General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
The Recursive Maximum Likelihood Proportion Estimator—User's Guide and Test Results

by

D. L. Van Rooy
Rice University

ABSTRACT:

In this report, we describe our implementation of the recursive maximum likelihood proportion estimator proposed by D. Kazakos in "Recursive Estimation of Prior Probabilities Using the Mixture Approach," (Rice University, ICSA Technical Report #275-025-019). A user's guide to the programs as they currently exist on the IBM 360/07 at LARS, Purdue is included, and test results on LANDSAT data are described. On Hill County data, the algorithm yields results comparable to the standard maximum likelihood proportion estimator.

Institute for Computer Services and Applications
Rice University
Houston, TX 77001
June, 1976

This work was supported under NASA contract NAS 9-12776
I. Introduction:

In this report, we describe our implementation of the recursive maximum likelihood proportion estimator proposed by D. Kazakos in [1]. Numerical results obtained with this algorithm using LANDSAT data are described, and a user's guide for the programs as they currently exist on the IBM 360/67 at LARS (terminal available at NASA-JSC) is included.

Section II contains a description of the algorithm as implemented. Section III serves as a user's guide to the programs available. In section IV, we describe the numerical results we have obtained with this algorithm. An appendix contains listings of the programs.

II. The Algorithm:

Given a set of n-dimensional measurement vectors \{\mathbf{x}\} from \(M\) normally distributed multivariate pattern classes \(H^j\), \(j = 1, 2, \ldots, M\) the \(M-1\) dimensional recursive maximum likelihood proportion estimate (RMLPE)\(^{(1)}\) \(p^i\) at the \(i^{th}\) data vector is given by

\[
p^i = p^{i-1} + \frac{1}{i} L \left[ g \left( p^{i-1}, \mathbf{x}_i \right) \right]^{-1} \left( f_1(x_i) - f_M(x_i), f_2(x_i) - f_M(x_i), \ldots, f_{M-1}(x_i) - f_M(x_i) \right) \]

where \(f_j(x)\) is the density function for the \(j^{th}\) class;
\[ f_j(x) = (2\pi)^{-n/2} |\mathbf{K}_j|^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} (x-\mathbf{u}_j)^T \mathbf{K}_j^{-1} (x-\mathbf{u}_j) \right] \tag{2} \]

where \( \mathbf{u}_j \) and \( \mathbf{K}_j \) are the mean and covariance matrix, respectively, for the \( j \)th class; \( g(p^{i-1}, x_i) \) is the mixture distribution estimate, i.e.,

\[ g(p^{i-1}, x_i) = f_M(x_i) + \sum_{\ell=1}^{m-1} p^{i-1}_\ell (f_\ell(x_i) - f_M(x_i)) \tag{3} \]

and \( L \) is a suitably chosen constant in this approximation. The proportion estimate for the \( M \)th class is denoted by \( p^i_M \) and given by

\[ p^i_M = 1 - \sum_{\ell=1}^{m-1} p^i_\ell \tag{4} \]

In our implementation of this algorithm, we have made several modifications to improve its performance. These include (1) clipping the value of the update (i.e., second) term in eq. (1); (2) renormalizing the \( p^i \) at each step so that all \( p^i_j \geq 0 \) and \( \sum_{\ell=1}^{m} p^i_\ell = 1 \); and (3) introducing an additional damping term in the update term of eq. (1). The final form of the algorithm is

\[ p^i = \text{NORM} \left\{ \epsilon, p^{i-1} + \frac{1}{i+n_0} \text{LMT} [T, L g(p^{i-1}, x_i) \cdot (f_1(x_i) - f_M(x_i), f_2(x_i) - f_M(x_i), \ldots, f_{M-1}(x_i) - f_M(x_i))] \right\} \]

where \( \text{LMT}(a, b) \) is the clipping function defined by
LMT(a, b) = \tilde{b}

with \quad \tilde{b}_i = \text{sign}(b_i) \min \left( a, |b_i| \right)

NORM is the renormalizing function defined by

NORM(\epsilon, y) = \text{the first M-1 elements of } \tilde{y}

where

\tilde{y}_m = 1 - \sum_{i=1}^{m-1} y_i

y_i \quad y_i

If

\min(\tilde{y}_i) \geq \epsilon > 0 \text{ then finish else}

\tilde{y}_i - y_i - \min(\tilde{y}_i) + \epsilon \quad i = 1, 2, \ldots M

\tilde{y}_i - \tilde{y}_i / \sum_{k=1}^{m} \tilde{y}_i

and \( n_o \) is a positive constant used to damp out early oscillations of the estimate.

Two other algorithms used in conjunction with this one are (1) an algorithm to calculate an approximation to \( L \) and (2) an algorithm to scramble all of the data (the RMLPE uses the stochastic approximation, so the data needs to appear in a random order). The first algorithm calculates the following approximation to \( L \)

\[ L = \left( u \cdot \min_i |K_j|^{\frac{1}{2}} \right)^{-1} \]
where \( u \) is the minimum eigenvalue of \( H \) with \( H = \{ h_{k,s} \} \) and

\[
    h_{k,s} = (2\pi)^{n/2} \int_{E^n} \left( f_k(x) - f_m(x) \right) \left( f_s(x) - f_m(x) \right) dx
\]

\( k, s = 1, 2, \ldots, m \)

The scrambling algorithm employs a procedure described on page 125 of [2].

III. Program Description and Users Guide

Three programs have been written to implement this algorithm: the proportion estimation program, a program to calculate an approximation to \( L \) and a program to scramble data prior to estimating proportions. These programs are described below and listings are provided in the appendix.

Proportion Estimation Program:

This program runs on the IBM 360/67 at LARS. Parameters are read from cards describing characteristics of the data and the statistics, the processing to be performed, and the desired outputs. The same data may be processed with several sets of statistics. The data is assumed to be sixteen channel data (from which any subset of channels may be used) residing on file 11 with one logical record per data vector. The data may be labelled or unlabelled. If labelled, the program will calculate the true proportions and print out the means of the estimates along with the associated variances and mean squared error; if unlabelled, the true proportions are read from cards and the same quantities are then computed. Due to the use of the stochastic approximation in
this algorithm, the data vectors be scrambled before being put on file 11. (Program super AM may be used for this purpose.)

The correspondence between the notation used in the preview section and the variables in the program are:

<table>
<thead>
<tr>
<th>Above</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i^j$</td>
<td>$Q(J, *)$</td>
</tr>
<tr>
<td>$n_o$</td>
<td>ISTRT</td>
</tr>
<tr>
<td>$e$</td>
<td>EPS</td>
</tr>
<tr>
<td>$T$</td>
<td>TLM</td>
</tr>
<tr>
<td>$L$</td>
<td>$L$</td>
</tr>
<tr>
<td>$G$</td>
<td>$G$</td>
</tr>
<tr>
<td>$f_i(x)$</td>
<td>$F$ (a function subprogram)</td>
</tr>
<tr>
<td>$\sum_i u_i$</td>
<td>SG</td>
</tr>
<tr>
<td></td>
<td>MU</td>
</tr>
</tbody>
</table>

The programs are set up to handle up to 16 channel data from up to 15 classes with as many as 10 different blocking factors. They can treat an unlimited number of data points. The data enters the program in "lines" which contain \( \leq 1500 \) points.

[N.B. The total number of points need not be an integral multiple of the points per line, e.g. if there are 5100 total points, we may use \( NP \) (the number of points/line = 200 and \( NL \) (the number of lines) \( \geq 26 \)]
Figure 1

Data Deck Setup for the LANDSAT Version
Figure 1 shows the set-up of the data deck necessary to execute the program. The input parameters and their formats are described below:

1) **HEADING** - Title to be printed on the output (20A4)

2) **M, MXITER, NK, INSTRT, INQ, OUTFPT, L, TLM, EPS, (K(I), I=1, NK)**

(4 I 2, 2 L 1, 3 G 10.8, 10 X, 10 I 3)

- **M** - number of classes used
- **MXITER** - number of sets of statistics to use
- **NK** - number of blocking factors to use (set = 1)
- **INSTRT** - starting value of $n_0$ in eq. (1) (default = 99)
- **INQ** - $= F$ if the initial guess for the proportion estimates ($Q's$) are to be set $= \frac{1}{M}$ (then card set (3) are not used)
  
  $= T$ if the $Q's$ are to be read in (card set (3) is required)

- **OUTPT** - $= T$ if updated $Q's$ are to be printed after each line of data. Otherwise set = F

- **L** - the $L$ value to be used in eq. (1)

- **TLM** - the maximum permissible absolute value for the update quantity for the $Q's$ (i.e.,

$$
L \cdot \sum_{i=1}^{K} \frac{f_j(x_j) - f_m(x_s)}{G(p_{i-1}, x_s)}
$$

$$
= K \times (i-1) + L
$$


EPS - minimum allowable value for a $Q$ during the estimation procedure ($10^{-2}$ seems to be a good choice)

$K(I), I=1, NK$ - the blocking factors to be used (set $K(1)=1$)

3) $CSET$ (16 L 1)

$CSET_i = \begin{cases} 
T & \text{if } i^{th} \text{ channel is to be used} \\
F & \text{otherwise}
\end{cases}$

Optional 4) $((Q(I, J), I=1, M-1), J=1, NK)$ (16 G 5. 3)

The initial guess for the $Q$'s. Used only if INQ on card 2 is = T

5) $CL(I), \left( MU(J, I), J=1, 16 \right)$ (26 X, A 1/(5 X, 5 E 15.8) for $I=1, 2, \ldots M$

$SG (J, I), J=1, 136$ (\$ X, 5 E 15.8)

These cards contain the statistics for the $M$ classes. $CL$ is the class ID, $MU$, the mean vector, and $SG$ is the covariance matrix stored in symmetric storage mode (i.e. upper triangular part stored by columns). Note that there are 33 cards required for each class. Additional sets of statistics follow card set (7).

6) $NP, NL, OUTPP, OUTPX, TRUEP$ (2 I 5, 3 L 1)

$NP$ - number of points to use per "line" ($\leq 1500$)

$NL$ - maximum number of "lines" of data
OUTPP = T the current true proportions are printed after each line (used only if TRUEP = F)

= F do not print these proportions

OUTPX = T print the data vectors

= F do not print the data vectors

TRUEP = T if the true proportions are to be read in (card set (7) is then required)

= F the class ID is associated with each data vector and the program will calculate the true proportions (card set (7) not used).

Optional 7) \( (\text{CLS}(J), \text{GT}(J), J=1,M) \)

\( (8(A2, G8.6)) \)

CLS(J) - the class ID for the \( j \)th class

GT(J) - the true proportions for the \( j \)th class

The data vectors should be on file 11 with 1 data vector per logical record in the format

CL, \((X(J), J=1,16)\)

\((8X, A1, 6X, 16F4.0, 1X)\)

where CL is the class ID (used only if TRUEP on card set 6 = F) and \( X(J) \) contains the 16 dimensional data value for a pixel.

The subroutines used in this program are briefly described below:

INSTAT - reads and prints statistics

SUBSET - for LANDSAT data (LD) version, this selects appropriate subsets of the statistics.
F - computes the value of the density function at \( X \)

TP\(\Phi\)SE - for the pseudo-random (PR) version, transposes the data matrix in situ

GEDATA - obtains or generates a line of data in the required format and order. Also computes the true proportions.

MCHLSK - computes the modified Cholesky decomposition of a covariance matrix stored in symmetric storage mode.

Program to Calculate \( L \):

This program calculates the following approximation to \( L \)

\[
L = \left( h \cdot \min \left| \sum_j h_j \right| \right)^{-1}
\]

where \( h = \min \left( \text{eval } ( H ) \right) \)

\[
H_{ks} = (2\pi)^n / 2 \int_{E^n} \left( f_k(x) - f_m(x) \right) \left( f_s(x) - f_m(x) \right) \, dx
\]

All notation is as before.

Input parameters to the program are

CSET, M, N

(16L1, 2X, 12, 2X, 12)

[ N.B. Our (limited) experience with the proportion estimation algorithm indicates that a value of \( \sim 3 \) for \( L \) appears optimal despite what this program computes. ]
Scrambling Program:

This algorithm scrambles the order of records in a data set and creates a new data set. Two storage arrays are used: one containing the integers 1, 2, ..., N where N is the total number of records and the other containing space for one data record. A temporary direct access data set, which is the same size as the original data set, is used. The algorithm is described below:

1) Set \( a_i = i \) for \( i = 1, 2, ..., N \)
2) Scramble the elements of the vector \( a \).
   (see e.g. ref. [2]).
3) For \( i = 1, 2, ..., N \)
   a) Read \( i^{th} \) record of original data set and store it in vector \( d \).
   b) Write \( d \) in \( a_i^{th} \) record in temporary data set.
4) For \( i = 1, 2, ..., N \)
   a) Read \( i^{th} \) record of temporary data set and store it in vector \( d \).
   b) Write \( d \) on \( i^{th} \) record of new data set.
5) Finished.

Note that step 1 may not be necessary if one can use the data from the temporary direct access data set.

IV. Numerical Results:

A variety of numerical experiments were conducted with this program to determine its characteristic. Both pseudo-random and LANDSAT data were used.
The most significant effect of this algorithm is due to the scrambling (i.e. the order in which the data is input). If the data is not scrambled (i.e. blocks of points from single classes appear to the program) unreliable estimates will be produced. Our experience with LANDSAT data indicates that the entire data set, whose proportions are to be estimated, needs to have the individual pixels scrambled. Various scramblings will produce different estimates with a theoretical variance of $L/N$ where $N$ is the total number of pixels.

Another effect that we noticed was that the variance of the estimate for the $M^{th}$ class was always larger than for other classes. This asymmetry, we feel, is due to the fact that the algorithm estimates proportions for the first $M-1$ classes, and the estimate for the $M^{th}$ class is then computed as $1 - \sum_{i=1}^{M-1} p_i$. By reordering the classes and then again estimating proportions, it was determined that the variance of the $M^{th}$ class would decrease from $\sim 10\%$ to $\sim 30\%$, so the effect may not be too harmful. However, the user should be aware of this and assure that the estimate for the $M^{th}$ class is of the least interest.

Detailed tests of this algorithm were run on some Hill County LANDSAT data in order to compare results with those obtained by Coberly and Odell [3] with five other proportion estimation algorithms. Table 1 shows the results obtained from the recursive maximum likelihood estimator (RMLE) for 2600 pixels of the labelled data as compared to the other five estimators. Note that the RMLE and MLE have almost equal variances and mean squared errors.
Table II shows the results obtained from the RMLE for 8400 pixels of the unlabelled data. Here again the variances and mean squared error are approximately the same as those of the MLE.

V. Conclusions:

Our experience with this algorithm indicates several important factors need be taken account of in using this algorithm: (1) all of the data needs to be scrambled point by point, (2) the class of least importance should be used as the last class, and (3) a value of $\sim 3$ for the parameter $L$ appears close to optimal.

Our tests indicate that the recursive maximum likelihood estimator (RMLE) produces results of comparable variance and accuracy as the standard maximum likelihood estimator (MLE) of ref. [3]. The amount of computation involved for the RMLE is equivalent to the first iteration of the MLE plus the scrambling of the data. Also, no additional storage is required by this algorithm to store the density functions for each data point.

Further tests of this algorithm with other LANDSAT data will be necessary to determine the effectiveness of this algorithm in the general situation.
### Table 1
Summary of Experiment 1
(Labeled Data, 2600 Pixels)

<table>
<thead>
<tr>
<th></th>
<th>CLASS</th>
<th>ODELL</th>
<th>MLE</th>
<th>RMLE</th>
<th>MIX</th>
<th>MCM</th>
<th>GT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MEAN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WH</td>
<td>.282384</td>
<td>.297041</td>
<td>.300439</td>
<td>.307807</td>
<td>.294352</td>
<td>.274749</td>
<td>.371900</td>
</tr>
<tr>
<td>FA</td>
<td>.265500</td>
<td>.274428</td>
<td>.296764</td>
<td>.277315</td>
<td>.312696</td>
<td>.235257</td>
<td>.286200</td>
</tr>
<tr>
<td>BA</td>
<td>.186089</td>
<td>.174306</td>
<td>.176688</td>
<td>.188004</td>
<td>.183300</td>
<td>.197059</td>
<td>.115400</td>
</tr>
<tr>
<td>GR</td>
<td>.101679</td>
<td>.085066</td>
<td>.085825</td>
<td>.058506</td>
<td>.074200</td>
<td>.075168</td>
<td>.079200</td>
</tr>
<tr>
<td>ST</td>
<td>.161346</td>
<td>.168155</td>
<td>.140178</td>
<td>.168364</td>
<td>.135448</td>
<td>.217765</td>
<td>.147300</td>
</tr>
<tr>
<td><strong>VAR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WH</td>
<td>.000038</td>
<td>.000174</td>
<td>.000083</td>
<td>.000094</td>
<td>.000201</td>
<td>.000459</td>
<td></td>
</tr>
<tr>
<td>FA</td>
<td>.000445</td>
<td>.001385</td>
<td>.000408</td>
<td>.000347</td>
<td>.002216</td>
<td>.003027</td>
<td></td>
</tr>
<tr>
<td>BA</td>
<td>.000031</td>
<td>.000190</td>
<td>.000084</td>
<td>.000125</td>
<td>.000123</td>
<td>.000288</td>
<td></td>
</tr>
<tr>
<td>GR</td>
<td>.000080</td>
<td>.000051</td>
<td>.000060</td>
<td>.000050</td>
<td>.000165</td>
<td>.000463</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>.000344</td>
<td>.001829</td>
<td>.000383</td>
<td>.000533</td>
<td>.002376</td>
<td>.004145</td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL VAR</strong></td>
<td>.000937</td>
<td>.003628</td>
<td>.001017</td>
<td>.001149</td>
<td>.005020</td>
<td>.008383</td>
<td></td>
</tr>
<tr>
<td><strong>MSE</strong></td>
<td>.015172</td>
<td>.013322</td>
<td>.010086</td>
<td>.010778</td>
<td>.016572</td>
<td>.032066</td>
<td></td>
</tr>
</tbody>
</table>
Table 2  
Summary of Experiment II  
(Total Data Set, 8400 Pixels)

<table>
<thead>
<tr>
<th></th>
<th>CLASS</th>
<th>ODELL</th>
<th>MLE</th>
<th>RMLE</th>
<th>MIX</th>
<th>MCM</th>
<th>GT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>WH</td>
<td>.252733</td>
<td>.262044</td>
<td>.272700</td>
<td>.267325</td>
<td>.226467</td>
<td>.084333</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>.207300</td>
<td>.183526</td>
<td>.185133</td>
<td>.197484</td>
<td>.353019</td>
<td>.023415</td>
</tr>
<tr>
<td></td>
<td>BA</td>
<td>.167633</td>
<td>.154997</td>
<td>.151200</td>
<td>.142416</td>
<td>.218460</td>
<td>.325809</td>
</tr>
<tr>
<td></td>
<td>GR</td>
<td>.186466</td>
<td>.187198</td>
<td>.189166</td>
<td>.158599</td>
<td>.158955</td>
<td>.184282</td>
</tr>
<tr>
<td></td>
<td>ST</td>
<td>.186066</td>
<td>.212231</td>
<td>.201700</td>
<td>.234173</td>
<td>.043098</td>
<td>.382159</td>
</tr>
<tr>
<td>Var</td>
<td>WH</td>
<td>.000095</td>
<td>.000263</td>
<td>.000310</td>
<td>.000307</td>
<td>.000232</td>
<td>.000949</td>
</tr>
<tr>
<td></td>
<td>FA</td>
<td>.000269</td>
<td>.000743</td>
<td>.000472</td>
<td>.000550</td>
<td>.002872</td>
<td>.002114</td>
</tr>
<tr>
<td></td>
<td>BA</td>
<td>.000063</td>
<td>.000299</td>
<td>.000206</td>
<td>.000220</td>
<td>.000135</td>
<td>.000610</td>
</tr>
<tr>
<td></td>
<td>GR</td>
<td>.000261</td>
<td>.000363</td>
<td>.000529</td>
<td>.000609</td>
<td>.001042</td>
<td>.000900</td>
</tr>
<tr>
<td></td>
<td>ST</td>
<td>.000141</td>
<td>.000969</td>
<td>.000832</td>
<td>.000903</td>
<td>.002975</td>
<td>.003402</td>
</tr>
<tr>
<td>Total Var</td>
<td>.000829</td>
<td>.017924</td>
<td>.002350</td>
<td>.002590</td>
<td>.007257</td>
<td>.007974</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td>.008622</td>
<td>.057982</td>
<td>.010273</td>
<td>.008176</td>
<td>.055378</td>
<td>.180347</td>
</tr>
</tbody>
</table>
REFERENCES


APPENDIX

FILE... DVR FORTRAN P1

REAL*4 SG(136,30), MU(16,30), Q(30,10), L,X(16,1500), DEF(30),
1 (30,10)/300*0.1/,
REAL*4 NUM(2700),
REAL*4 AT(30,6,10),
REAL*4 OS(30,10), OR(30,10)/300*0.1/, OV(30,10)/300*0.1/, D1(15,1)
REAL*4 GT(15),
REAL*4 NSF(20),
REAL*4 REND(20)
REAL*8 S, SS
INTEGER*4 K(10), P(110)/10*0/
INTEGER*4 CHAN(16),
INTEGER*2 CL(15),
LOGICAL*1 INO, IFNP, INO, OUTPT, END
LOGICAL*1 CSFT(16),
LOGICAL*1 FIXST, TRUF, /
COMMON /PASS/ SG, AN, HA
COMMON /RSET/ HT, GT
COMMON /GEPTS/ N
COMMON /ORDR, CL
COMMON /DATA/ M, NK, ISTRT, INO, OUTPT, L, TM, EPS, XMX, MXITER, CSET
MXCHN=16
NXPTS=1500
MXCLS=30

C
C M - NUMBER OF CLASSES USED (LEF, 30)
C N - NUMBER OF CHANNELS USED (LEF, 16)
C MXITER - NUMBER OF TIMES TO REDO THE RUN WITH DIFFERENT DATA
C MAXKS - NUMBER OF KS TO RE USED
C ISTRT - INITIAL VALUE OF J IN (K=1) J=0/(J=1) 1STRT (DEF=1)
C INO - LOGICAL VARIABLE INDICATING WHETHER TO READ INITIAL
C OUTPT - LOGICAL VARIABLE IF ESTIMATE OF N IS TO BE PRINTED
C L - THE L VALUE USED BY THE ALGORITHM
C TLM - LIMITING VALUE OF L,ASM(E,F,M)/G (DEF=INFINITY)
C XLS - LOWER LIMIT ALLOWED FOR THE 0'S
C XMX - UPPER LIMIT ALLOWED FOR THE 0'S
C CSET - ARRAY INDICATING WHICH OF THE 16 CHANNELS ARE TO BE USED
C K - THE BLOCKING FACTORS TO BE USED (LEF, 10 OF THEM, EACH LEF CPF
C N - THE ESTIMATES OF THE PRIORS
C SG - COVARIANCE MATRICES STORED IN SYM STORAGE MODE
C NK - MEAN VECTORS
C X - THE DATA VECTORS FOR 1 "LINE" OF DATA
P12=2.83,14159265
END=FALSE.

READ PARAMETERS
READ (5,1) REND
1 FORMAT (20A4)
WRITE (6,4) REND
4 FORMAT (20A4, 1X, 20A4, 1X)
READ (5,2) M, MXITER, NK, ISTRT, INO, OUTPT, L, TLM, EPS, XMX, (K(I), I=1, NK)
2 FORMAT (412,2L1,16G10.8,10T13)
READ (9,4) CSET
3 FORMAT (16L1)
IF (ISTRT.EQ.0) ISTRT=10
IF (TLM.LT.1.E-2) TLM=1.E70
P1M=1
P1I=0
IF (INO) GO TO 5
GO TO 10

THEN READ IN INITIAL GUESS FOR PRIORS FOR EACH BLOCKING USED.
5 READ (5,7) ((I,J), I=1, M), J=1, NK)
7 FORMAT (16G5.3)
GO TO 20

ELSE SET INITIAL GUESS FOR PRIORS ALL EQUAL
10 Y=1/M
DO 15 I=1, M
15 0.011, J=1
15 0.0(11)=Y
FILE. . .  OVR  FORTRAN P1

C 20 CONTINUE
   DO 26 I=1,M
   DO 26 J=1,NK
   26 OS(I,J)=O(I,J)
   WRITE (6,101AT)
   TL=TL-1
   WRITE (6,21) (K(I),I=1,NK)
   21 FORMAT (' K= ',10I5)
   DO 23 I=1,NK
   23 WRITE (6,22) (O(I,J),J=1,M)
   22 FORMAT (' INITIAL O=',AG16.8)
   GET STATS FOR THE CLASSES
   28 CONTINUE
   CALL INSTAT (CSET, CHAN)
   IF (IFK.EQ.0) PI2=PI2**(N/2.)
   DO 34 II=1,NK
   34 PI2(II)=0
   CALL CHOLSK (SG(1,1),N,X,DFT(I))
   25 DFT(I)=1.0/((SORT(DFT(I))*PI2)

C**********************************************************************
C FETCH ONE LINE OF DATA
C**********************************************************************

C 60 CALL C-DATA(X,MP, CHAN,N,5100)

C IF (.NOT.FIRST) GO TO 32
   FIRST=.FALSE.
   32 CONTINUE
   IFM=.FALSE.
   DO 30 I=1,NP
   IF (I.EQ.MP) IFM=.TRUE.

C**********************************************************************
C F YIELDS THE VALUE OF THE DENSITY FUNCTION FOR THE CLASS
C**********************************************************************

C  F = F(X(1,1),N,SG(1,1),MU(1,1),DFT(M))
   THE MIXTURE DISTRIBUTION IS SAMPLED IN G
   DO 35 II=1,NK
   35 G(II)=FM
   DO 40 J=1,M
   FJ=F(X(1,1),N,SG(1,1),MU(1,1),DFT(J))
   AJ(J)=FJ-FM
   AJA(J)
   DO 45 II=1,NK
   45 G(II)=G(II)+AJ(J,II)*AJ
   40 CONTINUE
   LOOP TO UPDATE PRIORS FOR EACH BLOCKING FACTOR
   DO 50 II=1,NK
   IND=.FALSE.
   KI=K(II)
   MI=M sữa(1, KI)
   GI=G(II)30
   IF (MI.EQ.0. OR. IFM) GO TO 52
   GO TO 53
   THEN PREPARE TO UPDATE II-TH PRIORS
   52 IK=MIN(MK,KI)
   IF (IK.EQ.0) IK=KI
   IND=.TRUE.
   PI2(II)=PI2(II)+IK
   53 CONTINUE
   COMPUTE UPDATED SUMS

C

ORIgINAL PAGE IS T009
OF POOR QUALITY

(ii)
FILE... FORTRAN P1

S=0.
DO 55 J=1, ,N1
XX=ABS(J)/51
T(J,J,N1)+SIGN(AMIN1(ABS(XX), TM1),XX)
IF (.INDO) GO TO 56
GO TO 55
C.
C. UPDATE THE PRIORS AND RESET
C.
54 XX=S0(J,1)+LTF1(J,1)/1111+ISTRT
T(J,1)=S
OK(J,1)=XX
S=S+XX
55 CONTINUE
IF (.NOT.INDO) GO TO 50
C.
C. RENORMALIZE THE UPDATED ESTIMATES OF THE PRIORS
C.
0N(J,J,J1)=0D-S
S=S0(J,1)
DO 58 J=1.1
IF (O(J,1)=T.E.S) S=S0(J,1)
54 CONTINUE
IF (S.GT.EPS) GO TO 64
SS=0.
DO 58 J=1.1
O(J,1)=O(J,1)+S+EPS
58 CONTINUE
IF (J(J,1)=0(J,1))/SS
56 CONTINUE
50 CONTINUE
C.
C. PRINT OUT NEW ESTIMATE OF PRIORS
C.
72 DO 70 J=1, ,N1
S=0.
DO 75 J=1,1
75 S=S0(J,1)
O(J,1)=S(J,1)-S
70 WRITE (6,170) MSF,K(J,J),O(J,1),J=1,N1
170 WRITE (6,76) (134/12, CLASS=1, A2, 0, G1, X)
IF (.NOT.INDO) GO TO 60
C.
C. UPDATE MEANS AND VARIANCES OF THE ESTIMATES FOR THIS ITERATION
C.
ITR=ITR+1
DO 116 J=1, ,N1
XX=O(J,1)
OR(J,1)=OR(J,1)+XX
OV(J,1)=OV(J,1)+XX*XX
115 O(J,1)=MSF(J)
116 CONTINUE
NSF=0.
ENDF=.FALSE.
IF (ITR.LT.MXITER) GO TO 2A
C.
C. FINISHED WITH ALL DATA, PRINT OUT ESTIMATES & STOP
C.
DO 118 J=1, ,N1
S=0.
DO 119 J=1,1
MSF(J)=OV(J,1)-2.010(J,1)-OR(J,1))/MXITER+GT(J,1)*GT(J,1)
S=S+MSF(J)
119 XX=OR(J,1)/ITER
OR(J,1)=XX
OV(J,1)/ITER-XX
118 CONTINUE
WRITE (6,131) XX(J,J),MSF(J),J=1,N1
131 FORMAT (1(K'=12,15.1, MEANS AND SD'S OF THE ESTIMATES & THE MSE/ 1 (A3,G16.8, 1, G16.8, G16.8, G16.8))
S=S1/N
WRITE (6,121) S1

ORIINAL PAGE 3S
OF POOR QUALITY.
CONTINUE
121 FORMAT (10*** MEAN MSE*,G16.8)
STOP
CCC
FINISHED WITH ALL DATA FOR THIS ITERATION
100 END=TRUE.
GO TO 72
END

FUNCTION :=(X,N,L,MU,DET)
CCC
COMPUTE THE VALUE OF THE DENSITY FUNCTION AT X
ccc
REAL*4 X(1),L(1),MU(1),Y(16)
REAL*8 TF,S

SOLVE L Y=X-MU WHERE L IS THE CHOLESKY DECOMP OF COVAR MATRIX.
DIAG ELEMENTS OF L ARE STORED AS RECIPROCALS.
S=X(1)-MU(1)
Y(1)=S
TF=S*S*L(1)
IF (N.LE.1) GO TO 15
K=1

LOOP TO COMPUTE Y(I)'S
DO 10 I=2,N
S=X(1)-MU(I)
JJ=I-I
DO 20 J=1,JJ
K=K+1
20 S=S-L(K)*Y(J)
K=K+1
Y(I)=S
TF=TF+S*S*L(K)
10 CONTINUE
CONTINUE
15 IF (TF.LT.325.) GO TO 17
F=0.
RETURN
17 F=EXP(SQRT(-TF/2.))*DET
RETURN
END
SUBROUTINE INSTAT (CSLET, CHAN)
REAL*4 SG(136, 30), MU(16, 30)
INTEGER*4 CHAN(I)
INTEGER*2 CL(15), NC(15), NP(15)
LOGICAL*1 CSLET(16)
COMMON /PASS/ SG, MU, M, N
COMMON /WORD/ CL
DO 5 I=1, M
READ (5, 1) CL(I), (MU(J, I), J=1, 16)
1 FORMAT (26X, A1/(5X, 5F15.8))
5 READ (5, 4) (SG(J, I), J=1, 136)
4 FORMAT (5X, SF15.8)
CALL CSLET(CSLET, CHAN)
DO 10 I=1, M
WRITE (6, 2) CL(I), NC(I), NP(I), (MU(J, I), J=1, N)
2 FORMAT (/ 'CLASS', A2, ' ', NO. OF PTS=', I5, ' MEAN', 1DF15.4/)
1 I=I+1
WRITE (6, 3)
3 FORMAT (/ 'COVARIANCE' /)
J=1
J=J+1
DO 20 J=1, I
J=J+1
20 WRITE (6, 2) (SG(L, I), L=J1, J2)
21 FORMAT (/ 'CLASS', A2, ' ', NO. OF PTS=', I5, ' MEAN', 1DF15.4/)
10 CONTINUE
RETURN
END

SUBROUTINE CSLET (CSLET, CHAN)
LOGICAL*1 CSLET(I)
REAL*4 SG(136, 30), MU(16, 30)
INTEGER*4 CHAN(I)
COMMON /PASS/ SG, MU, M, K

FINI CHANNELS DESIRED
K=0
DO 10 I=1, 16
IF (I .NE. CSLET(I)) GO TO 10
K=K+1
10 CONTINUE

SELECT APPROPRIATE SUBSETS OF SG & MU
J=0
DO 20 I=1, K
20 J=J+1

STORE DIAGONAL ELEMENTS
J=J+1
DO 25 L=1, M
SG(J, L)=SG(IDAG(CHAN(I)), L)
25 MU(I, L)=MU(CHAN(I), L)
IF (1 .EQ. K) RETURN

MOVE ALL ELEMENTS OF NEXT ROW EXCEPT THE DIAGONAL ONE
L=CHAN(I+1)
DO 30 J=1, L-1
J=J+1
30 SG(J, L)=SG(ISR(L, L), L)
20 CONTINUE
STOP
END

(v)

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE GFDATA (X, NP, CHAN, KK, * )
REAL 4 SG (136,30), MU(16,30)
REAL 4 X (16,1500), GT(15)
INTEGER 4 CHAN(15)
INTEGER 2 C, PTS(15)/1500/, IFR(0), SCLS(15)
INTEGER 2 LPTS(15)/1500/, IFR(15)/1500/
LOGICAL 1 FSTT, TRUEF,
LOGICAL 1 OUTPP, OUTPX,
LOGICAL 1 TRUEP
FOR CURFURY'S DATA
COMMON /PASS/ SG, MU,M
COMMON /RSET/ LINE, GT
COMMON /ORDR/ SCLS
COMMON /GRPTS/ ML
IF (.NOT.FSTT) GO TO 10
FSTT = .FALSE.
LINE = 0
&
& NP = NUMBER OF POINTS PER 'LINE' (.LE. 1500)
NL = NUMBER OF LINES
OUTPP = PRINT RUNNING TRUE PROPORTIONS
OUTPX = PRINT DATA VECTORS
TRUEP = READ IN TRUE PROPORTIONS
READ (5,3) (CLS(J), GT(J), J=1, M)
1 FORMAT (215, ALL)
NP = NP + 1
IF (LINE.LE.NL) GO TO 20
FINISHED WITH THIS PASS OF THE DATA
REWIND 11
IFR = IFR+1
IF (IFR.GT.11) RETURN 1
COMPUTE TRUE PROPORTIONS & REARRANGE CLASSES TO THOSE IN STATS
IF (TRUEP) K = M
JJ = 0
DO 50 I = 1, K
IF (TRUEP) PTS(I) = 0
DO 55 J = 1, K
IF (CLS(J).EQ.SCLS(I)) GO TO 52
55 CONTINUE
WRITE (6,53) SCLS(I)
53 FORMAT (' CLASS NOT FOUND', A3)
GO TO 50
52 IF (I.EQ.J) GO TO 50
L = CLS(I)
CLS(I) = CLS(J)
CLS(J) = L
IF (TRUEP) GO TO 50
L = PTS(I)
PTS(I) = PTS(J)
PTS(J) = L
50 CONTINUE
IF (TRUEP) GO TO 57
DO 54 I = 1, K
JJ = JJ + PTS(I)
54 CONTINUE
PRINT OUT PROPORTIONS
XJ = JJ
WRITE (6,51)
51 FORMAT (4F10.5)
ORIGINIAL PAGE IS OF POOR QUALITY
51 FORMAT ('1 DATA OBSERVED/1 CLASS',T10,'POINTS',T20,'PROPORTIONS')
DO 60 J=1,K
IF (TRUEP) GO TO 50
X=PTS(J)/X
Y=CLS(J)/X
ST(J)=X*Y
60 WRITE (6,61) CLS(J),PTS(J),GT(J)
61 FORMAT (A3,T10,16,T20,516.3)
RETURN

C ........................................................................
20 CONTINUE
DO 30 I=1,IP
C
C READ OBSERVATION VECTOR
C PLAN (11,31,F5=35,FRR=35) CL, (X(J),J=1,16)
C
31 FORMAT (A8,6A,16F4.0)
SELECT SUBSET OF CHANNELS DESIRED
DO 37 L=1,XX
37 Y(L,J)=X(CHAN(L),J)
IF (TRUEP) GO TO 30
IF (ITER.GE.1) GO TO 30
IF (I,J.EQ.0) GO TO 42
C
C TALLY FOR COMPUTING TRUE PROPORTIONS
DO 40 J=1,K
40 CONTINUE
K=K+1
C
CLS(K)=CL
PTS(J)=PTS(J)+1
30 CONTINUE
IF (ITER.GE.1) RETURN
46 CONTINUE
IF (.NOT.,OUTPP OR .TRUEP) GO TO 87
C
C COMPUTE PROPORTIONS OF DATA FOR THIS LINE AND TO DATE
C
L=0
LT=0
DO 80 J=1,K
L=L+PTS(J)
LPTS(J)=PTS(J)/L
ipts(J)=PTS(J)/L
80 L=L+PTS(J)
XX=L
DO 85 J=1,K
PROP(J)=PTS(J)/XX
PROP(J)=LPTS(J)/FLAT(LT)
85 WRITE (A,66) LINE,CLS(J),PROP,TPR
85 FORMAT ('LINE',15,'CLASS',15,'TRUE PROPORTIONS FOR THIS LINE =
15,16F4.0,'TOTAL TRUE PROP. TO DATE =',15,16F4.0)
C
C WRITE OUT OBSERVATION VECTORS.
C
87 CONTINUE
IF (.NOT.,OUTPP) RETURN
WRITE (A,66)
WRITE (A,66)
65 FORMAT ('1',X=31)
DO 70 I=1,IP
L=1+(LINE-1)*IP
70 WRITE (6,62) L,X(J,J),J=1,XX
62 FORMAT (I5,16F4.0)
RETURN

35 WRITE (A,36) LINE,
36 FORMAT ('1 END OF DATA ON LINE',15,' AND PIXEL',15)
RETURN
MP=1
LINE=NL
IF (LJ.LT.LT.) GO TO 70
IF (.TRUEP) GO TO 64
RETURN
END
SUBROUTINE MCHLSK(KK, NV, NUM, DFT)

*****************************************************************************

THIS ROUTINE COMPUTES THE MODIFIED CHOLESKY DECOMPOSITION OF
THE COVARIANCE MATRIX. THE DECOMPOSITIONS OVERLAY THE ELEMENTS
OF THE COVARIANCE MATRIX.

KK = L * L

*****************************************************************************

KK - THE COVARIANCE MATRIX STORED IN SYMMETRIC STORAGE MODE.
NV - THE NUMBER OF CHANNELS USED
NUM - A WORK AREA OF SIZE NV * NV.
DFT - THE DETERMINANT OF THE COVARIANCE MATRIX.

REAL KK(1:NV, 1:NV)
LOGICAL JF
JF = TRUE.
J1 = 0
J2 = 0
DFT = 1.

LOOP OVER ALL CHANNELS.
DO 10 J = 1, NV
  KL = J - 1
  L = J + 1
  J0 = J - 1
  J1 = J + 1
  TF = KK(J, J)
  IF (JF) GO TO 12
  K1 = 0
  CONTINUE
  COMPUTE THE DIAGONAL ELEMENTS OF K AND STORE IN KK.
  TEMPORARILY STORE THE PRODUCT KK(J, 1) = KK(J, 1) IN NUM(J).
  DO 15 J = 1, KL
    R = KK(J, J + 1)
    K1 = K1 + 1
    R1 = KK(K1, R)
    TF = TF - R1 * R
    NUM(J) = 1
    15 CONTINUE
    KK(J, 1) = TF
  12 CONTINUE
  DFT = DFT * TF
  IF (L .GT. NV) GO TO 10
  JRD = J - 1 + 1
  CONTINUE
  DO 20 J = 1, NV
    JRD = JRD + J - 1
    T1 = KK(JRD, J)
    IF (JF) GO TO 16
    DO 25 J = 1, KL
      T1 = T1 - NUM(I) * KK(JRD + 1)
      25 CONTINUE
      KK(JRD + J) = T1 / TF
    20 CONTINUE
    JF = FALSE.
  10 CONTINUE
  J1 = 0
  CONTINUE
  STORE THE ELEMENTS OF K IN THIS FORM FOR USE IN SUBROUTINE
  CLASS.
  DO 30 J = 1, NV
    J1 = J1 + 1
  30 KK(J1) = 1 / KK(J1)
  RETURN
END

ORIGINAL PAGE IS OF POOR QUALITY.
This program scrambles NP records (.LE.10000) each of length IY and puts the results on file 12

C
C GENERATE THE INT(1)S 1,2,......NP AND STORE IN ARRAY INT
C
ISN 0007 DC LO I=1, NP
ISN 0008 20 INT(I)=I
C
ISN 0009 J=NP
C
ISN 0010 25 R=GGUBF(ISEED)
C
ISN 0011 IT=INT(J)
ISN 0012 INT(I)=INT(I)
ISN 0013 INT(I)=IT
ISN 0014 J=J+1
ISN 0015 IF (J.GT.1) GO TO 23
ISN 0016 WRITE (6,7)
ISN 0017 FORMAT (SHFL*)
C
ISN 0018 LCP TO PUT 1-TH RECORD IN INT(1)-TH POSITION ON FILE 3 (A TEMP.
C
ISN 0019 DIRECT ACCESS FILE)
C
ISN 0020 DC 30 I=1, NP
ISN 0021 L=INT(I)
ISN 0022 FIND(B*L)
ISN 0023 READ (11,1) DAT
ISN 0024 1 FORMAT (2I3,2A1,10I1)
ISN 0025 WRITE (B*L) DAT
ISN 0026 WRITE (6,8)
ISN 0027 FORMAT (SHFL *)
C
ISN 0028 COPY FILE 2 TO FILE 12
C
ISN 0029 DC 40 I=1, NP
ISN 0030 11=1+1
ISN 0031 READ (B*11) DAT
ISN 0032 FIND(B*11)
ISN 0033 IF (11.GT.3) WRITE (6,3) I
ISN 0034 3 FORMAT (1d)
ISN 0035 WRITE (12,1) DAT
ISN 0036 STOP
ISN 0037 END

(ix)