_2 h_2(\eta : x) .$$

Once β and η have been specified, an existing feature selection technique is used (see [2]) to produce a $1 \times n$ vector B^* of norm one which minimizes the transformed probability of misclassification in one dimension; that is, a B^* with $\|B^*\| = 1$ is found which minimizes the function g (of B only) defined by

$$g(B) = \beta_1 \int_{S_2(B)} h_1(\eta : y, B) dy + \beta_2 \int_{S_1(B)} h_2(\eta : y, B) dy ,$$

where the transformed densities (as a function of nonzero B) are defined for $y \in R^1$ by

$$f_i(y, B) = \frac{1}{(2\pi)^{1/2} (B\Sigma_i B^T)^{1/2}} \exp\left(-\frac{1}{2} \frac{(y - B\mu_i)^2}{B\Sigma_i B^T}\right), \quad i = 1, 2, \dots, m,$$

$$h_1(\eta : y, B) = \sum_{i=1}^k \frac{\eta_i}{v_1} f_i(y, B)$$

$$h_2(\eta : y, B) = \sum_{i=k+1}^m \frac{\eta_i}{v_2} f_i(y, B)$$

and the associated Bayes decision regions are given by

$$S_1(B) = \{y \in R^1 : \beta_1 h_1(\eta : y, B) \geq \beta_2 h_2(\eta : y, B)\}.$$

$$S_2(B) = \{y \in R^1 : \beta_2 h_2(\eta : y, B) > \beta_1 h_1(\eta : y, B)\}.$$

The resulting associated error matrix Q at B^* is given by

$Q = (q_{ij})$, where

$$q_{ij} = \int_{S_j(B^*)} h_j(\eta : y, B^*) dy, \quad i, j = 1, 2.$$

Having determined the decision regions $S_1(B^*)$ and $S_2(B^*)$, the

classification rule

$$(*) \quad C(\omega) = i \quad \text{iff} \quad B^*(X(\omega)) \in S_i(B^*)$$

is used to classify the random sample ω^N to produce a 2-dimensional vector of (classification) proportions.

4. SPECIFIC ESTIMATORS

We now discuss several methods for estimating γ . The first (ESTIMATOR 1) is nothing more than an obvious modification of the m-class estimator discussed in Section 2 and [1]. It is used in the subsequent discussion to obtain other estimators.

ESTIMATOR 1

The procedure for estimating $\alpha = (\alpha_1, \dots, \alpha_m)^T$ discussed in Section 2 and [1] produces a vector $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_m)^T$ minimizing $\|P\alpha - \hat{e}\|$ and satisfying $\sum_{i=1}^m \hat{\alpha}_i = 1$, $\hat{\alpha}_i \geq 0$, $1 \leq i \leq m$. Then

$$\begin{aligned} \hat{\gamma} &= \begin{pmatrix} \hat{\gamma}_1 \\ \hat{\gamma}_2 \end{pmatrix} = \begin{pmatrix} \hat{\alpha}_1 + \dots + \hat{\alpha}_k \\ \hat{\alpha}_{k+1} + \dots + \hat{\alpha}_m \end{pmatrix} \\ &= \begin{pmatrix} \overbrace{1 \dots 1}^k & 0 \dots 0 \\ 0 \dots 0 & \underbrace{1 \dots 1}_{m-k} \end{pmatrix} \begin{pmatrix} \hat{\alpha}_1 \\ \vdots \\ \hat{\alpha}_m \end{pmatrix} \\ &= A\hat{\alpha} \end{aligned}$$

is an estimator for γ . If P is invertible and $P^{-1}\hat{e}$ satisfies the nonnegativity constraints, then $\hat{\alpha} = P^{-1}\hat{e}$ and $\hat{\gamma} = AP^{-1}\hat{e}$ is an unbiased estimate of γ .

ESTIMATOR 2

Using the m-class estimator $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_m)^T$, and $\hat{\gamma} = (\hat{\gamma}_1, \hat{\gamma}_2)^T$ we obtain the conditional densities

$$\hat{h}_1(x) = h_1(\hat{\alpha} : x) = \sum_{i=1}^k \frac{\hat{\alpha}_i}{\hat{\gamma}_1} f_i(x)$$

$$\hat{h}_2(x) = h_2(\hat{\alpha} : x) = \sum_{i=k+1}^m \frac{\hat{\alpha}_i}{\hat{\gamma}_2} f_i(x)$$

and mixture density

$$\begin{aligned} \hat{h}(x) &= h(\hat{\gamma}, \hat{\alpha} : x) \\ &= \hat{\gamma}_1 h_1(\hat{\alpha} : x) + \hat{\gamma}_2 h_2(\hat{\alpha} : x) \\ &= \hat{\gamma}_1 \hat{h}_1(x) + \hat{\gamma}_2 \hat{h}_2(x) . \end{aligned}$$

The subsequent feature selection produces (at the minimizing B^*) the decision regions

$$\hat{S}_1 = \{y \in R^1 : \hat{\gamma}_1 \hat{h}_1(y, B^*) \geq \hat{\gamma}_2 \hat{h}_2(y, B^*)\}$$

$$\hat{S}_2 = \{y \in R^1 : \hat{\gamma}_2 \hat{h}_2(y, B^*) > \hat{\gamma}_1 \hat{h}_1(y, B^*)\}$$

and associated error matrix $\hat{Q} = (\hat{q}_{ij})$, where

$$\hat{q}_{ij} = \int_{\hat{S}_i} \hat{h}_j(y, B^*) dy, \quad i, j = 1, 2.$$

Using the classification rule (*) with the decision regions \hat{S}_1, \hat{S}_2 to

classify the random sample $\omega^N = (\omega_1, \dots, \omega_N)$, we obtain the vector

of proportions

$$\hat{d} = \begin{pmatrix} \hat{d}_1 \\ \hat{d}_2 \end{pmatrix} .$$

The resulting problem

$$\text{minimize } \|\hat{Q}\xi - \hat{d}\|$$

$$\text{subject to } \xi_1 + \xi_2 = 1, \quad \xi_i \geq 0, \quad i = 1, 2$$

is readily solved to produce the estimator $\hat{\gamma}$ of γ . When \hat{Q} is invertible and $\hat{Q}^{-1}\hat{d}$ satisfies the nonnegativity constraints, then

$$\hat{\gamma} = \hat{Q}^{-1}\hat{d}.$$

ESTIMATOR 3

Letting $\eta = (\frac{1}{m}, \dots, \frac{1}{m})^T$, $\beta = (\frac{1}{2}, \frac{1}{2})^T$ we obtain the

conditional densities

$$\bar{h}_1(x) = h_1(\eta : x) = \frac{1}{k} \sum_{i=1}^k f_i(x)$$

$$\bar{h}_2(x) = h_2(\eta : x) = \frac{1}{m-k} \sum_{i=k+1}^m f_i(x),$$

and the mixture density

$$\begin{aligned} \bar{h}(x) &= h(\eta, \beta : x) \\ &= \frac{1}{2} h_1(\eta : x) + \frac{1}{2} h_2(\eta : x) \\ &= \frac{1}{2} \bar{h}_1(x) + \frac{1}{2} \bar{h}_2(x). \end{aligned}$$

The subsequent feature selection produces (at the minimizing B^*) the decision regions

$$\bar{S}_1 = \{y \in R^1 : \bar{h}_1(y, B^*) \geq \bar{h}_2(y, B^*)\}$$

$$\bar{S}_2 = \{y \in R^1 : \bar{h}_2(y, B^*) > \bar{h}_1(y, B^*)\}$$

and associated error matrix $\bar{Q} = (\bar{q}_{ij})$ where

$$\bar{q}_{ij} = \int_{\bar{S}_i} \bar{h}_j(y, B^*) dy, \quad i, j = 1, 2.$$

Using the classification rule (*) with the decision regions \bar{S}_1, \bar{S}_2 , we obtain the vector of proportions

$$\bar{d} = \begin{pmatrix} \bar{d}_1 \\ \bar{d}_2 \end{pmatrix}.$$

The resulting problem

$$\text{minimize } \|\bar{Q}\xi - \bar{d}\|$$

$$\text{subject to } \xi_1 + \xi_2 = 1, \quad \xi_i \geq 0, \quad i = 1, 2$$

is readily solved to produce the estimator $\bar{\gamma}$ of γ . When \bar{Q} is invertible and $\bar{Q}^{-1}\bar{d}$ satisfies the nonnegativity constraints, then $\bar{\gamma} = \bar{Q}^{-1}\bar{d}$.

ESTIMATOR 4

Using the decision regions \bar{S}_1, \bar{S}_2 and B^* determined in ESTIMATOR 3, and conditional densities \hat{h}_1, \hat{h}_2 from ESTIMATOR 2, let $Q^* = (q_{ij}^*)$, where

$$q_{ij}^* = \int_{\bar{S}_i} \hat{h}_j(y, B^*) dy, \quad i, j = 1, 2 \quad .$$

When \bar{Q} is invertible let

$$\gamma^* = \hat{\gamma} \pm (\bar{\gamma} - \bar{Q}^{-1} Q^* \hat{\gamma}) \quad .$$

Then γ^* is an estimator of γ . If $\bar{\gamma} = \bar{Q}^{-1} \bar{d}$ and $\hat{\gamma} = AP^{-1} \hat{e}$, then

$$\gamma^* = \hat{\gamma} \pm \bar{Q}^{-1} (\bar{d} - Q^* AP^{-1} \hat{e}) \quad ,$$

and, for either choice of sign, γ^* is an unbiased estimator of γ .

5. PRELIMINARY NUMERICAL RESULTS

The four ESTIMATORS discussed in Section 4 have been implemented (FORTRAN PROGRAM ESTPRO) and undergone preliminary testing. Testing was accomplished using 16-dimensional data from four registered passes (May 5, May 23, June 11, June 29, 1973) of LANDSAT 1 MSS measurements acquired over Hill County (N), Montana. Training data was provided to the program for the five classes: WHEAT, FALLOW, BARLEY, GRASS, STUBBLE. A random sample of 16-dimensional vectors of size 2417 comprised of the above five classes in the following proportions was used:

<u>CLASS</u>	<u># OF VECTORS</u>	<u>TRUE PROPORTION</u>
WHEAT	784	.3244
FALLOW	244	.3078
BARLEY	300	.1241
GRASS	206	.0852
STUBBLE	383	.1585

The results for ESTIMATORS 1-4 in estimating the proportions of WHEAT VS. NONWHEAT appear in Tables 1-4. The estimated proportions obtained from ESTIMATOR 4 using both plus and minus signs are included. Results for estimating the proportions of BARLEY VS. NON-BARLEY appear in Tables 5-8. The results from all four estimators are summarized in Table 9 for WHEAT VS. NON-WHEAT and Table 10 for BARLEY VS. NON-BARLEY.

The feature selection program used in ESTPRO is LFSPMC(VERSION 2) discussed in [2]. Solution of the constrained least squares problem (when needed) is accomplished using LSI from [3].

$$P = \begin{pmatrix} 0.71597 & 0.00182 & 0.19020 & 0.08876 & 0.01057 \\ 0.00463 & 0.64462 & 0.00018 & 0.12748 & 0.45221 \\ 0.12068 & 0.00000 & 0.80158 & 0.00000 & 0.00002 \\ 0.15869 & 0.17698 & 0.00805 & 0.78348 & 0.16139 \\ 0.00002 & 0.17658 & 0.00000 & 0.00028 & 0.37581 \end{pmatrix} \quad \text{Error Matrix}$$

$$\hat{e} = \begin{pmatrix} .27927 \\ .26107 \\ .13860 \\ .18866 \\ .13240 \end{pmatrix} \quad \text{Classification vector}$$

$$\hat{\alpha} = \begin{pmatrix} .34499 \\ .21119 \\ .12096 \\ .06987 \\ .25299 \end{pmatrix} \quad \begin{array}{l} \text{WHEAT} \\ \text{FALLOW} \\ \text{BARLEY} \\ \text{GRASS} \\ \text{STUBBLE} \end{array}$$

$$\hat{\gamma} = \begin{pmatrix} .34499 \\ .65501 \end{pmatrix} \quad \left. \begin{array}{l} \text{WHEAT} \\ \text{NON-WHEAT} \end{array} \right\} \text{ESTIMATOR 1}$$

Table 1. ESTIMATOR 1: WHEAT VS. NON-WHEAT

$$\hat{\gamma} = \begin{pmatrix} .34499 \\ .65501 \end{pmatrix} \quad \left. \begin{array}{l} \text{WHEAT} \\ \text{NON-WHEAT} \end{array} \right\} \quad \text{From ESTIMATOR 1}$$

$$\hat{Q} = \begin{pmatrix} .90271 & .07694 \\ .08729 & .92306 \end{pmatrix} \quad \text{Error Matrix}$$

$$\hat{d} = \begin{pmatrix} .34050 \\ .65950 \end{pmatrix} \quad \text{Classification Vector}$$

$$\hat{\gamma} = \begin{pmatrix} .31918 \\ .68082 \end{pmatrix} \quad \left. \begin{array}{l} \text{WHEAT} \\ \text{NON-WHEAT} \end{array} \right\} \quad \text{ESTIMATOR 2}$$

Table 2. ESTIMATOR 2: WHEAT VS. NON-WHEAT

$$\bar{Q} = \begin{pmatrix} .93341 & .11058 \\ .06659 & .88492 \end{pmatrix} \quad \text{Error Matrix}$$

$$\bar{d} = \begin{pmatrix} .37319 \\ .62681 \end{pmatrix} \quad \text{Classification Vector}$$

$$\bar{Y} = \begin{pmatrix} .31541 \\ .68459 \end{pmatrix} \quad \left. \begin{array}{l} \text{WHEAT} \\ \text{NON-WHEAT} \end{array} \right\} \quad \text{ESTIMATOR 3}$$

Table 3. ESTIMATOR 3: WHEAT VS. NON-WHEAT

$$\hat{\gamma} = \begin{pmatrix} .34499 \\ .65501 \end{pmatrix} \left. \begin{array}{l} \text{WHEAT} \\ \text{NON-WHEAT} \end{array} \right\} \text{From ESTIMATOR 1}$$

$$\bar{\gamma} = \begin{pmatrix} .31541 \\ .68459 \end{pmatrix} \left. \begin{array}{l} \text{WHEAT} \\ \text{NON-WHEAT} \end{array} \right\} \text{From ESTIMATOR 3}$$

$$\bar{Q} = \begin{pmatrix} .93341 & .11058 \\ .06659 & .88492 \end{pmatrix} \text{Error Matrix From ESTIMATOR 3}$$

$$\bar{d} = \begin{pmatrix} .36318 \\ .62681 \end{pmatrix} \text{Classification Vector From ESTIMATOR 3}$$

$$Q^* = \begin{pmatrix} .93341 & .10348 \\ .06659 & .89652 \end{pmatrix} \text{Error Matrix}$$

$$\gamma^*(\text{plus sign}) = \begin{pmatrix} .32470 \\ .67530 \end{pmatrix} \left. \begin{array}{l} \text{WHEAT} \\ \text{NON-WHEAT} \end{array} \right\} \text{ESTIMATOR 4}$$

$$\gamma^*(\text{minus sign}) = \begin{pmatrix} .36529 \\ .63471 \end{pmatrix} \left. \begin{array}{l} \text{WHEAT} \\ \text{NON-WHEAT} \end{array} \right\} \text{ESTIMATOR 4}$$

Table 4. ESTIMATOR 4: WHEAT VS. NON-WHEAT

$$P = \begin{pmatrix} 0.71597 & 0.00182 & 0.19020 & 0.08876 & 0.01057 \\ 0.00463 & 0.64462 & 0.00018 & 0.12748 & 0.45221 \\ 0.12068 & 0.00000 & 0.80158 & 0.00000 & 0.00002 \\ 0.15869 & 0.17698 & 0.00805 & 0.78348 & 0.16139 \\ 0.00002 & 0.17658 & 0.00000 & 0.00028 & 0.37581 \end{pmatrix} \quad \text{Error Matrix}$$

$$\hat{e} = \begin{pmatrix} .27927 \\ .26107 \\ .13860 \\ .18866 \\ .13240 \end{pmatrix} \quad \text{Classification vector}$$

$$\hat{a} = \begin{pmatrix} .34499 \\ .21119 \\ .12096 \\ .06987 \\ .25299 \end{pmatrix} \quad \begin{array}{l} \text{WHEAT} \\ \text{FALLOW} \\ \text{BARLEY} \\ \text{GRASS} \\ \text{STUBBLE} \end{array}$$

$$\hat{\gamma} = \begin{pmatrix} .12096 \\ .87904 \end{pmatrix} \quad \left. \begin{array}{l} \text{BARLEY} \\ \text{NON-BARLEY} \end{array} \right\} \text{ESTIMATOR 1}$$

Table 5. ESTIMATOR 1: BARLEY VS. NON-BARLEY

$$\hat{\gamma} = \left(\begin{array}{l} .12096 \\ .87904 \end{array} \right) \left. \begin{array}{l} \text{BARLEY} \\ \text{NON-BARLEY} \end{array} \right\} \text{From ESTIMATOR 1}$$

$$\hat{Q} = \left(\begin{array}{cc} .93122 & .00805 \\ .06878 & .99195 \end{array} \right) \text{Error Matrix}$$

$$\hat{d} = \left(\begin{array}{l} .12329 \\ .87671 \end{array} \right) \text{Classification Vector}$$

$$\hat{\hat{\gamma}} = \left(\begin{array}{l} .12484 \\ .87516 \end{array} \right) \left. \begin{array}{l} \text{BARLEY} \\ \text{NON-BARLEY} \end{array} \right\} \text{ESTIMATOR 2}$$

Table 6. ESTIMATOR 2: BARLEY VS. NON-BARLEY

$$\bar{Q} = \begin{pmatrix} .98549 & .02169 \\ .01451 & .97831 \end{pmatrix} \quad \text{Error Matrix}$$

$$\bar{d} = \begin{pmatrix} .14522 \\ .85478 \end{pmatrix} \quad \text{Classification Vector}$$

$$\bar{y} = \begin{pmatrix} .12817 \\ .87183 \end{pmatrix} \quad \left. \begin{array}{l} \text{BARLEY} \\ \text{NON-BARLEY} \end{array} \right\} \quad \text{ESTIMATOR 3}$$

Table 7. ESTIMATOR 3: BARLEY VS. NON-BARLEY

$$\gamma = \left(\begin{array}{l} .12096 \\ .87904 \end{array} \right) \left. \begin{array}{l} \text{BARLEY} \\ \text{NON-BARLEY} \end{array} \right\} \text{From ESTIMATOR 1}$$

$$\bar{\gamma} = \left(\begin{array}{l} .12817 \\ .87183 \end{array} \right) \left. \begin{array}{l} \text{BARLEY} \\ \text{NON-BARLEY} \end{array} \right\} \text{From ESTIMATOR 3}$$

$$\bar{Q} = \left(\begin{array}{cc} .98549 & .02169 \\ .01451 & .97831 \end{array} \right) \text{Error Matrix From ESTIMATOR 3}$$

$$\bar{d} = \left(\begin{array}{l} .14522 \\ .85478 \end{array} \right) \text{Classification Vector From ESTIMATOR 3}$$

$$Q^* = \left(\begin{array}{cc} .98549 & .03115 \\ .01451 & .96885 \end{array} \right) \text{Error Matrix}$$

$$\gamma^*(\text{plus sign}) = \left(\begin{array}{l} .11955 \\ .88045 \end{array} \right) \left. \begin{array}{l} \text{BARLEY} \\ \text{NON-BARLEY} \end{array} \right\} \text{ESTIMATOR 4}$$

$$\gamma^*(\text{minus sign}) = \left(\begin{array}{l} .12238 \\ .87762 \end{array} \right) \left. \begin{array}{l} \text{BARLEY} \\ \text{NON-BARLEY} \end{array} \right\} \text{ESTIMATOR 4}$$

Table 8. ESTIMATOR 4: BARLEY VS. NON-BARLEY

	TRUE	EST 1	EST 2	EST 3	EST 4(+)	EST 4(-)
WHEAT	.3244	.3450	.3192	.3154	.3247	.3653
NON-WHEAT	.6756	.6550	.6808	.6846	.6753	.6347

Table 9. ESTIMATORS 1-4: WHEAT VS. NON-WHEAT

	TRUE	EST 1	EST 2	EST 3	EST 4(+)	EST 4(-)
BARLEY	.1241	.1210	.1248	.1282	.1196	.1224
NON-BARLEY	.8759	.8790	.8752	.8718	.8805	.8776

Table 10. ESTIMATORS 1-4: BARLEY VS. NON-BARLEY

REFERENCES

1. L. F. Guseman, Jr. and Bruce P. Marion, A Method for Estimating Proportions, NASA Contract NAS-9-13894-1S, Texas A&M University, Department of Mathematics, Report #4, April, 1975.
2. L. F. Guseman, Jr. and Bruce P. Marion, LFSPMC (VERSION 2): Linear Feature Selection Program Using the Probability of Misclassification, NASA Contract NAS-9-14689-4S, Texas A&M University, Department of Mathematics, Report #6, March, 1976.
3. Charles L. Lawson and Richard J. Hanson, Solving Least Square Problems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.
4. Howard G. Tucker, An Introduction to Probability and Mathematical Statistics, Academic Press, New York, 1962.

USERS GUIDE
ESTPRO: ESTIMATION OF PROPORTION PROGRAM USING
FEATURE SELECTION

by

L. F. Guseman, Jr., Bruce P. Marion
and M. Swasdee

Department of Mathematics
Texas A&M University

Report #8

Prepared For

Earth Observations Division
NASA/Johnson Space Center
Houston, Texas

Contract NAS-9-14689-4S

May, 1976

TABLE OF CONTENTS

Section

1.0	INTRODUCTION	1
2.0	SUMMARY OF ESTIMATORS	2
3.0	PROGRAM PARAMETERS	9
4.0	OPERATING PROCEDURE	12
5.0	EXAMPLE INPUT	15

1.0 INTRODUCTION

The following is intended as a users guide for the FORTRAN program ESTPRO. Program ESTPRO provides the necessary computations for performing proportion estimation by the four methods discussed in [3].

The estimation algorithms available in ESTPRO were developed to treat the following problem:

Give LANDSAT data (multitemporal) over a LACIE sample segment, and training data for the classes in the segment, estimate what proportion of the sample segment is in WHEAT.

The program assumes that a finite number of component classes (e.g. crops) are present in the segment, and that each component class is described by a multivariate normal density function with known mean vector and covariance matrix (usually obtained from a training sample).

Under the above assumptions, the estimation algorithms performed by ESTPRO can be summarized as follows:

- (a) The density function for each of the classes WHEAT and NON-WHEAT is expressed as an appropriate convex combination of the component multivariate normal density functions.
- (b) One or more feature selections are performed to produce corresponding optimal one-dimensional Bayes classifiers for WHEAT VS. NON-WHEAT whose associated confusion matrices are known.
- (c) Using the one-dimensional classifiers and associated confusion matrices, estimates of the true proportion of WHEAT in the sample segment are made.

2. SUMMARY OF ESTIMATORS

The following discussion forms the basis for the estimation procedures in ESTPRO.

Let Π_1, \dots, Π_m be distinct classes with true (but unknown) a priori probabilities $\alpha_1, \dots, \alpha_m$, respectively. We assume that $\bigcup_{i=1}^m \Pi_i$ has mixture density

$$f = f_X = \sum_{i=1}^m \alpha_i f_i,$$

where each class conditional density function $f_i = f_{X/\Pi_i}$ is assumed to be $N(\mu_i, \Sigma_i)$ with μ_i, Σ_i known and Σ_i positive definite, $1 \leq i \leq m$.

Let $\Gamma_1 = \bigcup_{i=1}^k \Pi_i$, $\Gamma_2 = \bigcup_{i=k+1}^m \Pi_i$. Then Γ_1 and Γ_2 have a priori probabilities $\gamma_1 = \alpha_1 + \dots + \alpha_k$ and $\gamma_2 = \alpha_{k+1} + \dots + \alpha_m$, respectively.

Letting

$$h_1 = \sum_{i=1}^k \frac{\alpha_i}{\gamma_1} f_i$$

and

$$h_2 = \sum_{i=k+1}^m \frac{\alpha_i}{\gamma_2} f_i,$$

we have

$$f = \sum_{i=1}^m \alpha_i f_i = \gamma_1 h_1 + \gamma_2 h_2,$$

where the density functions h_1 and h_2 are convex combinations of normals. Henceforth it will be convenient to denote the above expression for f as a convex combination of convex combinations by h ; that is $h = \gamma_1 h_1 + \gamma_2 h_2$.

Throughout, x_1, \dots, x_N will denote a random sample of n -dimensional vectors from the sample segment. Feature selection is performed using the algorithm LFSPMC from [2]. The constrained least squares problem (constrained quadratic minimization) is solved using LSI from [4].

We now summarize the four algorithms available in ESTPRO for estimating $\gamma = (\gamma_1, \gamma_2)^T$.

ESTIMATOR 1

Step 1. Using the mixture density

$$f = \frac{1}{m} \sum_{i=1}^m f_i,$$

perform feature selection to obtain a $1 \times n$ vector B^* of norm one which minimizes

$$g(B) = 1 - \sum_{i=1}^m \frac{1}{m} \int_{R_i} f_i(y, B) dy,$$

where,

$$R_i(B) = \left\{ y \in E^1 : f_i(y, B) = \max_{1 \leq j \leq m} f_j(y, B) \right\}, \quad 1 \leq i \leq m,$$

and

$$f_i(y, B) \sim N(B\mu_i, B\Sigma_i B^T).$$

Step 2. Compute the $m \times m$ error matrix $P = (p_{ij})$ associated with the decision regions $R_i(B^*)$ defined by B^* , where

$$p_{ij} = \int_{R_i(B^*)} f_j(y, B^*) dy, \quad i, j = 1, 2, \dots, m.$$

Step 3. Using the decision regions $R_i(B^*)$, $1 \leq i \leq m$, classify the random sample x_1, \dots, x_N to obtain the $m \times 1$ classification vector $\hat{e} = (\hat{e}_1, \dots, \hat{e}_m)^T$, where $\hat{e}_i = \frac{N_i}{N}$, and N_i is the number of x_j in the sample such that $B^*x_j \in R_i(B^*)$, $1 \leq i \leq m$.

Step 4. Using P and \hat{e} , determine $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_m)^T$ which minimizes $\|P\hat{\alpha} - \hat{e}\|$ subject to $\sum_{i=1}^m \alpha_i = 1$, $\alpha_i \geq 0$, $1 \leq i \leq m$.

Step 5. ESTIMATOR 1 is given by

$$\begin{aligned} \hat{\gamma} &= \begin{pmatrix} \hat{\gamma}_1 \\ \hat{\gamma}_2 \end{pmatrix} = \begin{pmatrix} \hat{\alpha}_1 + \dots + \hat{\alpha}_k \\ \hat{\alpha}_{k+1} + \dots + \hat{\alpha}_m \end{pmatrix} \\ &= \begin{pmatrix} \overbrace{1 \dots 1}^k & 0 \dots 0 \\ 0 \dots 0 & \underbrace{1 \dots 1}_{m-k} \end{pmatrix} \begin{pmatrix} \hat{\alpha}_1 \\ \vdots \\ \hat{\alpha}_m \end{pmatrix} \\ &= A\hat{\alpha}. \end{aligned}$$

If P is invertible and $P^{-1}\hat{e}$ satisfies the nonnegativity constraints, then $\hat{\alpha} = P^{-1}\hat{e}$ and $\hat{\gamma} = AP^{-1}\hat{e}$ is an unbiased estimate of γ .

ESTIMATOR 2

Step 1. Perform ESTIMATOR 1 to obtain $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_m)^T$ and $\hat{\gamma} = (\hat{\gamma}_1, \hat{\gamma}_2)^T$ and let

$$\hat{h}_1 = \sum_{i=1}^k \frac{\hat{\alpha}_i}{\hat{\gamma}_1} f_i,$$

$$\hat{h}_2 = \sum_{i=k+1}^m \frac{\hat{\alpha}_i}{\hat{\gamma}_2} f_i.$$

Step 2. Using the mixture density

$$\hat{h} = \hat{\gamma}_1 \hat{h}_1 + \hat{\gamma}_2 \hat{h}_2,$$

perform feature selection to obtain a $1 \times n$ vector B^* of norm one which minimizes

$$g(B) = \hat{\gamma}_1 \int_{S_2(B)} \hat{h}_1(y, B) dy + \hat{\gamma}_2 \int_{S_1(B)} \hat{h}_2(y, B) dy,$$

where

$$S_1(B) = \{y \in E^1 : \hat{\gamma}_1 \hat{h}_1(y, B) \geq \hat{\gamma}_2 \hat{h}_2(y, B)\}$$

$$S_2(B) = \{y \in E^1 : \hat{\gamma}_1 \hat{h}_1(y, B) < \hat{\gamma}_2 \hat{h}_2(y, B)\}$$

Step 3. Compute the 2×2 error matrix $\hat{Q} = (\hat{q}_{ij})$ associated with the resulting decision regions $\hat{S}_1 = S_1(B^*)$ and $\hat{S}_2 = S_2(B^*)$ defined by B^* , where

$$\hat{q}_{ij} = \int_{\hat{S}_i} \hat{h}_j(y, B^*) dy, \quad i, j = 1, 2.$$

Step 4. Using the decision regions \hat{S}_1, \hat{S}_2 , classify the random sample x_1, \dots, x_N to obtain the 2×1 classification vector $\hat{d} = (\hat{d}_1, \hat{d}_2)^T$, where $\hat{d}_i = \frac{N_i}{N}$, and N_i is the number of x_j in the sample such that $B^* x_j \in \hat{S}_i$, $i = 1, 2$.

Step 5. ESTIMATOR 2 is given by that $\hat{\gamma} = (\hat{\gamma}_1, \hat{\gamma}_2)^T$ which minimizes $||\hat{Q}\hat{\gamma} - \hat{d}||$ subject to $\hat{\gamma}_1 + \hat{\gamma}_2 = 1$, $\hat{\gamma}_1 \geq 0$, $\hat{\gamma}_2 \geq 0$. When \hat{Q} is invertible and $\hat{Q}^{-1}\hat{d}$ satisfies the nonnegativity constraints, then $\hat{\gamma} = \hat{Q}^{-1}\hat{d}$.

ESTIMATOR 3

Step 1. Using conditional densities

$$\bar{h}_1 = \frac{1}{k} \sum_{i=1}^k f_i,$$

$$\bar{h}_2 = \frac{1}{m-k} \sum_{i=k+1}^m f_i$$

and mixture density

$$\bar{h} = \frac{1}{2} \bar{h}_1 + \frac{1}{2} \bar{h}_2$$

perform feature selection to obtain a $1 \times n$ vector B^* of norm one which minimizes

$$g(B) = \frac{1}{2} \int_{S_2(B)} h_1(y, B) dy + \frac{1}{2} \int_{S_1(B)} h_2(y, B) dy,$$

where

$$S_1(B) = \{y \in E^1 : \bar{h}_1(y, B) \geq \bar{h}_2(y, B)\}$$

$$S_2(B) = \{y \in E^1 : \bar{h}_1(y, B) < \bar{h}_2(y, B)\} .$$

Step 2. Compute the 2×2 error matrix $\bar{Q} = (\bar{q}_{ij})$ associated with the resulting decision regions $\bar{S}_1 = \bar{S}_1(B^*)$ and $\bar{S}_2 = \bar{S}_2(B^*)$ defined by B^* , where

$$\bar{q}_{ij} = \int_{\bar{S}_i} \bar{h}_j(y, B^*) dy$$

Step 3. Using the decision regions \bar{S}_1, \bar{S}_2 , classify the random sample x_1, \dots, x_N to obtain the 2×1 classification vector $\bar{d} = (\bar{d}_1, \bar{d}_2)^T$ where $\bar{d}_i = \frac{N_i}{N}$, and N_i is the number of x_j in the sample such that $B^*x_j \in \bar{S}_i$, $i = 1, 2$.

Step 4. ESTIMATOR 3 is given by that $\bar{\gamma} = (\bar{\gamma}_1, \bar{\gamma}_2)^T$ which minimizes $||\bar{Q} \bar{\gamma} - \bar{d}||$ subject to $\bar{\gamma}_1 + \bar{\gamma}_2 = 1$, $\bar{\gamma}_1 \geq 0$, $\bar{\gamma}_2 \geq 0$. When \bar{Q} is invertible and $\bar{Q}^{-1}\bar{d}$ satisfies the nonnegativity constraints, then $\bar{\gamma} = \bar{Q}^{-1}\bar{d}$.

ESTIMATOR 4

Step 1. Perform ESTIMATOR 1 to obtain $\hat{\alpha} = (\hat{\alpha}_1, \dots, \hat{\alpha}_m)^T$,

$\hat{\gamma} = (\hat{\gamma}_1, \hat{\gamma}_2)^T$ and subsequent conditional densities

Step 2. Compute B^* and decision regions \bar{S}_1 and \bar{S}_2 from ESTIMATOR 3, and obtain \bar{Q}, \bar{d} .

$$\hat{h}_1 = \sum_{i=1}^k \frac{\hat{\alpha}_i}{\hat{\gamma}_2} f_i, \text{ and}$$

$$\hat{h}_2 = \sum_{i=k+1}^m \frac{\hat{\alpha}_i}{\hat{\gamma}_2} f_i.$$

Step 3. Compute the 2x2 error matrix $Q^* = (q_{ij}^*)$, where

$$q_{ij}^* = \int_{\bar{S}_i} \hat{h}_j(y, B^*) dy, \quad i, j = 1, 2.$$

Step 4. When \bar{Q} is invertible, ESTIMATOR 4 is given by

$$\gamma^* = \hat{\gamma} \pm (\bar{\gamma} - \bar{Q}^{-1} Q^* \hat{\gamma}).$$

If $\bar{\gamma} = \bar{Q}^{-1} \bar{d}$ and $\hat{\gamma} = AP^{-1} \hat{e}$ then

$$\gamma^* = \hat{\gamma} \pm \bar{Q}^{-1} (\bar{d} - Q^* AP^{-1} \hat{e}),$$

and, for either choice of sign, γ^* is an unbiased estimator of γ .

3.0 PROGRAM PARAMETERS

Apart from the various program parameters and command cards (discussed in the sequel), the basic input data to ESTPRO consists of the class names, mean vectors and covariance matrices which comprise the component class statistics deck. All input data to the program is from unit reference 5 (usually punched cards). All output from the program is printed on unit reference 6. Several additional options are built into the program which provide the user with the capability of making successive runs using designated subsets of the original features provided by the component class statistics deck.

3.1 Parameter Initialization

All input variables to the program are of a fixed format and must be entered as shown in Section 4.0 and as illustrated in the examples in Section 5.0. These variables are:

- MTOT : Number of classes in the component class statistics deck.
- NFPC : Number of features per class in the component class statistics deck.
- N : Dimension of feature space, \leq NFPC.
- CLS : Class names, 12 characters, double subscripted array.
- KCLS : Numeric labels of the designated classes from the MTOT classes in the component class statistics deck, single subscripted array.
- IFEA : Numeric labels of the N designated features from the NFPC features in the component class statistics deck, single subscripted array.
- COVARB : Input covariance matrices, triple subscripted array.

- XMEANB : Input mean vectors, double subscripted array.
- ICMB : Numeric labels of component classes as defined in vector KCLS used in defining convex combinations (Need not be in ascending order), double subscripted array.

Command codes select program options as follows:

- STAT : definition and entry of a given component class statistics deck.
- FEAT : definition of a designated subset of features from the current component class statistics deck.
- EST1 }
EST2 }
EST3 } : choice of estimator to be used with current
EST4 } statistics deck and designated subset of features.

When the STAT command is used, values of MTOT and NFPC for the new statistics deck are entered. The names for the respective component classes in the statistics deck are defined on succeeding cards. The component class statistics deck, comprised of the MTOT mean vectors in the order of ascending class numbers followed by the MTOT covariance matrices in the order of ascending class numbers, is entered. The entries of each mean vector in the order of ascending feature number are entered according to the format (5X, 5D15.8). The NFPC(NFPC + 1)/2 elements on and above the diagonal of each covariance matrix are entered by column in the format (5X, 5D15.8). It is assumed that the diagonal elements of each covariance matrix are in order of ascending feature number. The first entry of each new mean vector or covariance matrix starts on a new card. The entire statistics deck with appropriate class names is printed.

If the FEAT command is selected, a new value for N and the numeric labels of the desired features (IFEA) are entered and printed.

Choice of estimators is made by using one of the command cards EST1, EST2, EST3 or EST4.

Once a particular estimator is chosen, successive cards are used to define convex combinations of the component classes. Each card consists of the numeric labels of the desired component classes defining that convex combination. If each component class is a separate convex combination (e.g., as in EST1), then a single card with the numeric labels of all component classes is used. After the convex combinations are defined, the names of the component classes defining the convex combinations and their respective a priori probabilities are output. Parameters initialized using the STAT and FEAT commands remain in effect until the respective command is again used.

The N -dimensional sample pixels to be used in generating classification vectors are entered using a variable FORMAT (read in at object time). If available, the number of sample pixels in each component class is entered. Otherwise, any positive numbers can be entered for each component class with the restriction that their sum equals the total number of sample pixels.

Use of EST2 or EST4 requires intermediate determination of EST1 and consequently definition of two sets of convex combinations and two classifications (see section 5.0).

4.0 OPERATING PROCEDURE

In order to simulate object time dimensioning, the user must provide a calling routine of the following form:

```
DIMENSION ALRGE(IDIM)
DOUBLE PRECISION BLRGE(IDIM2)
COMMON MX,NX
MX =
NX =
CALL PRDIM(ALRGE,BLRGE)
STOP
END
```

The values of MX, NX, IDIM and IDIM2 are determined as follows:

MX = maximum value of MTOT for the program run.

NX = maximum value of NFPC for the program run.

$$IDIM = MX(23+4MX+NX(\frac{NX+1}{2})) + NX(\frac{5}{2}NX + \frac{5}{2}) + 12$$

$$IDIM2 = MX(7+3MX + \frac{MX^2}{3} + 3NX + 2NX^2) + NX(\frac{NX}{2} + \frac{11}{2}) + 7$$

If available storage is not a problem, the user can incorporate maximum fixed dimensions into the program.

Input parameters are of a fixed format and must be in a specified order. Shown below are the variable names as described in Section 3.0 and the card formats for the command code sequences.

Statistics Definition:

"STAT",MTOT,NFPC
class names (one per card) [FORMAT(3A4)]
mean vectors [FORMAT(5X,5D15.8)]
covariance matrices [FORMAT(5X,5D15.8)]

Feature Definition:

"FEAT",N [FORMAT(A4,I2)]
IFEA [FORMAT(24(I2,1X))]

Several cards may be used to define IFEA if $N > 24$.

Selection of Estimator

"EST1",MTOT,N,00000001
all class labels [FORMAT(26(I2,1X),A2)]
number of sample pixels/class [FORMAT(16(I4,1X))]
variable FORMAT for sample pixels [FORMAT(20A4)]
sample pixels according to variable FORMAT

"EST2",MTOT,N,00000001
all class labels [FORMAT(26(I2,1X),A2)]
number of sample pixels/class [FORMAT(16(I4,1X))]
variable FORMAT for sample pixels [FORMAT(20A4)]
sample pixels according to variable FORMAT
class labels, convex combination 1 [FORMAT(26(I2,1X),A2)]
class labels, convex combination 2 [FORMAT(26(I2,1X),A1)]

number of sample pixels/class	[FORMAT(16(I4,1X))]
"EST3",MTOT,N,00000001	
class labels, convex combination 1	[FORMAT(26(I2,1X),A2)]
class labels, convex combination 2	[FORMAT(26(I2,1X),A2)]
number of sample pixels/class	[FORMAT(16(I4,1X))]
variable FORMAT for sample pixels	[FORMAT(20A4)]
sample pixels	according to variable FORMAT
"EST4",MTOT,N,00000001	
all class labels	[FORMAT(26(I2,1X),A2)]
number of sample pixels/class	[FORMAT(16(I4,1X))]
variable FORMAT for sample pixels	[FORMAT(20A4)]
sample pixels	according to variable FORMAT
class labels, convex combination 1	[FORMAT(26(I2,1X),A2)]
class labels, convex combination 2	[FORMAT(26(I2,1X),A2)]
number of sample pixels/class	[FORMAT(16(I4,1X))]

If the same statistics deck and sample pixels are being used with different estimators, then the variable FORMAT and sample pixels need not be re-entered. If more than twenty-six component classes constitute a single convex combination, a slash (/) in column 79 of the current card indicates a continued definition of the convex combination on succeeding cards.

5.0 EXAMPLE INPUT

Example 1.

A 5 class, 16 dimensional component class statistics deck of MSS measurements from Hill County, Montana, is entered. ESTIMATOR 1 is then called to estimate the proportions of all five classes using a random sample of 2417 pixels. Then ESTIMATOR 2, ESTIMATOR 3 and ESTIMATOR 4 are selected to estimate the proportion of class 1 (WHEAT) versus classes 2-5 (FALLOW, BARLEY, GRASS, STUBBLE) using the same statistics deck and sample pixels.

Col. 1

STAT0516

WHEAT

FALLOW

BARLEY

GRASS

STUBBLE

statistics deck

EST1051600000001

01,02,03,04,05

0784,0744,0300,0206,0383

(8X,16I3,24X)

sample pixels

EST2051600000001

01,02,03,04,05

0784,0744,0300,0206,0383

01
02,03,04,05
0784,0744,0300,0206,0383
EST3051600000001
01
02,03,04,05
0784,0744,0300,0206,0383
EST4051600000001
01,02,03,04,05
0784,0744,0300,0206,0383
01
02,03,04,05
0784,0744,0300,0206,0383

Example 2.

The component class statistics deck and sample pixels from Example 1 are entered. For this run, ESTIMATOR 1 is called to estimate the proportions of all five classes. Then ESTIMATORS 2-4 are selected to estimate the proportion of class 3 (BARLEY) versus classes 1, 2, 4, 5 (WHEAT, FALLOW, GRASS, STUBBLE).

Col. 1

STAT0516
WHEAT
FALLOW
BARLEY
GRASS
STUBBLE

statistics deck

EST1041500000001

01,02,03,04,05

0784,0744,0300,0206,0383

(8X,16I3,24X)

sample pixels

EST2051600000001

01,02,03,04,05

0784,0744,0300,0206,0383

03

01,02,04,05

0784,0744,0300,0206,0383

EST3051600000001

03

01,02,04,05

0784,0744,0300,0206,0383

EST4051600000001

01,02,03,04,05

0784,0744,0300,0206,0383

03

01,02,04,05

0784,0744,0300,0206,0383

REFERENCES

1. T. W. Anderson, An Introduction to Multivariate Statistical Analysis, John Wiley and Sons, New York, 1953.
2. L. F. Guseman, Jr. and Bruce P. Marion, LFSPMC (VERSION 2): Linear feature selection using the probability of misclassification, NASA Contract NAS-9-14689-4S, Texas A&M University, Department of Mathematics, Report #6, March, 1976.
3. L. F. Guseman, Jr. and Jay R. Walton, Methods for estimating proportions of convex combinations of normals, NASA Contract NAS-9-14689-4S, Texas A&M University, Department of Mathemtaics, Report #7, March, 1976.
4. Charles L. Lawson and Richard J. Hanson, Solving Least Squares Problems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

COMPUTER PROGRAM DOCUMENTATION
PROGRAM NONLN1 AND NONLN2
NONLINEAR COLOR DISPLAY PROGRAM

by

Jack D. Bryant, David Nobles and Manot Swasdee
Department of Mathematics
Texas A&M University

Report #9

Prepared For

Earth Observations Division
NASA/Johnson Space Center
Houston, Texas

Contract NAS-9-14589-4S

May, 1976

TABLE OF CONTENTS

Section	Page
Abstract	1
Application.	1
Source Language.	2
Restrictions	2
I/O Configuration.	2
Deck Set-up.	2
Usage.	3
Timing	4
Storage Requirements	4
Possible Extension, Suggestions for Improvements	4
Additional Information	6
Mathematical Description: NONLN1 and NONLN2	7
Description of Program Variables.	10
Flow Charts	13
Listing	16

LIST OF TABLES

Table	Page
1. Local Variables : NONLN1.	10
2. Variables in COMMON block : NONLN3.	11
3. Local Variables : NONLN2.	12

COMPUTER PROGRAM DOCUMENTATION

Program NONLN1 and NONLN2

Nonlinear Color Display Program

Abstract. These programs together perform a transformation of a four dimensional real row DATA to produce a four dimensional row TDATA. NONLN1, which is called first (and only once), sets parameters for NONLN2. NONLN1 uses statistics (means and diagonal of covariance matrix) on NCLASS classes and the class numbers IW and IG of distinguished classes (called "wheat" and "grass" here). NONLN2 takes input DATA into numbers 0. to 64. so that input data near wheat will map to TDATA high in channel 4 and low in channel 1 and 2, and input data near grass will become high in channel 1 and low in 2 and 4. With standard color assignments used to display ERTS data, wheat becomes red, grass green.

This method of transforming data is highly nonlinear and is quite sensitive to noise in the input data; however, the output color spread is spectacular.

Application. The program was developed for a specific application: when multispectral multitemporal data is transformed to lower dimensionality using a feature selection program, the transformed data has no intrinsic meaning--no "reality." However, the transformed data (if, say, four dimensional) should be more managable provided it can be displayed at all. The problem is not just the range of the transformed data, or the

fact that real data, unlike classes the feature selection program was "trained" on, transform wildly; the real problem is the lack of any consistent relationship in the transformed data. Thus it is easy to produce a color display but hard to analyze what the colors mean. Program NONLN1-2 attempts to align the selection of colors so that certain classes have certain colors, and this assignment will be relatively independent of which feature selection program produced the reduction in dimensionality.

The program is applicable to raw single pass data. Although wheat and grass are probably not separated well enough, other classes are. The program should greatly enhance color display of single pass data.

Source Language. FORTRAN IV 100%

Restrictions. Name NONLN3 is reserved (the name of a COMMON block). If classes IW and IG are too close the transformed data will be noisy and lose separation. NCLASS must be at least 3.

I/O Configuration. Both programs are I/O free.

Deck Set-up.

Job Control Cards
Calling program*
NONLN1
NONLN2
FOLD64
Job Control Cards
Data

*for comments on the calling program, see Usage.

Usage. Calling sequence, NONLN1:

CALL NONLN1 (NCLASS,BMU,BSBTD,IW,IG), where

NCLASS number of classes for which training statistics
 are given

BMU mean vector for each class: NCLASS by 4

BSBTD diagonal elements of covariance matrix: NCLASS by 4

IW class to be made red

IG class to be made green

CALL NONLN2 (DATA,TDATA,LDA), where

DATA input row, type real, LDA by 4

TDATA output row, type real, LDA by 4

LDA number of pixels in a row

Comments on the calling program. The calling program may have a
general flow as follows:

1. Set up data set with data to be transformed and transformed data (may be the same data set since input is only needed once).
2. Input or otherwise determine parameters for NONLN1.
3. Call NONLN1.
4. Itialize a row counter loop.
5. Input a row of data.
6. Convert to type REAL if necessary and store in DATA
7. Call NONLN2.
8. Store a row of data, packing or converting from type REAL if necessary.
9. Next row.

Timing. NONLN2 is fast, probably faster than whatever I/O and data conversions are going on. For each pixel, about five floating point multiplications and ten additions, and ten each integer additions and multiplications are required.

Storage Requirements. Approximately 5000 8 bit bytes code for both NONLN1 and NONLN2; array storage (assuming 32 bit real numbers) will be about $4 * (47 + 8 * (NCLASS + LDA))$ 8 bit bytes.

Possible Extension, Suggestions for Improvements.

These comments refer to batch operation only.

1. Suggestion for improving speed: This suggestion is based on two observations: first, owing to noise, the transformed data need have no more than (say) 64 levels. Second, numbers in channel I between S3MIN(I) and S3MAX(I) are mapped into numbers 0. to 64. Thus a vector for each channel can be set up to map DATA into TDATA by simply computing an index and referencing a vector. The following sketch of a program shows how this might be done:

```
SUBROUTINE NONLN4(CH1,CH2,CH3,CH4)
DIMENSION DATA(100,4),TDATA(100,4),CH1(100),CH2(100),CH3(100),CH4(100)
COMMON/NONLN3/SIMIN(4),S2MAX(4),S3MIN(4),S3MAX(4)
get data NCLASS,BMU,BSBTD,IW,IG
CALL NONLN1(NCLASS,BMU,BSBTD,IW,IG)
DO 5 J = 1,4
X = S3MIN(J)
DX = (S3MAX(J) - S3MIN(J))/100
DO 5 I = 1, 100
DATA(I,J) = X
5 X = X + DX
CALL NONLN2(DATA,TDATA,100)
DO 10 I = 1, 100
CH1(I) = TDATA(I,1)
CH2(I) = TDATA(I,2)
CH3(I) = TDATA(I,3)
CH4(I) = TDATA(I,4)
RETURN
END
```

The output of this program could be applied as follows:

For each pixel: determine if all four values are in range S3MIN to S3MAX; if not, set all transformed values 0. If so, set transformed value in channel 1 equal to CH1(KX) where

$$KX = 1 + (S3MAX(1) - X) * 100 / (S3MAX(1) - S3MIN(1))$$

and so on for channel 2, 3 and 4.

2. In a number of important applications the four dimensional data will be type INTEGER, perhaps even six or seven bit integers (0 to 63 or 0 to 127). For instance, the data may have been transformed and then scaled and packed; or the data may be one pass data. In such a case it is clear that no scaling at all is necessary to produce a transformation of the data--only larger (perhaps) vectors CH1 through CH4. Further, here the transformation would most likely be back to integer variables anyway, so that CH1 through CH4 can be integer vectors.
3. The last possibility (that the transformation is from integer to integer) can be improved even more if the computer has a capability similar to the translate under mask instruction of the IBM 360/370. Four masks can be set up and a whole row of packed data can be translated at once (i.e., in just a few microseconds). This would obviously be the way to go if it were possible.
4. Suggestion for testing "tamer" versions: Most of the wildness of NONLN1-2 is caused by the size of fudge factor GWGRN and WGRED (see the mathematical documentation and Table 2 for definition). For example, with the test data given WGRED

is nearly 300 and GWGRN over twice this. (Note, however, this only corresponds to a noise amplification factor of a little under 5 in the red channel and about 11 for green. The blue channel is hardly affected at all by amplification of noise since FFBLU is relatively small.) In any case, one may be willing to accept less spectacular colors in exchange for less noise. One suggestion would be to decrease the number 64. in the definition of GWGRN and WGRED to, say 32. It would probably still turn out that wheat was the reddest thing around and grass the greenest.

5. A restriction of NONLN1 is that NCLASS \geq 3. If NCLASS = 2, use instead program ROTAT1-2.

Additional Information.

Mathematical description: NONLN1 and NONLN2

Table 1. Local variables : NONLN1

Table 2. Variables in COMMON block : NONLN3

Table 3. Local variables : NONLN2

General flow chart : NONLN1

General flow chart : NONLN2

General flow chart : Test program

Listing

Test program and test listing

Mathematical Description: NONLN1 and NONLN2

The idea behind this transformation is to force one user-selected class to be red and another user-selected class to be green. In the program itself, these classes are referred to as "wheat" and "grass" respectively. A third class is selected by the program to be saturated in blue. Of course, if the user-selected classes are close together this procedure results in a noisy image; worse, unless considerable care is taken, much of the original separation will be lost in the transformed data. Most of the complication of the program is concerned with preserving as much separation as possible.

Although NONLN1 is called first to set the program parameters in NONLN2 (which performs the actual transformation), we describe NONLN2 first. Program NONLN1 is easier to understand once the use to which the parameters are put is known.

On each call to NONLN2, one row DATA of LDA pixels is transformed and returned in TDATA. There is no assumption made anywhere that DATA consists of (for example) positive numbers, or numbers in any certain range. The first task NONLN2 accomplishes is to examine a pixel and decide if it is almost certainly not like any of classes which NONLN1 used to define the transformation. This is done by seeing if one of the pixel's four coordinates lies outside the largest and smallest expected significant value as defined in NONLN1. In the program this value is S3MAX(I) and S3MIN(I) respectively, I = 1 to 4, and represents maximum of mean + 3 sigma over NCLASS classes and minimum of mean - 3 sigma over NCLASS classes per coordinate. Failure to be acceptable in any coordinate results in a zero value in each output coordinate and movement to the next pixel.

Acceptable data is now mapped into numbers (generally) between 0 and 1 linearly in each channel by the map $y = (x-m)/(M-m)$, where x is the input coordinate, M is the maximum mean + sigma over NCLASS classes, m is the minimum mean - sigma over NCLASS classes, and y is the output value. In the program, x and y are both stored in DATA, m is S1MIN(I), $I = 1$ to 4, and $1/(M-m)$ is S1MAX(I), $I = 1$ to 4.

The stage is now set for the actual transformation. Let $w(i)$ and $g(i)$ denote the transformed means of class IW and IG, $i = 1$ to 4. (These parameters are set by NONLN1 of course.) Let f denote the function with $f(64 \cdot I) = 0$, I even, $f(64 \cdot I) = 64$, I odd and f linear between. Let i_g, i_r, i_b and i_o denote the green, red, blue and other channel numbers selected by NONLN1. The green, red and other channels are straightforward: with input x ,

$$\text{green: } y(1) = f \left(64 \cdot \frac{x(i_g) - w(i_g)}{g(i_g) - w(i_g)} \right)$$

$$\text{red: } y(4) = f \left(64 \cdot \frac{x(i_r) - g(i_r)}{w(i_r) - g(i_r)} \right)$$

$$\text{other: } y(3) = f(64 \cdot x(i_o))$$

For the blue channel, things get a little tricky: let

$$t = \max \left(0, \left| x(i_b) - \frac{g(i_b) + w(i_b)}{2} \right| - \left| \frac{w(i_b) - g(i_b)}{2} \right| \right)$$

Let t_M denote the maximum of t for x ranging over the means of all classes. Then

blue: $y(2) = f(64 \cdot t/t_M)$.

Now we describe the selection of channels i_r , i_g and i_b . As can be seen immediately, NONLN2 is a violent mapping if classes IW and IG are not well separated. Thus we select for red and green the index i_r with $w(i) - g(i)$ maximum and i_g the index with maximum remaining $w(i) - g(i)$, $i \neq i_r$. Blue index i_b , on the other hand, is selected to be that index with minimum $w(i) - g(i)$. Once the indices are set, simple variables are set equal to the various fudge factors; these variables are passed from NONLN1 to NONLN2 through named COMMON block NONLN3.

TABLE 1. Local variables : NONLNI

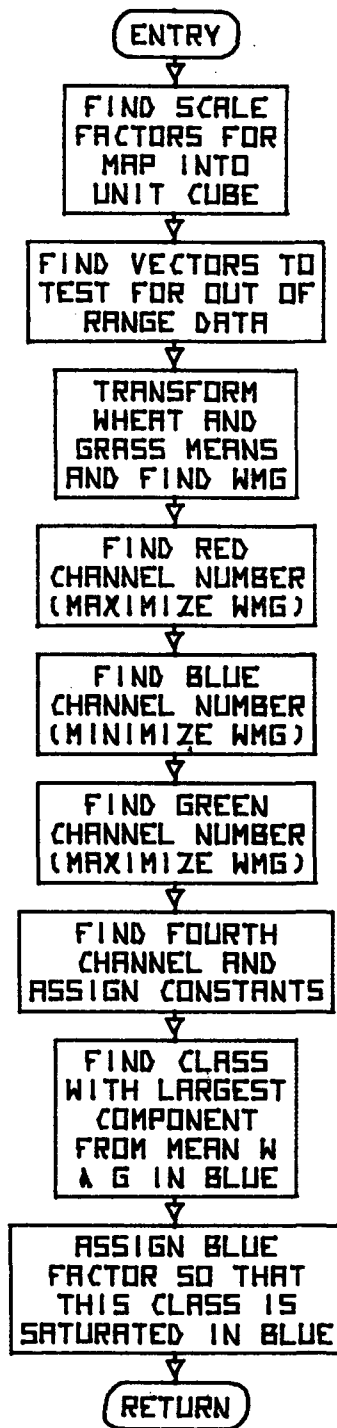
I	DO loop index
J	DO loop index
NCLASS	Number of classes for which training statistics are furnished
SIG	Temporary value of standard deviation
BSBTD(J,I)	Diagonal of covariance matrix in channel I, class J
BMU(J,I)	Mean, channel I, class J
TEMP	Temporary real variable used for getting maximum or minimum
W(4)	Transformed (into unit cube) class IW mean vector $w(i)$ in mathematical description
G(4)	Transformed class IG mean vector $g(i)$ in mathematical description
WMG(I)	$ w(i) - g(i) $
XTEMP	Another temporary real variable
IW	Class number of wheat
IG	Class number of grass

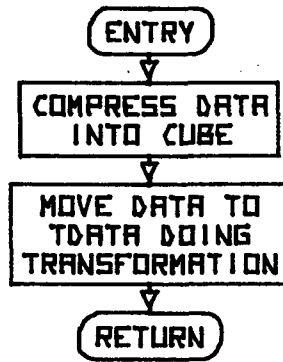
TABLE 2. Variables in COMMON block : NONLN3

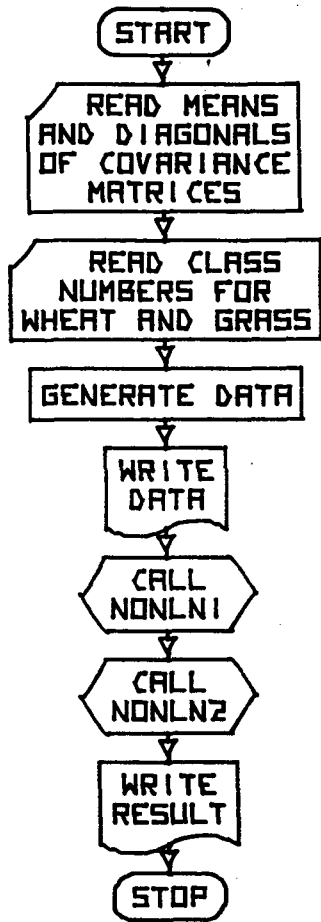
S1MIN(4)	Minimum over classes of mean in channel I minus sigma in channel I, $I = 1,4$
S1MAX(4)	$1./(S1MAX(I) - S1MIN(I))$, where S1MAX(I) is first set to be the maximum over classes of mean plus sigma in channel I.
S3MIN(4) S3MAX(4)	Minimum (respectively maximum) of mean - 3 sigma (respectively mean + 3 sigma) over classes for channel I.
IRED	Index of red channel i_r
IBLUE	Index of blue channel i_b
IGRN	Index of green channel i_g
WGRN	$w(i_g)$
GWGRN	$64./(w(i_r) - g(i_r))$
GWBLU	$(g(i_b) + w(i_b))/2$
GRED	$g(i_r)$
WGRED	$64./(w(i_r) - g(i_r))$
WMGB2	$ w(i_b) - g(i_b) /2$
FFBLU	$64/t_M$ (see the mathematical description for definition of t_M)
I4TH	$10-(i_r + i_g + i_n)$ (Index of other channel)

TABLE 3. Local variables : NONLN2

DATA(LDA,4)	One row of four dimensional real data to be transformed
TDATA(LDA,4)	The transformed data
LDA	Number of pixels per row
IPT	DO loop index--pixel numbers
ICHNL	DO loop index--channel number
TEMP	Temporary real variable







```
C
C      SUBROUTINE NCNLN1 (NCLASS, BMU, ESBTD, IW, IG)
C
C      FUNCTION      - TO SET PARAMETERS FOR SUBROUTINE NONLN2
C      USAGE        - CALL NCNLN1(NCLASS, BMU, ESBTD, IW, IG)
C      PARAMETERS   -
C      NCLASS       - NUMBER OF CLASSES FOR WHICH TRAINING STATISTICS EXIST
C      BMU          - MEAN VECTOR FOR EACH CLASS: NCLASS BY 4
C      ESBTD        - DIAGONAL ELEMENTS OF COVARIANCE MATRIX: NCLASS BY 4
C      IW           - CLASS NUMBER OF CLASS TO BE MADE RED
C      IG           - CLASS NUMBER OF CLASS TO BE MADE GREEN
C      PRECISION    - SINGLE
C      REGD. RTNS.  - NONE (SINGLE PURPOSE -- CALLED BEFORE NONLN2)
C      RESTRICTIONS
C      NONLN3       - A RESERVED NAME OF A COMMON BLOCK
C      NCLASS       - MUST BE AT LEAST 3
C      LANGUAGE     - FORTRAN IV
```

```
C
C      SUBROUTINE NCNLN1 (NCLASS, BMU, ESBTD, IW, IG)
C      DIMENSION BMU(NCLASS,4),BSBTD(NCLASS,4),W(4),G(4),WNG(4)
C      COMMON NONLN3/SIMIN(4),S1MAX(4),S3MIN(4),S3MAX(4),IRED,IBLUE,
C      *          IGRN,WGRN,GWGFN,GWBLU,GRED,WGRED,WGMB2,FFBLU,IATH
```

```
C
C      FIND FUDGE FACTORS FOR CUBE
C
C      DO 40 I=1,4
C      SIMIN(I)=BMU(I,I)-SQRT(BSBTD(I,I))
C      S1MAX(I)=SIMIN(I)
C      S3MIN(I)=BMU(I,I)-3*SQRT(BSBTD(I,I))
C      S3MAX(I)=S3MIN(I)
C      DO 40 J=1,NCLASS
C      SIG=SQRT(BSBTD(J,I))
C      TEMP=BMU(J,I)-SIG
C      IF(TEMP.GT.SIMIN(I))GO TO 50
C      SIMIN(I)=TEMP
50  TEMP=BMU(J,I)+SIG
C      IF(TEMP.LT.S1MAX(I))GO TO 41
C      S1MAX(I)=TEMP
41  TEMP=BMU(J,I)-3*SIG
C      IF(TEMP.GT.S3MIN(I))GO TO 42
C      S3MIN(I)=TEMP
42  TEMP=BMU(J,I)+3*SIG
C      IF(TEMP.LT.S3MAX(I))GO TO 40
C      S3MAX(I)=TEMP
40  CONTINUE
```

```
C
C   FOR EFFICIENCY REPLACE FUDGE DIVISION BY MULTIPLICATION
C
C   DO 45 I=1,4
45  SIMAX(I)=1./(SIMAX(I)-SIMIN(I))
C
C   NORMALIZE WHEAT AND GRASS MEANS AND FIND W(I)-G(I)
C
C   DO 60 I=1,4
W(I)=(RMU(IW,I)-SIMIN(I))*SIMAX(I)
G(I)=(RMU(IG,I)-SIMIN(I))*SIMAX(I)
60  WMG(I)=ABS(W(I)-G(I))
C
C   FIND RED CHANNEL
C
C   TEMP=WMG(1)
DO 70 I=1,4
  IF(TEMP.GT.WMG(I))GO TO 70
  TEMP=WMG(I)
  IRED=I
70  CONTINUE
C
C   FIND BLUE CHANNEL
C
C   DO 80 I=1,4
  IF(I.EQ.IRED)GO TO 80
  IF(WMG(I).GT.TEMP)GO TO 80
  TEMP=WMG(I)
  IBLUE=I
80  CONTINUE
C
C   FIND GREEN CHANNEL
C
C   DO 90 I=1,4
  IF(I.EQ.IRED)GO TO 90
  IF(I.EQ.IBLUE)GO TO 90
  IF(TEMP.GT.WMG(I))GO TO 90
  TEMP=WMG(I)
  IGRN=I
90  CONTINUE
C
C   ASSIGN FOURTH CHANNEL
C
C   I4TH=10-[IRED-IBLUE-IGRN]
C
C   ASSIGN A FEW CONSTANTS
C
C   WGRN=W(IGRN)
GWBLU=(G(IBLUE)+W(IBLUE))/2.
GRED=G(IRED)
GWGRN=64./(G(IGRN)-W(IGRN))
WGRED=64./(W(IRED)-G(IRED))
```

```
C
C FIND REMAINING CLASS WITH LARGEST COMPONENT AWAY
C FROM THE MEAN OF WHEAT AND GRASS
C
TEMP=GWBLU
DO 100 I=1,NCLASS
IF(I.EQ.IW.OR.I.EQ.IG)GO TO 100
XTEMP=ABS((BMU(I,IBLUE)-S1MIN(IBLUE))*S1MAX(IEIJE)-GWELU)
IF(XTEMP.LT.TEMP)GO TO 100
TEMP=XTEMP
100 CONTINUE
C
C NOW FIND THE BLUE RIDGE FACTOR
C
WMGB2=WMG(IBLUE)/2.
FFBLJ=54./(AES(TEMP-WMGB2))
RETURN
END
```

```
C
C  SUBROUTINE NCNLN2 (DATA,TDATA,LDA)
C  FUNCTION      = TO TRANSFORM FOUR DIMENSIONAL REAL DATA A ROW AT A
C                TIME TO ANOTHER REAL ROW OF FOUR DIMENSIONAL DATA
C                IN THE RANGE 0.-64. ATTEMPTING TO MAKE CLASS IW
C                RED (HIGH IN CHANNEL 4, LOW IN 1 AND 2) AND CLASS
C                IG GREEN (HIGH IN CHANNEL 1, LOW IN 2 AND 4).
C  USAGE        = CALL NCNLN2(DATA,TDATA,LDA)
C  PARAMETERS   =
C    DATA      = ONE ROW OF FOUR DIMENSIONAL REAL DATA: LDA BY 4
C    TDATA      = THE TRANSFORMED DATA: LDA BY 4
C    LDA        = THE NUMBER OF PIXELS IN A ROW
C  PRECISION    = SINGLE
C  REQD. RTNS. = NONLN1 MUST BE CALLED TO SET PROGRAM VARIABLES
C                FUNCTION FOLD64 IS REFERENCED
C  RESTRICTIONS
C    NONLN3     = A RESERVED NAME OF A COMMON BLOCK
C  LANGUAGE     = FORTRAN IV
C
C  SUBROUTINE NONLN2(DATA,TDATA,LDA)
C  DIMENSION DATA(LDA,4),TDATA(LDA,4)
C  COMMON/NONLN3/S1MIN(4),S1MAX(4),S3MIN(4),S3MAX(4),IREC,IRLUE,
C  *           IGRN,WGRN,GWGRN,GWBLU,GRED,WGRED,WMB2,FFBLU,I4TH
C
C  COMPRESS DATA INTO CUBE
C
C  DO 130 IPNT=1,LDA
C  DO 132 ICHNL=1,4
C  TEMP=DATA(IPNT,ICHNL)
C  IF(TEMP.GT.S3MAX(ICHNL).OR.TEMP.LT.S3MIN(ICHNL))GO TO 145
132 CONTINUE
C  DO 135 ICHNL=1,4
136 DATA(IPNT,ICHNL)=(DATA(IPNT,ICHNL)-S1MIN(ICHNL))*S1MAX(ICHNL)
138 CONTINUE
C
C  MOVE DATA TO TRANSFORM ARRAY, DOING TRANSFORMATION
C
C  TDATA(IPNT,1)=FOLD64((DATA(IPNT,IGRN)-WGRN)*GWGRN)
C  TDATA(IPNT,3)=FOLD64(DATA(IPNT,I4TH)*64.)
C  TDATA(IPNT,4)=FOLD64((DATA(IPNT,IREC)-GRED)*WGRED)
C  TEMP=ABS(DATA(IPNT,IRLUE)-GWBLU)-WMB2
C  IF(TEMP.LT.0.)TEMP=0.
C  TDATA(IPNT,2)=FOLD64(TEMP*FFBLU)
C  GO TO 130
145 DO 145 I=1,4
146 TDATA(IPNT,I)=0.
130 CONTINUE
C  RETURN
C  END
```

C
C FUNCTION - TO MAP A REAL NUMBER INTO ONE BETWEEN 0. AND 64.
C BY "FOLDING" RATHER THAN CLIPPING
C
C USAGE - $Y = \text{FOLD}(4(X))$
C PARAMETERS -
C X - ANY REAL NUMBER
C Y - THE RESULTANT BETWEEN 0. AND 64. Y IS CONGRUENT TO
C $X \pmod{64}$ IF X IS IN AN INTERVAL $(64 \cdot I, 64 \cdot (I+1))$
C WITH I EVEN; OTHERWISE Y IS CONGRUENT TO $64 - X \pmod{64}$
C RESTRICTIONS NONE
C X SHOULD NOT BE TOO LARGE OR EFFICIENCY SUFFERS
C PRECISION - SINGLE
C LANGUAGE - FORTRAN IV
C

```
FUNCTION FOLD64(X)
  FOLD64=ABS(X)
  IF(FOLD64.LE.64.)RETURN
1 FOLD64=ABS(FOLD64-128.)
  IF(FOLD64.GT.64.)GOTO 1
  RETURN
END
```

COMPUTER PROGRAM DOCUMENTATION
PROGRAM ROTAT1 AND ROTAT2
ROTATION TO PRODUCE COLOR DISPLAYS

by

Jack D. Bryant, David Nobles and Manot Swasdee

Department of Mathematics
Texas A&M University

Report #10

Prepared For

Earth Observations Division
NASA/Johnson Space Center
Houston, Texas

Contract NAS-9-14689-4S

May, 1976

TABLE OF CONTENTS

Section	Page
Abstract	1
Application.	1
Source Language.	2
Restrictions	2
I/O Configuration.	2
Deck Set-up.	2
Usage.	3
Timing	4
Storage Requirements	4
Possible Extensions, Suggestions for Improvements.	4
Additional Information	
Mathematical Description	5
Description of Program Variables	20
Flow Charts	24
Listing	28

LIST OF FIGURES

Figure	Page
1. Coordinates i_r , i_g and i_b of scaled and complemented row mean vectors	11
2. Absolute values of transformation of Fig. 1 by A	12
3. Absolute value of transformation of Fig. 2 by C	13
4. Change 2 applied to C to make wheat redder	14
5. Change 1 applied to C to increase spread in blue channel .	15
6. Change 1 and 2	16
7. Random data transformed, original method	17
8. Random data, change 1 and 2.	18
9. (Upper) Change 1 and 2, six classes. (Lower) Original, six classes	19

LIST OF TABLES

Table	Page
1. Local Variables : NONLN1	20
2. Variables in COMMON block : ROTAT3	22
3. Local Variables : ROTAT2	23

COMPUTER PROGRAM DOCUMENTATION

Program ROTAT1 and ROTAT2

Rotation to Produce Color Displays

Abstract. These programs together perform a transformation of four dimensional real row DATA to produce a four dimensional row TDATA. ROTAT1, which is called first (and only once), sets parameters for ROTAT2. ROTAT1 uses statistics (means and diagonals of covariance matrices) on NCLASS classes and the class numbers IW and IG of distinguished classes (called "wheat" and "grass" here). ROTAT2 takes input DATA into numbers 0. to 64. so that input data near wheat will map to TDATA high in channel 1 and low in channel 2 and 4, and input data near grass maps to data with more channel 2 and low in channel 4. With standard color assignments used to display ERTS data, wheat becomes a saturated red, grass orange or perhaps yellow.

This method of transforming data is nearly linear and is insensitive to noise in the input data; however, the output color spread is not spectacular. Only two classes need be trained on.

Application. The program was developed for a specific application: when multispectral multitemporal data is transformed to lower dimensionality using a feature selection program, the transformed data has no intrinsic meaning--no "reality." However, the transformed data (if, say, four dimensional) should be more managable provided it can be displayed at all.

The problem is not just the range of the transformed data, or the fact that real data, unlike classes the feature selection program was "trained" on, transform wildly; the real problem is the lack of any consistent relationship in the transformed data. Thus it is easy to produce a color display but hard to analyze what the colors mean. This program attempts to align the selection of colors so that certain classes have certain colors, and this assignment will be relatively independent of which feature selection program produced the reduction in dimensionality.

This program is applicable to raw single pass data. Even confused classes (as, for example, wheat and grass are likely to be) can be displayed with some enhancement of their separation.

Source Language. FORTRAN IV 100%

Restrictions. Name ROTAT3 is reserved (the name of a common block).

If the wheat vector points to the "middle" of four dimensional data some separation may be lost.

I/O Configuration. Both programs are I/O free.

Deck Set-up.

- Job Control Cards
- Calling program*
- ROTAT1
- ROTAT2
- FOLD64
- Job Control Cards
- Data

*for comments on the calling program, see Usage.

Usage. Calling sequence, ROTAT1

CALL ROTAT1(NCLASS,BMU,BSBTD,IW,IG), where

NCLASS number of classes for which training statistics
 are given

BMU mean vector for each class: NCLASS by 4

BSBTD diagonal elements of covariance matrix: NCLASS by 4

IW class to be made red

IG class to be made yellow

Calling sequence, ROTAT2

CALL ROTAT2(DATA,TDATA,LDA), where

DATA input row, type real, LDA by 4

TDATA output row, type real, LDA by 4

LDA number of pixels in a row

Comments on the calling program. The calling program may have a
general flow as follows:

1. Set up data set with data to be transformed and transformed data (may be the same data set since input is only needed once).
2. Input or otherwise determine parameters for ROTAT1.
3. Call ROTAT1.
4. Intialize a row counter loop.
5. Input a row of data.
6. Convert to type REAL if necessary and store in DATA.
7. Call ROTAT2.
8. Store a row of data, packing or converting from type REAL if necessary.
9. Next row.

Timing. ROTAT2 crunches numbers; for each pixel, about 25 each fixed point additions and multiplications, 24 floating point additions and 20 floating point multiplications are required.

Storage Requirements. Approximately 7000 8 bit bytes code for both ROTAT1 and ROTAT2; array storage (assuming 32 bit real numbers) will be about $4 * (91 + 8 * (NCLASS + LDA))$ 8 bit bytes. Exact storage requirements will depend on computer type.

Possible Extensions, Suggestions for Improvements.

1. As can be readily seen from the example, the blue channel has maximum value less than 0.5 in each of the six classes. This may be acceptable; however, should the user wish, this is easily adjusted by the following changes in ROTAT1:

Change	to
$C(3,2) = -C(2,3)$	$C(3,2) = -2.*C(2,3)$
$C(3,3) = C(2,2)$	$C(3,3) = 2.*C(2,2)$
2. A similar change can be made in the red channel so that wheat will be pure red; this will not be as violent as the map in NONLN1. The change is
 $C(1,1) = 1.$ to $C(1,1) = 1./WNORM.$
The effect of these two changes is displayed in the mathematical description of ROTAT1-2. Changes 1 and 2 have been made in the version of ROTAT1 delivered.
3. ROTAT2 can be speeded up at some cost in complication. This amounts to writing out the loop which performs the multiplication

of a vector of data by E. The reason it will be faster is that the current version uses 16 references to array E and 16 floating point multiplications while actually E is zero in 6 of these places. This change would probably speed up ROTAT2 by 40 percent.

Additional Information.

Mathematical description: ROTAT1 and ROTAT2

Table 1. Local variables: ROTAT1

Table 2. Variables in COMMON block: ROTAT3

Table 3. Local variables: ROTAT2

General flow chart: ROTAT1

General flow chart: ROTAT2

General flow chart: Test program

Listing

Test program and test listing

Mathematical description: ROTAT1 and ROTAT2

The idea behind this transformation is to align the color display in a reproducible fashion while maintaining the original geometric relationships. Since this is generally not possible, the program introduces mild nonlinearities.

Program ROTAT1 defines the transformation; ROTAT2 applies it to one row of data at a time. There is no assumption made anywhere that the input data has any certain range. ROTAT first examines a pixel and decides if one of its four coordinates is out of the interval

(S3MIN(I),S3MAX(I)) , I = 1 to 4 , the minimum and maximum of mean \pm 3 sigma over NCLASS classes per coordinate. Failure of any coordinate to lie in the interval results in a zero output in each coordinate and movement to the next pixel.

Acceptable data is now mapped into numbers which are generally between 0 and 1 in each coordinate by the map $y = (x-m)/(M-m)$, where x is the input coordinate, M is the maximum mean + sigma over NCLASS classes, m is the minimum mean - sigma over NCLASS classes, and y is the output value. Depending on a parameter supplied by ROTAT1, either y or $-1 + y$ is stored. (We will see this has the effect of complementing data in that coordinate.)

If Y denotes the 4-vector of data at this point, the transformation $X = E Y$ is applied; E is a 4x4 matrix which is a permutation of 64 times the product of two rotations, described in detail below. Output X is now folded by the scalar function f with $f(64 \cdot I) = 0$, I an even integer, $f(64 \cdot I) = 64$, I an odd integer and f linear between. ROTAT2 goes to the next pixel and returns when LDA pixels have been transformed.

We now describe ROTAT1 which supplies all these parameters. ROTAT1 first examines the NCLASS means and variances in each coordinate to determine S1MAX,S1MIN,S3MAX and S3MIN--the maximum and minimum of mean \pm sigma and mean \pm 3 sigma over all classes in each coordinate. The program then transforms means for wheat and grass into the unit 4-cube and selects three of the four indices as follows:

- i_r is selected to maximize the separation of wheat and grass
- i_g is selected to next maximize the separation of wheat and grass
- i_b is selected to minimize the separation of wheat and grass
- i_0 is the other coordinate index.

Thus, if w and g denote the images of wheat and grass in the unit 4-cube, we have

$$|w(i_b) - g(i_b)| \leq |w(i_0) - g(i_0)| \leq |w(i_g) - g(i_g)| \leq |w(i_r) - g(i_r)| .$$

The program examines $w(i_r)$, $w(i_g)$ and $w(i_b)$ as follows:

If $w(i_r) < 0.5$, complement data in coordinate i_r ; that is, replace a transformed value $x(i_r)$ by $1 - x(i_r)$. Similarly, complement data in coordinate i_g when $w(i_g) > 0.5$ and complement data in coordinate i_b when $w(i_b) > 0.5$. If $g(i_g) < g(i_b)$, interchange coordinates i_g and i_b . Fig. 1 depicts the result of performing these transformations to test program data; 1 is grass and 5 is wheat. Note that 5 is as near the i_r axis as it is possible to make it with this kind of mapping.

Now move the transformed vector 5 (which we again denote by w) to lie on the i_r -axis by first rotating into the $i_r - i_b$ plane about the i_b axis and then onto the i_r axis by rotating about the i_g axis. If $w_r = w(i_r)$, $w_g = w(i_g)$ and $w_b = w(i_b)$, then this mapping is given by

$$A = \begin{pmatrix} \frac{w_r}{||w||} & \frac{w_g}{||w||} & \frac{w_b}{||w||} \\ -\frac{v}{||w||} & \frac{w_r w_g}{v ||w||} & \frac{w_r w_b}{v ||w||} \\ 0 & -\frac{w_b}{v} & \frac{w_b}{v} \end{pmatrix}$$

where $w = \sqrt{w_r^2 + w_g^2 + w_b^2}$ and $v = \sqrt{w_g^2 + w_b^2}$. Figure 2 shows the result of applying A to the six mean vectors and then plotting the absolute value of each coordinate. For our data,

$$A = \begin{pmatrix} .828 & .394 & .399 \\ -.561 & .581 & .589 \\ 0 & -.712 & .702 \end{pmatrix}$$

Next the data is rotated about the i_r axis so that the transformed g vector will lie in the $i_r - i_g$ plane. This rotation is given by

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{g(i_g)}{n} & \frac{g(i_b)}{n} \\ 0 & -\frac{g(i_b)}{n} & \frac{g(i_g)}{n} \end{pmatrix}$$

where $n = \sqrt{g(i_g)^2 + g(i_b)^2}$ and g denotes the transformed grass vector.

In our test data, $g(i_b)$ is very small, so that this transformation does not do much. We have

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -.999 & -.036 \\ 0 & .036 & -.999 \end{pmatrix}$$

The product of C and A maps mean vectors as shown in Fig. 3 (with absolute values taken). We have

$$CA = \begin{pmatrix} .828 & .394 & .399 \\ .561 & -.556 & -.614 \\ -.020 & .732 & -.681 \end{pmatrix}$$

Before continuing our description of ROTAT1, we comment on suggested changes (change 1 and 2): Note that the wheat vector is not as large as grass, even in coordinate i_r . If (this is change 2) $C(1,1)$ is replaced by $1/||w||$, then we obtain the result shown in Fig. 4. (All discussion from now on refers to plots of folded (rather than clipped) vectors. This is exactly what program ROTAT2 does.) Necessarily, $||w|| > \frac{1}{2}$; it follows that the noise should not be amplified very much by this change.

In Fig. 5 we display the affect of change 1 only, and Fig. 6 the effect of both changes. Since these changes make the transformation less accurate geometrically, they have the potential to degrade separation and amplify noise. We therefore generated and plotted random data with statistics like those used as input to NONLN1 and plotted (for comparison) the effect of changes 1 and 2. Some of this is presented here. Fig. 7 shows wheat and grass random vectors as transformed by the unchanged method; Fig. 8 shows the same data under change 1 and 2.

(The line segments are drawn from the point in 3-space to the nearest coordinate plane.) In Fig. 9, the upper plot shows all six classes after change 1 and 2. The lower plot shows the original method. (Twenty points in each class may have been too many; the plot is somewhat confusing.)

As a result of these and other studies we have decided to deliver the version of NONLN1 with change 1 and 2 implemented. Final decision which version to use will have to be based on actual real data as viewed in color.

Returning to the description of NONLN1, a matrix E is defined which is a permutation of the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & & & \\ 0 & CA & & \\ 0 & & & \end{pmatrix}_{4 \times 4}$$

so that coordinate i_r goes to channel 4, i_g to channel 1, i_b to channel 2 and i_0 to channel 3. Also, a multiplication of ± 64 is performed to each row (depending on whether that row is complemented).

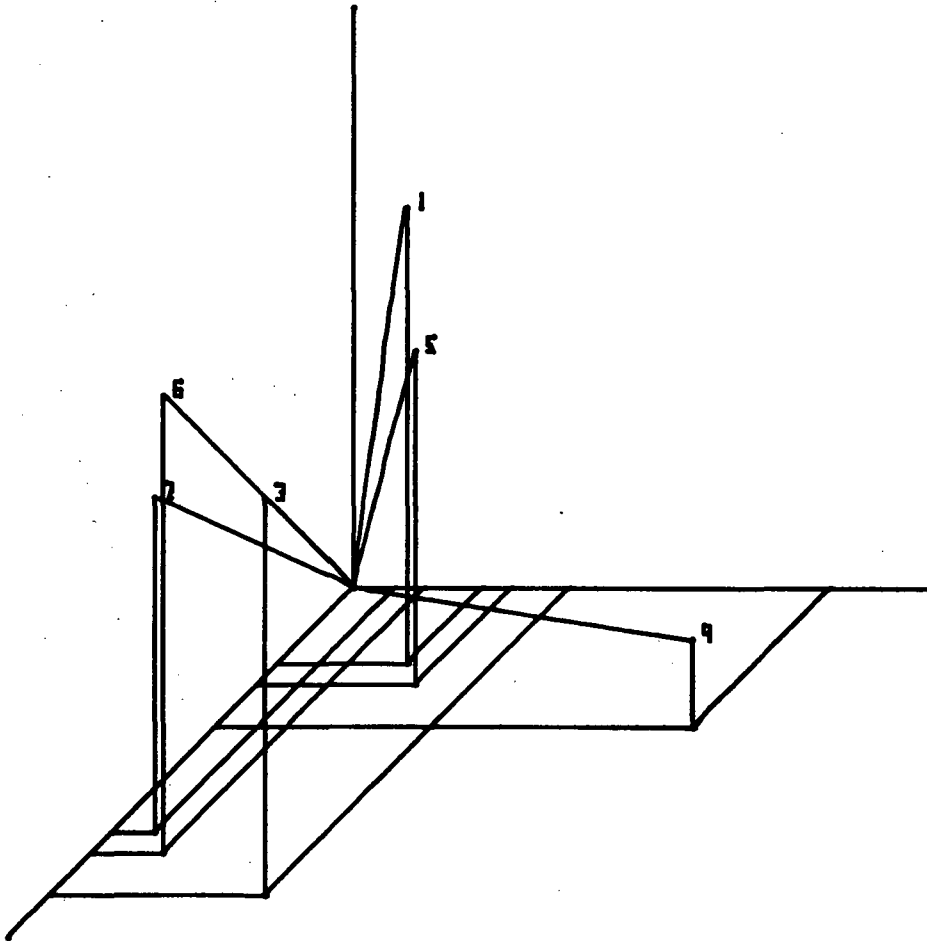


Fig. 1. Coordinates i_r , i_g and i_b of scaled and complemented row mean vectors.

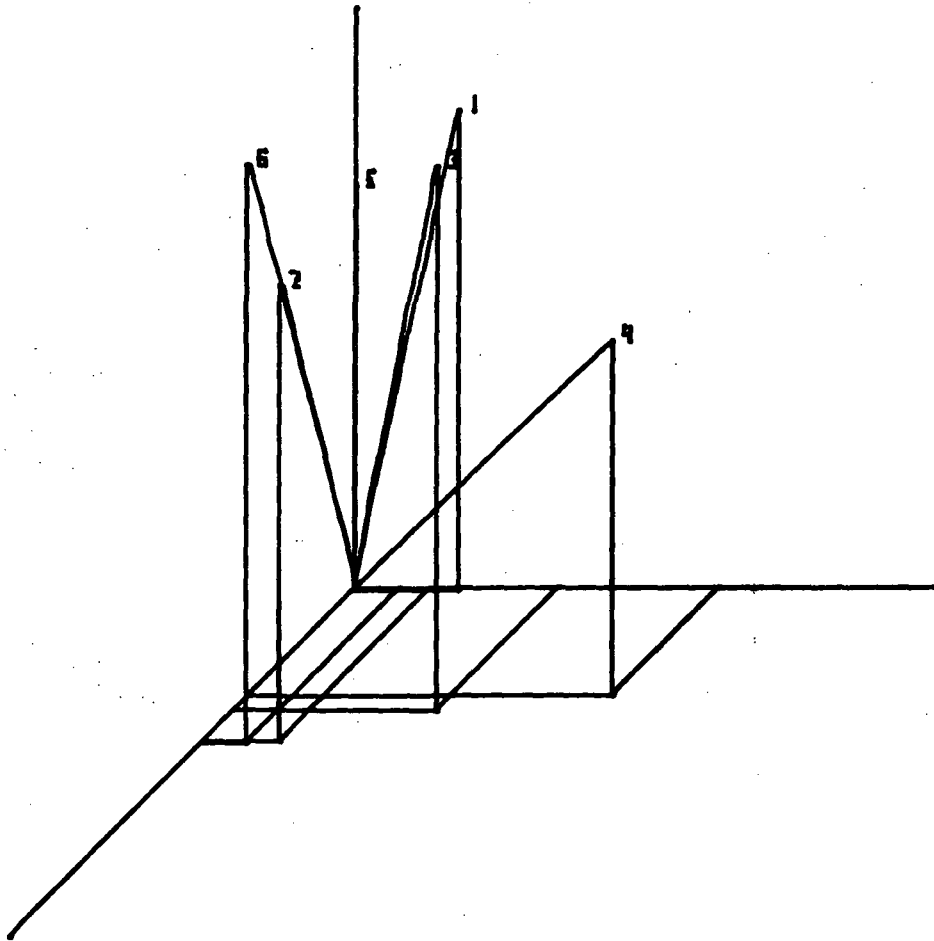


Fig. 2. Absolute values of transformation of Fig. 1 by A.

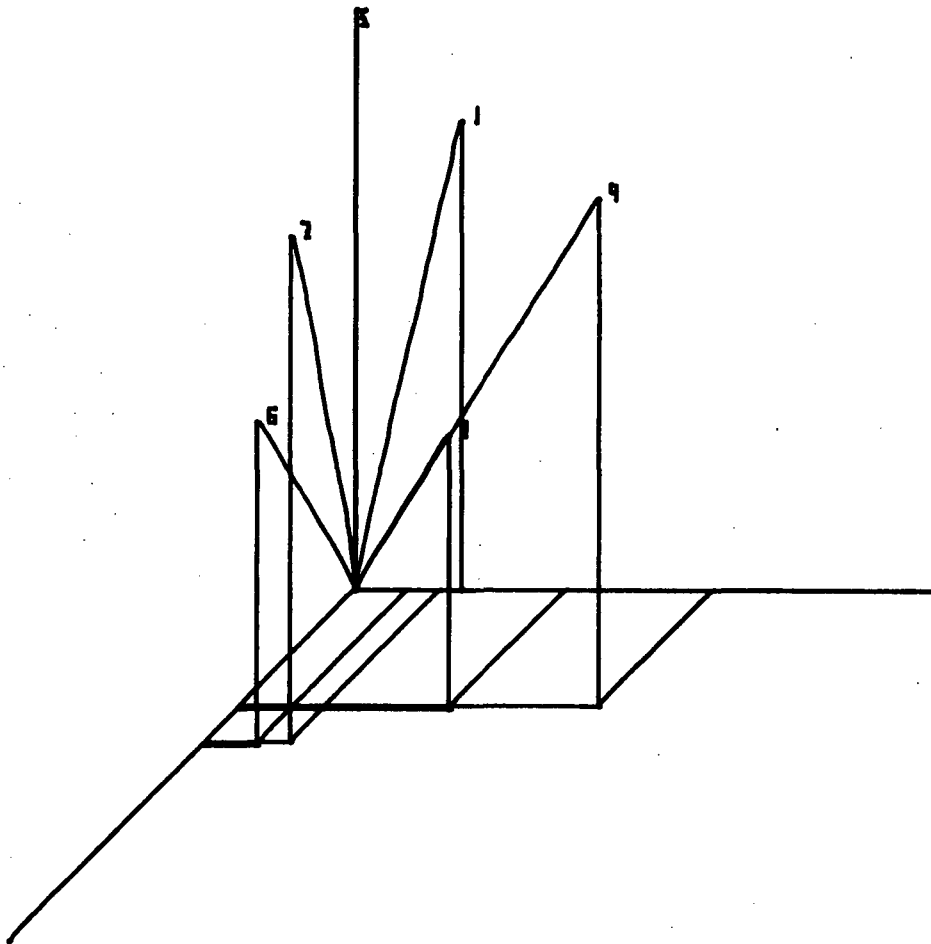


Fig. 4. Change 2 applied to C to make WHEAT redder.

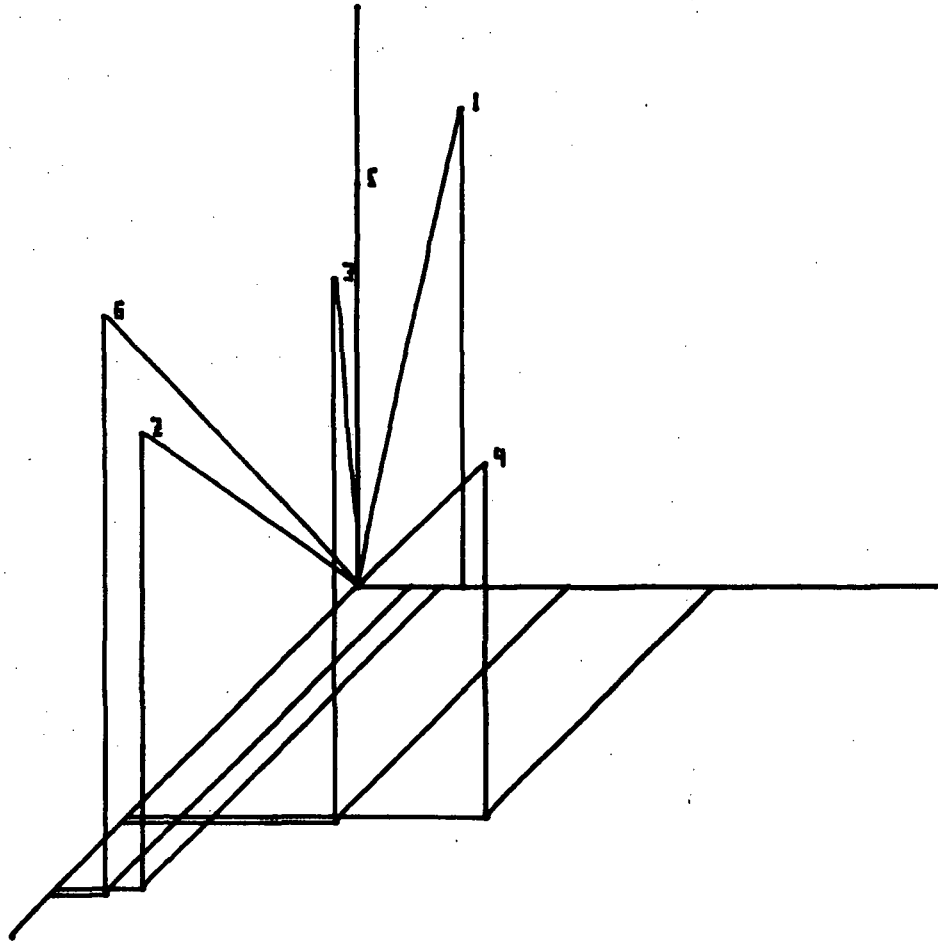


Fig. 5. Change 1 applied to C to increase spread in blue channel.

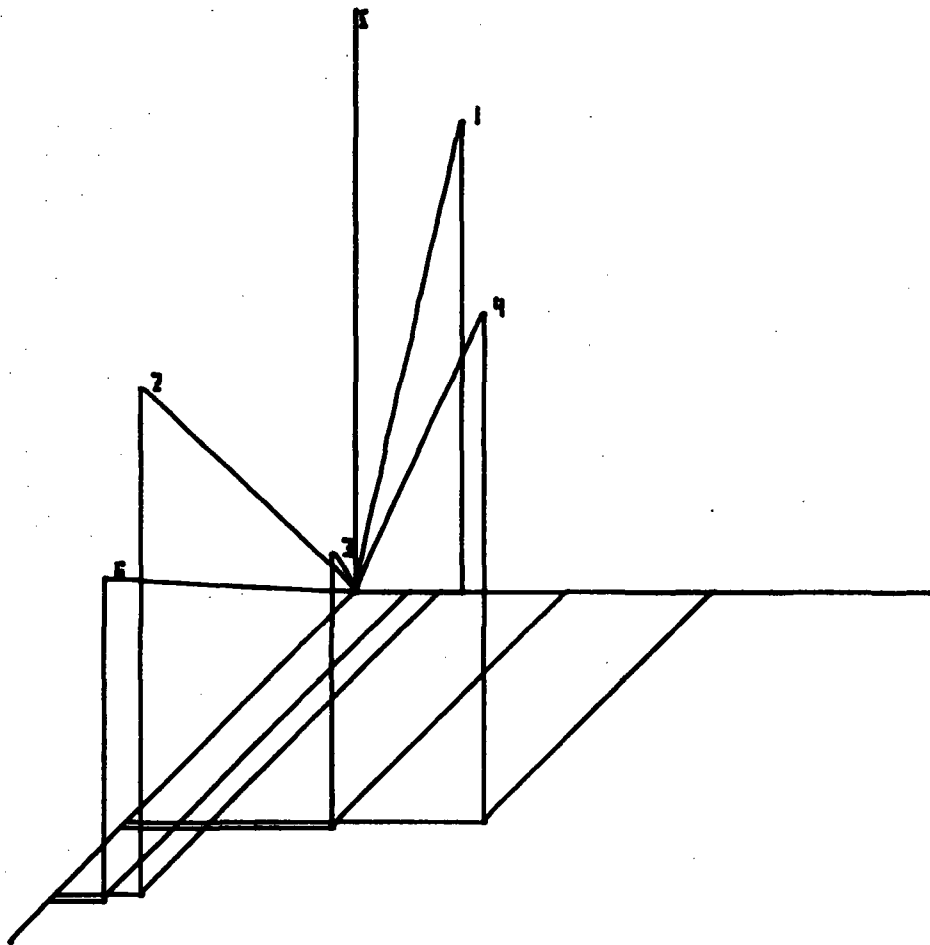


Fig. 6. Change 1 and 2.

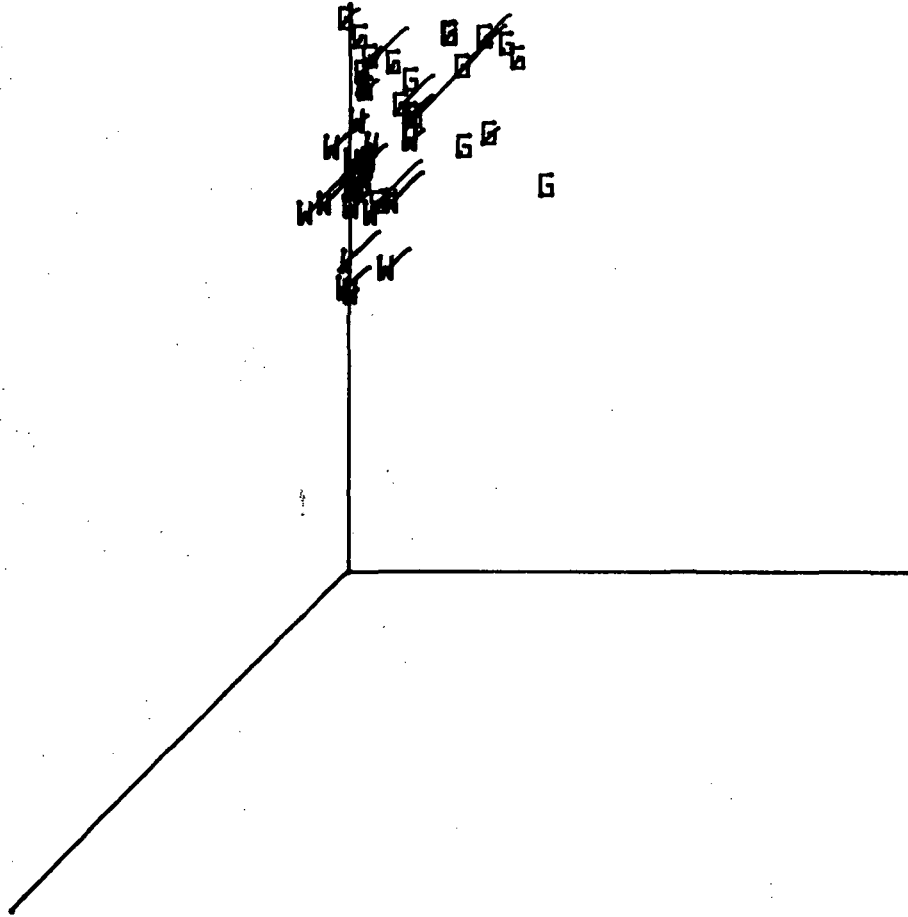


Fig. 7. Random data transformed, original method.

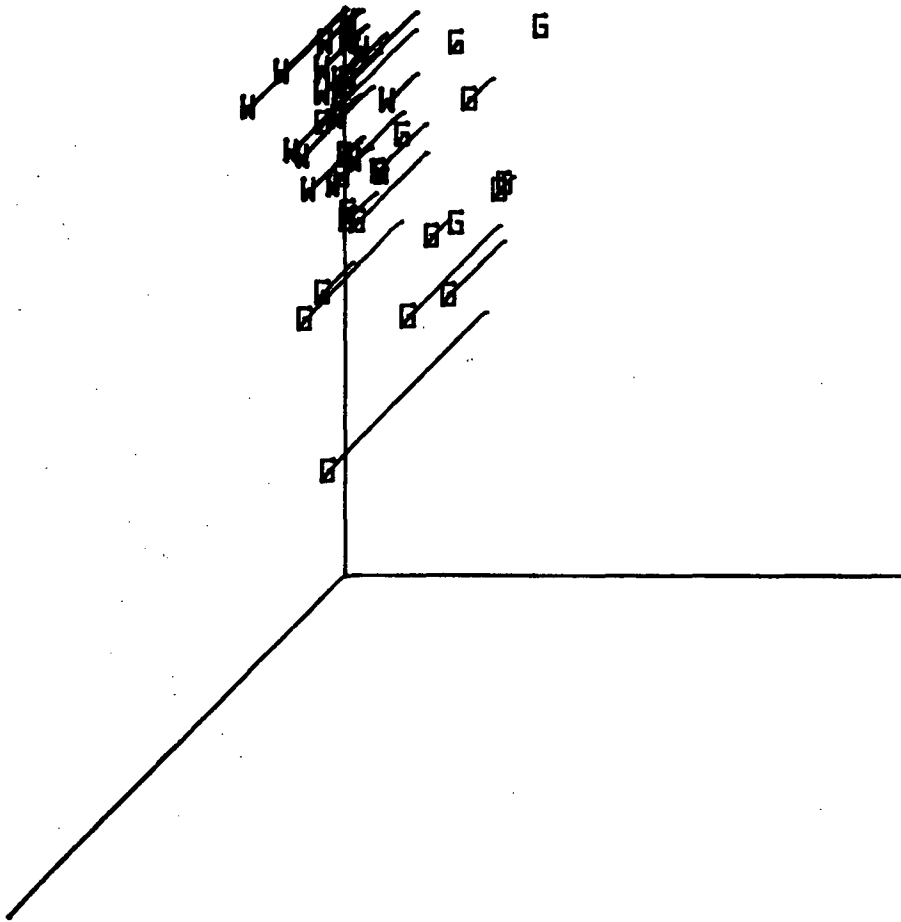


Fig. 8. Random data, change 1 and 2.

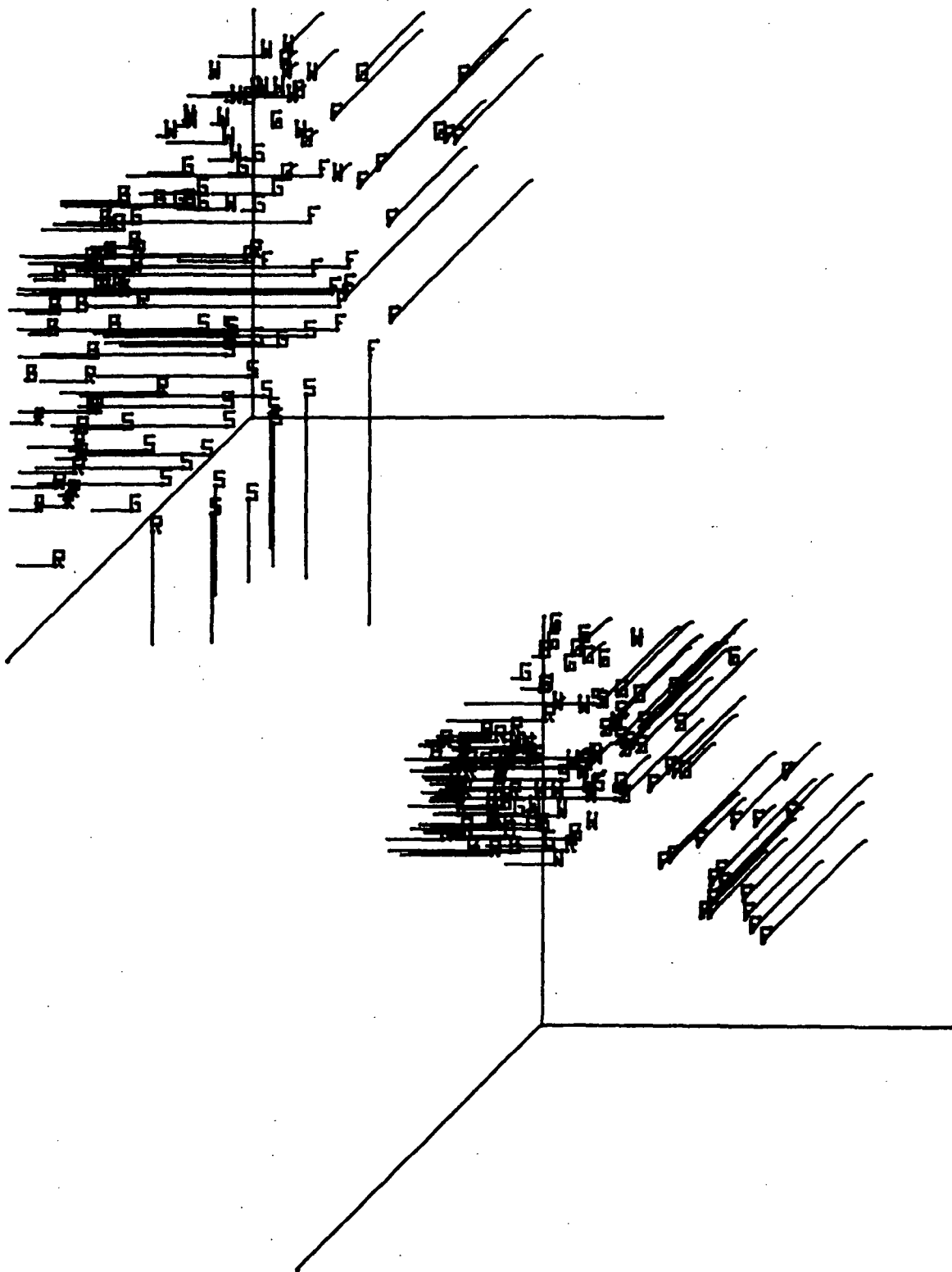


Fig. 9. (Upper) Change 1 and 2, six classes.
(Lower) Original, six classes.

TABLE 1. Local Variables : NONLN1

I	DO loop index
J	DO loop index
NCLASS	Number of classes for which training statistics are furnished
SIG	Temporary value of standard deviation
BSBTD(J,I)	Diagonal of covariance matrix in channel I, class J
BMU(J,I)	Mean, channel I, class J
TEMP	Temporary real variable used for getting maximum or minimum
IW	Class number of wheat
IG	Class number of grass
G(I,1)	Image in unit cube of grass mean vector
G(I,2)	1 - G(I,1)
W(I,1)	Image in unit cube of wheat mean vector
W(I,2)	1 - W(I,1)
K1	Index i_r
L1	A flag : L1 = 1, do not complement in class K1 L2 = 2, do complement in class K1
K2	Index i_g
L2	Complement flag for class K2
K3	Index i_b
L3	Complement flag for class K3
K4	10 - (K1+K2+K3)
XL(4)*	XL(K1) = 1 - L1 ; XL(K2) = 1 - L2 ; XL(K3) = 1 - L3 ; LX(K4) = 0 .

TABLE 1. (Continued)

WNORM	$ w $
W23	v
A(3,3)	The first transformation.
GH(3)	The transformed by A of the grass vector
G23	$\sqrt{(g(i_g))^2 + g(i_b)^2}$
C(3,3)	The rotation about i_r to annihilate the 3 component of GH.
D(4,4)	$\begin{pmatrix} CA & 0 \\ & 0 \\ & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
E(4,4)*	A permutation of D to give standard colors, taking into account K1-K4 .

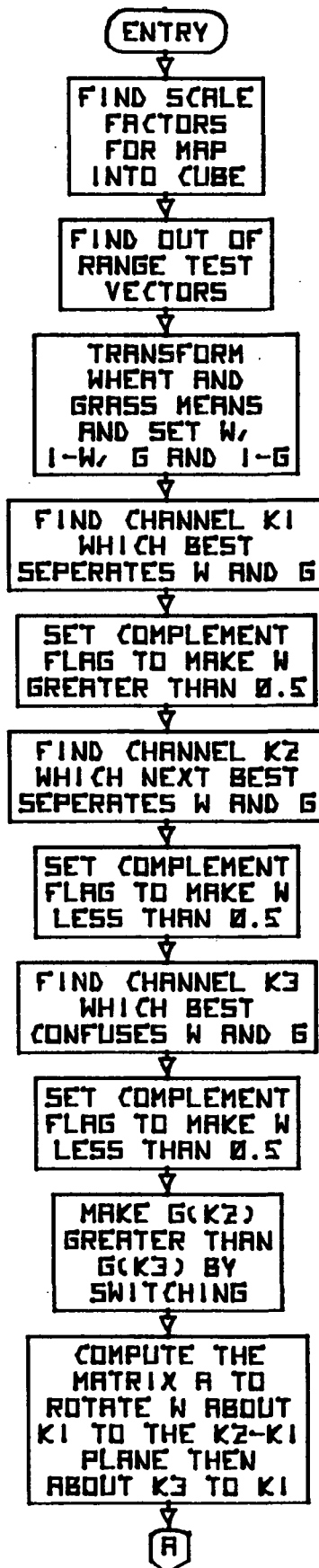
* These variables in COMMON block ROTAT3

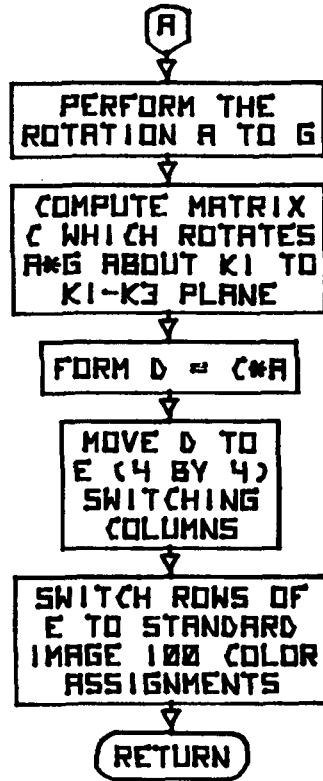
TABLE 2. Variables in COMMON block : ROTAT3

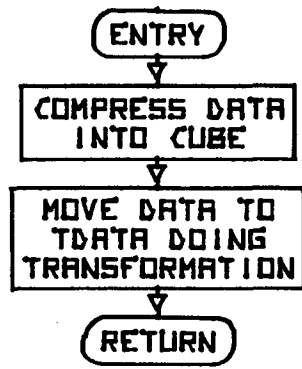
S1MIN(4)	Minimum over classes of mean in channel I minus sigma in channel I, I = 1,4
S1MAX(4)	$1./(\text{S1MAX}(I) - \text{S1MIN}(I))$, where S1MAX(I) is first set to be the maximum over classes of mean plus sigma in channel I.
S3MIN(4) S3MAX(4)	Minimum (respectively maximum) of mean - 3 sigma (respectively mean + 3 sigma) over classes for channel I.
XL(4)	Vector used in complementing data
E(4,4)	Matrix used to perform transformation

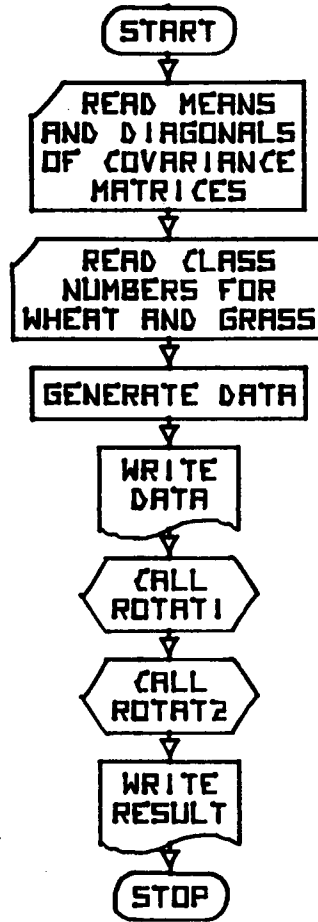
TABLE 3. Local Variables : ROTAT2

DATA(LDA,4)	One row of four dimensional real data to be transformed
TDATA(LDA,4)	The transformed data
LDA	Number of pixels per row
IPT	DO loop index--pixel numbers
ICHNL	DO loop index--channel number
TEMP	Temporary real variable
SUM	Temporary real variable used to generate vector-matrix product









```
C
C   SUBROUTINE ROTAT1 (NCLASS,BMU,BSBTD,IW,IG)
C
C   FUNCTION - TO SET PARAMETERS FOR SUBROUTINE ROTAT2
C   USAGE    - CALL RCTAT1(NCLASS, BMU, ESBTD, IW, IG)
C   PARAMETERS -
C     NCLASS - NUMBER OF CLASSES FOR WHICH TRAINING STATISTICS EXIST
C     BMU    - MEAN VECTOR FOR EACH CLASS:  NCLASS BY 4
C     ESBTD  - DIAGONAL ELEMENTS OF COVARIANCE MATRIX:  NCLASS BY 4
C     IW     - CLASS NUMBER OF CLASS TO BE MADE RED
C     IG     - CLASS NUMBER OF CLASS TO BE MADE YELLOW-GREEN
C   PRECISION - SINGLE
C   RECD. RTNS. - NONE (S INGLE PURPOSE--CALLED) BEFORE ROTAT2)
C   RESTRICTIONS
C     ROTAT3 - A RESERVED NAME OF A COMMON BLOCK
C   LANGUAGE - FORTRAN IV
```

```
C
C   SUBROUTINE ROTAT1(NCLASS,BMU,BSBTD,IG,IW)
C   DIMENSION BMU(NCLASS,4),BSBTD(NCLASS,4),W(4,2),G(4,2),A(3,3),
C   $ GH(3),D(4,4),C(3,3)
C   COMMON /ROTAT3/SIMIN(4),SIMAX(4),S3MIN(4),S3MAX(4),XL(4),E(4,4)
```

```
C
C   FIND FUDGE FACTORS FOR CUBE
```

```
C
C
C   DO 10 I=1,4
C     SIMIN(I)=BMU(1,I)-SQRT(BSBTD(1,I))
C     SIMAX(I)=SIMIN(I)
C     S3MIN(I)=BMU(1,I)-3*SQRT(BSBTD(1,I))
C     S3MAX(I)=S3MIN(I)
C     DO 10 J=1,NCLASS
C       SIG=SQRT(BSBTD(J,I))
C       TEMP=BMU(J,I)-SIG
C       IF(TEMP.GT.SIMIN(I))GO TO 20
C       SIMIN(I)=TEMP
20  TEMP=BMU(J,I)+SIG
C       IF(TEMP.LT.SIMAX(I))GO TO 12
C       SIMAX(I)=TEMP
12  TEMP=BMU(J,I)-3*SIG
C       IF(TEMP.GT.S3MIN(I))GO TO 14
C       S3MIN(I)=TEMP
14  TEMP=BMU(J,I)+3*SIG
C       IF(TEMP.LT.S3MAX(I))GO TO 10
C       S3MAX(I)=TEMP
10  CONTINUE
C     DO 15 I=1,4
15  SIMAX(I)=1./(SIMAX(I)-SIMIN(I))
```

```
C
C   NORMALIZE WHEAT AND GRASS MEANS
C
      DO 30 I=1,4
      G(I,1)=(BMU(IG,I)-SIMIN(I))*SIMAX(I)
      W(I,1)=(BML(IW,I)-SIMIN(I))*SIMAX(I)
      W(I,2)=1.0-W(I,1)
30   G(I,2)=1.0-G(I,1)
C
C   FIND K1 AND L1
C
      TEMP=0.0
      DO 40 I=1,4
      IF ( TEMP .GT. ABS(G(I,1)-W(I,1))) GO TO 40
      TEMP = ABS(G(I,1)-W(I,1))
      K1 = I
40   CONTINUE
      L1 = 1
      IF( W(K1,1) .LT. 0.5 ) L1 = 2
      XL(K1) = 1 - L1
C
C   FIND K2 AND L2
C
      TEMP = 0.0
      DO 50 I=1,4
      IF ( I .EQ. K1 ) GO TO 50
      IF ( TEMP .GT. ABS(G(I,1)-W(I,1)) ) GO TO 50
      TEMP = ABS(G(I,1)-W(I,1))
      K2 = I
50   CONTINUE
      L2 = 1
      IF ( W(K2,1) .GT. 0.5 ) L2 = 2
      XL(K2) = 1 - L2
C
C   FIND K3 AND L3
C
      TEMP = 2.0
      DO 60 I=1,4
      IF ( I .EQ. K1 .OR. I .EQ. K2 ) GO TO 60
      IF ( TEMP .LT. ABS(G(I,1) - W(I,1)) ) GO TO 60
      TEMP = ABS(G(I,1) - W(I,1))
      K3 = I
60   CONTINUE
      L3 = 1
      IF ( W(K3,1) .GT. 0.5 ) L3 = 2
      XL(K3) = 1 - L3
```

```
C
C   IF THE 2 COMPONENT OF GRASS IS SMALLER THAN THE 3 COMPONENT,
C   THEN SWITCH THE 2 AND 3 INDICES
C
  IF(G(K2,L2).GT.G(K3,L3)) GO TO 70
  I=K2
  K2=K3
  K3=I
  I=L2
  L2=L3
  L3=I
  TEMP=XL(K3)
  XL(K3)=XL(K2)
  XL(K2)=TEMP
C
C   COMPUTE A MATRIX
C
70  WNORM=SQRT(W(K1,L1)**2+W(K2,L2)**2+W(K3,L3)**2)
    W23=SQRT(W(K2,L2)**2+W(K3,L3)**2)
    A(1,1)=W(K1,L1)/WNORM
    A(1,2)=W(K2,L2)/WNORM
    A(1,3)=W(K3,L3)/WNORM
    A(2,1)=-W23/WNORM
    A(2,2)=W(K1,L1)*W(K2,L2)/(W23*WNORM)
    A(2,3)=W(K1,L1)*W(K3,L3)/(W23*WNORM)
    A(3,1)=0.
    A(3,2)=-W(K3,L3)/W23
    A(3,3)=W(K2,L2)/W23
C
C   DO THE A ROTATION ON THE GRASS VECTOR
C
  DO 80 I=1,3
80  GH(I)=G(K1,L1)*A(I,1)+G(K2,L2)*A(I,2)+G(K3,L3)*A(I,3)
C
C   GET THE C MATRIX FROM THE ROTATED GRASS VECTOR
C
  G23=SQRT(GH(2)**2+GH(3)**2)
  C(1,1)=1.0/WNORM
  C(1,2)=0.
  C(1,3)=0.
  C(2,1)=0.
  C(2,2)=GH(2)/G23
  C(2,3)=GH(3)/G23
  C(3,1)=0.
  C(3,2)=-C(2,3)*2.0
  C(3,3)=C(2,2)*2.0
```



```
C
C      FROM THE EIG C MATRIX
C
      DO 90 I=1,3
      D(4,I)=0.
      D(I,4)=0.
      DO 90 J=1,3
      TEMP=0.
      DO 100 L=1,3
100  TEMP=TEMP+C(I,L)*A(L,J)
      90  D(I,J)=TEMP
      D(4,4)=1.
C
C      MOVE C TO E SWITCHING COLUMNS
C
      K4=10-K1-K2-K3
      XL(K4)=0.
      DO 110 I=1,4
      E(I,K1)=64.*D(I,1)*((-1.)**(1+L1))
      E(I,K2)=64.*D(I,2)*((-1.)**(1+L2))
      E(I,K3)=64.*D(I,3)*((-1.)**(1+L3))
110  E(I,K4)=D(I,4)*64.
C
C      SWITCH ROWS TO STANDARD IMAGE 100 COLORS
C
      DO 120 I=1,4
      TEMP=E(I,I)
      DO 130 J=1,3
      JJ=J+1
130  E(J,I)=E(JJ,I)
120  E(4,I)=TEMP
      RETURN
      END
```

```
C
C SUBROUTINE ROTAT2 (DATA,TDATA,LDA)
C FUNCTION - TO TRANSFORM FOUR DIMENSIONAL REAL DATA A ROW AT A
C TIME TO ANOTHER REAL ROW OF FOUR DIMENSIONAL DATA
C IN THE RANGE 0.-64. ATTEMPTING TO MAKE CLASS IV
C RED (HIGH IN CHANNEL 4, LOW IN 1 AND 2) AND CLASS
C IG GREEN (HIGH IN CHANNEL 1, LOW IN 2 AND 4).
C USAGE - CALL ROTAT2 (DATA,TDATA,LDA)
C PARAMETERS -
C DATA - ONE ROW OF FOUR DIMENSIONAL REAL DATA: LDA BY 4
C TDATA - THE TRANSFORMED DATA: LDA BY 4
C LDA - THE NUMBER OF PIXELS IN A ROW
C PRECISION - SINGLE
C REQD. RTNS. - ROTAT1 MUST BE CALLED TO SET PROGRAM VARIABLES
C FUNCTION FOLD64 IS REFERENCED
C RESTRICTIONS
C ROTAT3 - A RESERVED NAME OF A COMMON BLOCK
C LANGUAGE - FORTRAN IV
C
C SUBROUTINE ROTAT2(DATA,TDATA,LDA)
C COMMON /ROTAT3/SIMIN(4),S1MAX(4),S3MIN(4),S3MAX(4),XL(4),E(4,4)
C DIMENSION DATA(LDA,4),TDATA(LDA,4)
C LOGICAL LTEMP
C
C COMPRESS DATA INTO CUBE
C
C DO 170 IPNT=1,LDA
C LTEMP=.TRUE.
C DO 172 ICHNL=1,4
C TEMP=DATA(IPNT,ICHNL)
C IF (TEMP.GT.S3MAX(ICHNL).OR.TEMP.LT.S3MIN(ICHNL))GO TO 176
172 CONTINUE
C DO 178 ICHNL=1,4
178 DATA(IPNT,ICHNL)=XL(ICHNL)+(DATA(IPNT,ICHNL)-S1MIN(ICHNL))
C $*S1MAX(ICHNL)
C LTEMP=.FALSE.
176 CONTINUE
C
C DC TRANSFORMATION
C
C DO 180 ICHNL=1,4
C SUM=0.
C IF(LTEMP)GO TO 180
C DO 185 I=1,4
185 SUM=SUM+E(ICHNL,I)*DATA(IPNT,I)
C
C PUT DATA INTO RANGE 0-64
C
180 TDATA(IPNT,ICHNL)=FOLD64(SUM)
170 CONTINUE
C RETURN
C END
```

C
C FUNCTION - TO MAP A REAL NUMBER INTO ONE BETWEEN 0. AND 64.
C BY "FOLDING" RATHER THAN CLIPPING
C USAGE - $Y = \text{FOLD64}(X)$
C
C PARAMETERS -
C X - ANY REAL NUMBER
C Y - THE RESULTANT BETWEEN 0. AND 64. Y IS CONGRUENT TO
C X (MOD 64) IF X IS IN AN INTERVAL $(64 * I, 64 * (I + 1))$
C WITH I EVEN; OTHERWISE Y IS CONGRUENT TO $64 - X \pmod{64}$
C RESTRICTIONS NONE
C X SHOULD NOT BE TOO LARGE OR EFFICIENCY SUFFERS
C PRECISION - SINGLE
C LANGUAGE - FORTRAN IV
C

```
FUNCTION FOLD64(X)
  FOLD64=ABS(X)
  IF(FOLD64.LE.64.)RETURN
1 FOLD64=ABS(FOLD64-128.)
  IF(FOLD64.GT.64.)GOTO 1
  RETURN
END
```