

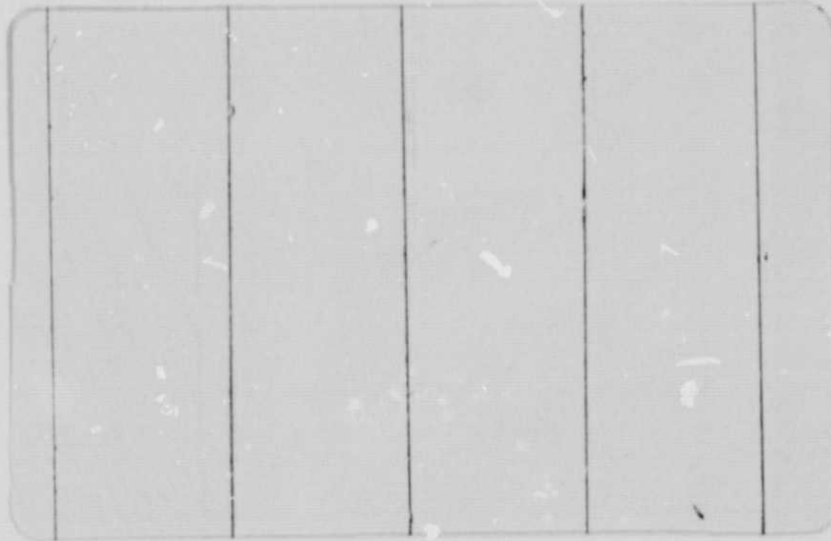
## General Disclaimer

### 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.

NASA CR-

147824



(NASA-CR-147824) THE RECURSIVE MAXIMUM  
LIKELIHOOD PROPORTION ESTIMATOR: JSEE'S  
GUIDE AND TEST RESULTS (Rice Univ.) 27 p.  
HC \$4.00 CSCI 09E

N76-27904

Unclas  
G3/61 44566



# ICSA

INSTITUTE FOR COMPUTER SERVICES AND APPLICATIONS

RICE UNIVERSITY

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  $\{x\}$  from  $M$  normally distributed multivariate pattern classes  $H^j$ ,  $j=1, 2, \dots, M$  the  $M-1$  dimensional recursive maximum likelihood proportion estimate (RMLPE)<sup>(1)</sup>  $p^i$  at the  $i^{\text{th}}$  data vector is given by

$$p^i = p^{i-1} + \frac{1}{i} L \left[ g(p^{i-1}, x_i) \right]^{-1} \\ \left( f_1(x_i) - f_M(x_i), f_2(x_i) - f_M(x_i), \dots, f_{M-1}(x_i) - f_M(x_i) \right) \quad (1)$$

where  $f_j(x)$  is the density function for the  $j^{\text{th}}$  class;

$$f_j(x) = (2\pi)^{-n/2} |K_j|^{-\frac{1}{2}} \exp \left[ (x-u_j)^T K_j^{-1} (x-u_j) \right] \quad (2)$$

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

$$g(p^{i-1}, x_i) = f_M(x_i) + \sum_{\ell=1}^{m-1} p_{\ell}^{i-1} (f_{\ell}(x_i) - f_M(x_i)) \quad (3)$$

and  $L$  is a suitably chosen constant in this approximation. The proportion estimate for the  $M^{\text{th}}$  class is denoted by  $p_m^i$  and given by

$$p_m^i = 1 - \sum_{\ell=1}^{m-1} p_{\ell}^i \quad (4)$$

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  $p_j^i \geq 0$  and  $\sum_{\ell=1}^m p_{\ell}^i = 1$ ; and (3) introducing an additional damping term in

the update term of eq.(1). The final form of the algorithm is

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

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

$$\text{LMT}(a, b) = \tilde{b}$$

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

NORM is the renormalizing function defined by

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

where

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

$$\tilde{y}_i = y_i$$

If

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

$$\tilde{y}_i \leftarrow \tilde{y}_i - \min_i(\tilde{y}_i) + \epsilon \quad i = 1, 2, \dots, M$$

$$\tilde{y}_i \leftarrow \tilde{y}_i / \sum_{\ell=1}^m \tilde{y}_\ell \quad ;$$

and  $n_0$  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  $L$

$$L = \left( u \cdot \min |K_j|^{\frac{1}{2}} \right)^{-1}$$

where  $u$  is the minimum eigenvalue of  $H$  with  $H = \{h_{ks}\}$   
and

$$h_{ks} = (2\pi)^{n/2} \int_{E^n} (f_k(x) - f_m(x)) (f_s(x) - f_m(x)) dx$$

$$k, s = 1, 2, \dots, m$$

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 should 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:

<u>Above</u>	<u>Program</u>
$P_i^j$	Q(J, *)
$n_o$	ISTR T
$\epsilon$	EPS
T	TLM
L	L
G	G
$f_i(x)$	F (a function subprogram)
$\sum_i$	SG
$u_i$	MU

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)  $\geq 26$  ]



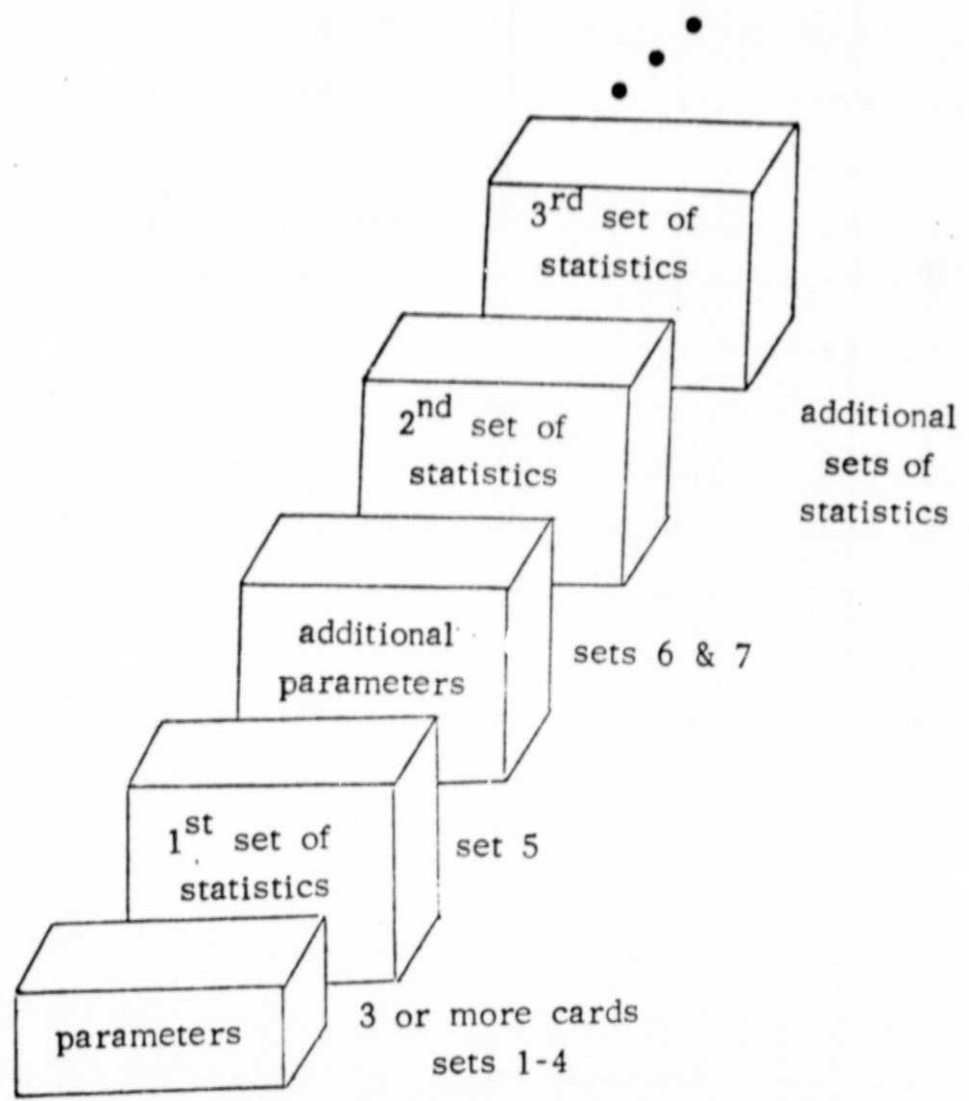


Figure 1  
Data Deck Setup for the LANDSAT Version

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) HEDNG - Title to be printed on the output (20A4)
- 2) M, MXITER, NK, ISTRT, INQ, ØUTPT, 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)
- ISTRT - 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)
- ØUTPT - = 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.

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

$$s = K * (i-1) + \ell$$

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

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)  $CL(I), (MU(J, I), J=1, 16)$   
(26 X, A 1/(5 X, 5 E 15.8)  
(SG (J, I), J=1, 136)  
(8 X, 5 E 15.8) } for  $I=1, 2, \dots, M$

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, ØUTPP, ØUTPX, TRUEP  
(2 I 5, 3 L 1)

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

NL - maximum number of "lines" of data

$\emptyset$ UTPP - = T the current true proportions are  
 printed after each line (used only if  
 TRUEP = F)  
 = F do not print these proportions  
 $\emptyset$ UTPX - = 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) (CLS(J), GT(J), J=1, M )  
 (8(A 2, G 8.6) )

CLS(J) - the class ID for the  $J^{\text{th}}$  class

GT(J) - the true proportions for the  $J^{\text{th}}$  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)

(8 X, A 1, 6 X, 16 F 4.0, 1 X)

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
- TPØSE - 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

$$= \left( h \cdot \min_j \left| \sum_j \right|^{\frac{1}{2}} \right)^{-1}$$

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

$$\begin{aligned} \& H_{ks} = (2\pi)^{n/2} \int_{E^n} \left( f_k(x) - f_m(x) \right) \cdot \\ & \cdot \left( f_s(x) - f_m(x) \right) dx \\ & k, s, = 1, 2, \dots, m \end{aligned}$$

All notation is as before.

Input parameters to the program are

CSET, M, N

(16L1, 2X, I2, 2X, I2)

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

Note that step 4 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  $M^{\text{th}}$  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  $M^{\text{th}}$  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  $M^{\text{th}}$  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  $M^{\text{th}}$  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 I  
 (Labeled Data, 2600 Pixels)

	CLASS	ODELL	MLE	RMLE	MIX	MCM	GT
MEAN	WH	.297041	.300439	.307807	.294352	.274749	.371900
	FA	.274428	.296764	.277315	.312696	.235257	.286200
	BA	.174306	.176688	.188004	.183300	.197059	.115400
	GR	.085066	.085825	.058506	.074200	.075168	.079200
	ST	.168155	.140178	.168364	.135448	.217765	.147300
VAR	WH	.000174	.000083	.000094	.000201	.000459	
	FA	.001385	.000408	.000347	.002216	.003027	
	BA	.000190	.000084	.000125	.000123	.000288	
	GR	.000051	.000060	.000050	.000165	.000463	
	ST	.001829	.000383	.000533	.002376	.004145	
TOTAL VAR		.003528	.001017	.001149	.005020	.008383	
MSE		.013322	.010086	.010778	.016572	.032066	

PRECEDING PAGE BLANK NOT FILMED

Table 2  
 Summary of Experiment II  
 (Total Data Set, 8400 Pixels)

	CLASS	ODELL	MLE	RMLE	MIX	MCM	GT
MEAN	WH	.262044	.272700	.267325	.226467	.084333	.294000*
	FA	.183526	.185133	.197484	.353019	.023415	.249000
	BA	.154997	.151200	.142416	.218460	.325809	.124000
	GR	.187198	.189166	.158599	.158955	.184282	.138000
	ST	.212231	.201700	.234173	.043098	.382159	.196000
VAR	WH	.000253	.000310	.000307	.000232	.000949	
	FA	.007430	.000472	.000550	.002872	.002114	
	BA	.000299	.000206	.000220	.000135	.000610	
	GR	.000363	.000529	.000609	.001042	.000900	
	ST	.009569	.000832	.000903	.002975	.003402	
TOTAL VAR		.017924	.002350	.002590	.007257	.007974	
MSE		.057982	.010273	.008176	.055378	.180347	

## REFERENCES

- [1] D. Kazakos, "Recursive Estimation of Prior Probabilities Using the Mixture Approach," ICSA Technical Report #275-025-019, Rice University, Houston, Texas, September, 1974.
- [2] D. Knuth, Seminumerical Algorithms, the Art of Computer Programming, Vol. 2, Addison-Wesley, Reading, Mass., 1969, p.125.
- [3] W.A. Coberly and P.L. Odell, "An Empirical Comparison of Five Proportion Estimators," from the annual report of the University of Texas at Dallas for NASA contract NAS 9-13512, January, 1975.

APPENDIX

FILE . . .	DVR	FORTRAN P1	
------------	-----	------------	--

	REAL*4 SG(136,30),MU(16,30),O(30,10),L,X(16,1500),DET(30),	DVR00010
	1 T(30,10)/300*0./	DVR00020
	REAL*4 DUM(2700)	DVR00030
	REAL*4 A(30),G(10)	DVR00040
	REAL*4 OS(30,10),OB(30,10)/300*0./,OV(30,10)/300*0./	DVR00050
	REAL*4 GT(15)	DVR00060
	REAL*4 MSE(30)	DVR00070
	REAL*4 HEDNG(20)	DVR00080
	REAL*8 S,SS	DVR00090
	INTEGER*4 K(10),IP1(10)/10*0/	DVR00100
	INTEGER*4 CHAN(16)	DVR00110
	INTEGER*2 CL(15)	DVR00120
	LOGICAL*1 INQ,INP,IND,OUTPT,FND	DVR00130
	LOGICAL*1 CSET(16)	DVR00140
	LOGICAL*1 FRST/.TRUE./	DVR00150
	COMMON /PASS/ SG,MU,M,N	DVR00160
	COMMON /RSFT/ NSET,GT	DVR00170
	COMMON /GEPTS/ NL	DVR00180
	COMMON /ORDR/ CL	DVR00190
	NAHLIST /IDAT/ M, NK,ISTRT,IND,OUTPT,L,TLM,FPS,XX,XXITER,CSET	DVR00200
	MXCHN=16	DVR00210
	MXPTS=1500	DVR00220
	MXCLS=30	DVR00230
C		DVR00240
C	M - NUMBER OF CLASSES USED (.LE.30)	DVR00250
C	N - NUMBER OF CHANNELS USED (.LE.16)	DVR00260
C	MXITER - NUMBER OF TIMES TO REDO THE RUN WITH DIFFERENT DATA	DVR00270
C	NK - NUMBER OF K'S TO BE USED	DVR00280
C	ISTRT - INITIAL VALUE OF J IN O(R+1)=O(R)-1/(J+R)*L*I. (DEF=10)	DVR00290
C	IND - LOGICAL VARIABLE INDICATING WHETHER TO READ INITIAL	DVR00300
C	GUESS FOR THE O'S OR NOT	DVR00310
C	OUTPT - LOGICAL VAR =T IF ESTIMATE OF O IS TO BE PRINTED	DVR00320
C	AFTER EACH LINE OF DATA	DVR00330
C	L - THE L VALUE USED BY THE ALGORITHM	DVR00340
C	TLM - LIMITING VALUE OF L*ABS(EJ-EM)/GI (DEF=INFINITY)	DVR00350
C	FPS - LOWER LIMIT ALLOWED FOR THE O'S	DVR00360
C	XX - UPPER LIMIT ALLOWED FOR THE O'S	DVR00370
C	CSET - ARRAY INDICATING WHICH OF THE 16 CHANNELS ARE TO BE USED	DVR00380
C	K - THE BLOCKING FACTORS TO BE USED (.LE. 10 OF THEM,FACH LE NPO)	DVR00390
C	O - THE ESTIMATES OF THE PRIORS	DVR00400
C	SG - COVARIANCE MATRICES STORED IN SYM STORAGE MODE	DVR00410
C	MU - MEAN VECTORS	DVR00420
C	X - THE DATA VECTORS FOR 1 'LINE' OF DATA	DVR00430
C		DVR00440
C	PI2=2.*3.14159265	DVR00450
C	END=.FALSE.	DVR00460
C		DVR00470
C	READ PARAMETERS	DVR00480
C		DVR00490
C	READ (5,1) HEDNG	DVR00500
C	1 FORMAT (20A4)	DVR00510
C	WRITE (6,4) HEDNG	DVR00520
C	4 FORMAT (//1X,20A4,///)	DVR00530
C	READ (5,2) M,XXITER,NK,ISTRT,IND,OUTPT,L,TLM,FPS,XX,(K(I),I=1,NK)	DVR00540
C	2 FORMAT (4I2,2L1,4G10.8,10I3)	DVR00550
C	READ (5,3) CSET	DVR00560
C	3 FORMAT (16I1)	DVR00570
C	IF (ISTRT.EQ.0) ISTRT=10	DVR00580
C	IF (TLM.LT.1.E-2) TLM=1.E70	DVR00590
C	M1=M-1	DVR00600
C	ITER=0	DVR00610
C	IF (IND) GO TO 5	DVR00620
C	GO TO 10	DVR00630
C		DVR00640
C	THEN READ IN INITAAL GUESS FOR PRIORS FOR EACH BLOCKING USED.	DVR00650
C		DVR00660
C	5 READ (5,7) ((O(I,J),I=1,M1),J=1,NK)	DVR00670
C	7 FORMAT (16G5.3)	DVR00680
C	GO TO 20	DVR00690
C		DVR00700
C	ELSE SET INITIAL GUESS FOR PRIORS ALL EQUAL	DVR00710
C		DVR00720
C	10 Y=1./M	DVR00730
C	DO 15 I=1,M1	DVR00740
C	DO 15 J=1,NK	DVR00750
C	15 O(I,J)=Y	DVR00760
C		DVR00770
C	C*****	DVR00780

ORIGINAL PAGE IS  
OF POOR QUALITY

C			DVR00790
	20	CONTINUE	DVR00800
		DO 26 I=1,M	DVR00810
		DO 26 J=1,NK	DVR00820
	26	OS(I,J)=O(I,J)	DVR00830
		WRITE (6,1DAT)	DVR00840
		TLM=TLM/L	DVR00850
		WRITE (6,21) (K(I),I=1,NK)	DVR00860
	21	FORMAT (' K=',10I5)	DVR00870
		DO 23 I=1,NK	DVR00880
	23	WRITE (6,22) (O(J,I),J=1,M1)	DVR00890
	22	FORMAT (' INITIAL O=',8G16.8)	DVR00900
C		GET STATS FOR THE CLASSES	DVR00910
C			DVR00920
	28	CONTINUE	DVR00930
		CALL INSTAT (CSET, CHAN)	DVR00940
		IF (IFR.FO.O) PI2=PI2**(N/2.)	DVR00950
		DO 34 II=1,NK	DVR00960
	34	IP1(II)=0	DVR00970
C			DVR00980
C		DO CHOLFSKY DECOMP OF THE COVARIANCES	DVR00990
C			DVR01000
		DO 25 I=1,M	DVR01010
		CALL BCHLSK (SG(1,I),N,X,DET(I))	DVR01020
	25	DET(I)=1.DO/(SORT(DET(I))*PI2)	DVR01030
C		*****	DVR01040
C			DVR01050
C		FETCH ONE LINE OF DATA	DVR01060
C			DVR01070
C		*****	DVR01080
	60	CALL CADATA(X,MP, CHAN,N,8100)	DVR01090
C			DVR01100
		IF (.NOT.FIRST) GO TO 32	DVR01110
		FIRST=.FALSE.	DVR01120
C			DVR01130
C		LOOP OVER ALL DATA POINTS TO UPDATE ESTIMATE OF PRIORS	DVR01140
C			DVR01150
	32	CONTINUE	DVR01160
		IENP=.FALSE.	DVR01170
		DO 30 I=1,MP	DVR01180
		IF (I.EQ.MP) IENP=.TRUE.	DVR01190
C			DVR01200
C		F YIELDS THE VALUE OF THE DENSITY FUNCTION FOR THE CLASS	DVR01210
C			DVR01220
		FM=F(X(1,I),N,SG(1,M),MU(1,M),DET(M))	DVR01230
C			DVR01240
C		THE MIXTURE DISTRIBUTION IS SCORED IN G	DVR01250
C			DVR01260
		DO 35 II=1,NK	DVR01270
	35	G(II)=FM	DVR01280
		DO 40 J=1,M1	DVR01290
		FJ=F(X(1,I),N,SG(1,J),MU(1,J),DET(J))	DVR01300
		A(J)=FJ-FM	DVR01310
		AJ=A(J)	DVR01320
		DO 45 II=1,NK	DVR01330
	45	G(II)=G(II)+O(J,II)*AJ	DVR01340
	40	CONTINUE	DVR01350
C			DVR01360
C		LOOP TO UPDATE PRIORS FOR EACH BLOCKING FACTOR	DVR01370
C			DVR01380
		DO 50 II=1,NK	DVR01390
		IND=.FALSE.	DVR01400
		KI=K(II)	DVR01410
		MIK=MOD(I,KI)	DVR01420
		GI=G(II)	DVR01430
		IF (MIK.FO.O.OR.IENP) GO TO 52	DVR01440
		GO TO 53	DVR01450
C			DVR01460
C		THEN PREPARE TO UPDATE II-TH PRIORS	DVR01470
C			DVR01480
	52	IK=MINO(MIK,KI)	DVR01490
		IF (IK.EQ.O) IK=KI	DVR01500
		IND=.TRUE.	DVR01510
		IP1(II)=IP1(II)+IK	DVR01520
	53	CONTINUE	DVR01530
C			DVR01540
C		COMPUTE UPDATED SUMS	DVR01550
C			DVR01560

ORIGINAL PAGE IS  
OF POOR QUALITY

```

S=0.
DO 55 J=1,M1
XX=A(J)/G1
T(J,II)=T(J,II)+SIGN(AMIN1(ABS(XX),TLM),XX)
IF (.NOT.IND) GO TO 56
GO TO 55
C
C
C
UPDATE THE PRIORS AND RESET
56 XX=O(J,II)+L*T(J,II)/(IP1(II) +ISTRT)
T(J,II)=0.
O(J,II)=XX
S=S+XX
55 CONTINUE
IF (.NOT.IND) GO TO 50
C
C
C
RENORMALIZE THE UPDATED ESTIMATES OF THE PRIORS
O(M,II)=1./DO-S
S=O(1,II)
DO 54 J=2,M
IF (O(J,II).LT.S) S=O(J,II)
54 CONTINUE
IF (S.GT.EPS) GO TO 64
SS=0.
DO 58 J=1,M
O(J,II)=O(J,II)-S+EPS
58 SS=SS+O(J,II)
DO 62 J=1,M1
62 O(J,II)=O(J,II)/SS
64 CONTINUE
50 CONTINUE
30 CONTINUE
C*****
IF (.NOT.OUTPUT) GO TO 60
C
C
C
PRINT OUT NEW ESTIMATE OF PRIORS
72 DO 70 II=1,NK
S=0.
DO 75 J=1,M1
S=S+O(J,II)
O(M,II)=1./DO-S
70 WRITE (6,76) NSET,K(II),(CL(J),O(J,II),J=1,M)
76 FORMAT ('/' UPDATED ESTIMATE OF THE PRIORS FOR LINE ',I5,' WITH RL
CKING FACTOR K=',I3/4(' CLASS=',A2,' O=',G15.8,3X))
IF (.NOT.END) GO TO 60
C
C
C
UPDATE MEANS AND VARIANCES OF THE ESTIMATES FOR THIS ITERATION.
ITER=ITER+1
DO 116 II=1,NK
DO 115 J=1,M
XX=O(J,II)
OB(J,II)=OB(J,II)+XX
OV(J,II)=OV(J,II)+XX*XX
115 O(J,II)=OS(J,II)
116 CONTINUE
NSET=0
END=.FALSE.
IF (ITER.LT.MXITER) GO TO 28
C
C
C
FINISHED WITH ALL DATA. PRINT OUT ESTIMATES & STOP
DO 118 II=1,NK
S1=0.
DO 119 J=1,M
MSE(J)=(OV(J,II)-2.*GT(J)*OB(J,II))/MXITER+GT(J)*GT(J)
S1=S1+MSE(J)
XX=OB(J,II)/ITER
OB(J,II)=XX
XX=OV(J,II)/ITER-XX*XX
119 OV(J,II)=SORT(XX)
WRITE (6,131) K(II),(CL(J),OB(J,II),OV(J,II),MSE(J),J=1,M)
131 FORMAT (' K=',I5,' MEANS AND SD'S OF THE ESTIMATES & THE MSE'/
1 (A3,G16.8,' +- ',G16.8,G18.8))
S1=S1/M
WRITE (6,121) S1
DVR01570
DVR01580
DVR01590
DVR01600
DVR01610
DVR01620
DVR01630
DVR01640
DVR01650
DVR01660
DVR01670
DVR01680
DVR01690
DVR01700
DVR01710
DVR01720
DVR01730
DVR01740
DVR01750
DVR01760
DVR01770
DVR01780
DVR01790
DVR01800
DVR01810
DVR01820
DVR01830
DVR01840
DVR01850
DVR01860
DVR01870
DVR01880
DVR01890
DVR01900
DVR01910
DVR01920
DVR01930
DVR01940
DVR01950
DVR01960
DVR01970
DVR01980
DVR01990
DVR02000
DVR02010
DVR02020
DVR02030
DVR02040
DVR02050
DVR02060
DVR02070
DVR02080
DVR02090
DVR02100
DVR02110
DVR02120
DVR02130
DVR02140
DVR02150
DVR02160
DVR02170
DVR02180
DVR02190
DVR02200
DVR02210
DVR02220
DVR02230
DVR02240
DVR02250
DVR02260
DVR02270
DVR02280
DVR02290
DVR02300
DVR02310
DVR02320
DVR02330
DVR02340

```

(iii)

FILE. . . DVR FORTRAN P1

	118 CONTINUE	DVR02350
	121 FORMAT (' *** MEAN MSE',G16.8)	DVR02360
	STOP	DVR02370
C		DVR02380
C	FINISHED WITH ALL DATA FOR THIS ITERATION	DVR02390
C		DVR02400
	100 END=.TRUE.	DVR02410
	GO TO 72	DVR02420
	END	DVR02430

	FUNCTION F(X,N,L,MU,DET)	DVR02440
C		DVR02450
C	COMPUTE THE VALUE OF THE DENSITY FUNCTION AT X	DVR02460
	REAL*4 X(1),L(1),MU(1),Y(16)	DVR02470
	REAL*8 TF,S	DVR02480
C		DVR02490
C	SOLVE L Y=X-MU WHERE L IS THE CHOLESKY DECOMP OF COVAR MATRIX.	DVR02500
C	DIAG ELEMENTS OF L ARE STORED AS RECIPROCAL.	DVR02510
	S=X(1)-MU(1)	DVR02520
	Y(1)=S	DVR02530
	TF=S*S*L(1)	DVR02540
	IF (N.EQ.1) GO TO 15	DVR02550
	K=1	DVR02560
C		DVR02570
C	LOOP TO COMPUTE Y(I'S)	DVR02580
		DVR02590
		DVR02600
	DO 10 I=2,N	DVR02610
	S=X(I)-MU(I)	DVR02620
	JJ=I-1	DVR02630
	DO 20 J=1,JJ	DVR02640
	K=K+1	DVR02650
20	S=S-L(K)*Y(J)	DVR02660
	K=K+1	DVR02670
	Y(I)=S	DVR02680
	TF=TF+S*S*L(K)	DVR02690
10	CONTINUE	DVR02700
15	CONTINUE	DVR02710
	IF (TF.LT.325.) GO TO 17	DVR02720
	F=0.	DVR02730
	RETURN	DVR02740
17	F=EXP(SNGL(-TF/2.))*DET	DVR02750
	RETURN	DVR02760
	END	DVR02770
		DVR02780

ORIGINAL PAGE IS  
OF POOR QUALITY

```

SUBROUTINE INSTAT (CSET, CHAN)
REAL*4 SG(136,30),MU(16,30)
INTEGER*4 CHAN(1)
INTEGER*2 CL(15),NC(15),NP(15)
LOGICAL*1 CSET(16)
COMMON /PASS/ SG,MU,M,N
COMMON /ORDR/ 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,5F15.8)
  CALL SUBSET(CSET,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,I5,' NO. OF PTS=',I5/' MEAN',10F11.4/
1  1X,6F11.4)
    WRITE (6,3)
3  FORMAT (' COVARIANCE')
    J1=1
    J2=0
    DO 20 J=1,N
      J2=J2+J
      J1=J1+J-1
20  WRITE (6,21) (SG(L,I),L=J1,J2)
21  FORMAT (/ (1X,13F10.4))
10  CONTINUE
  RETURN
END

```

DVR02790  
DVR02800  
DVR02810  
DVR02820  
DVR02830  
DVR02840  
DVR02850  
DVR02860  
DVR02870  
DVR02880  
DVR02890  
DVR02900  
DVR02910  
DVR02920  
DVR02930  
DVR02940  
DVR02950  
DVR02960  
DVR02970  
DVR02980  
DVR02990  
DVR03000  
DVR03010  
DVR03020  
DVR03030  
DVR03040  
DVR03050  
DVR03060  
DVR03070

```

SUBROUTINE SUBSET (CSET, CHAN)
LOGICAL*1 CSET(1)
REAL*4 SG(136,30),MU(16,30)
INTEGER*4 CHAN(1)
COMMON /PASS/ SG,MU,M,K
ISB(I,J)=(I*(I-1))/2+J
IDAG(I)=(I*(I+1))/2
FIND CHANNELS DESIRED
K=0
DO 10 I=1,16
  IF (.NOT.CSET(I)) GO TO 10
  K=K+1
  CHAN(K)=I
10 CONTINUE
SELECT APPROPRIATE SUBSETS OF SG & MU
JJ=0
DO 20 I=1,K
  STORE DIAGONAL ELEMENTS
  JJ=JJ+1
  DO 25 L=1,N
    SG(JJ,L)=SG(IDAG(CHAN(I)),L)
25  MU(I,L)=MU(CHAN(I),L)
    IF (I.EQ.K) RETURN
MOVE ALL ELEMENTS OF NEXT ROW EXCDPT THE DIAGONAL ONE
L1=CHAN(I+1)
DO 30 J=1,I
  L2=CHAN(J)
  JJ=JJ+1
  DO 30 L=1,N
30  SG(JJ,L)=SG(ISB(L1,L2),L)
20  CONTINUE
STOP
END

```

DVR03080  
DVR03090  
DVR03100  
DVR03110  
DVR03120  
DVR03130  
DVR03140  
DVR03150  
DVR03160  
DVR03170  
DVR03180  
DVR03190  
DVR03200  
DVR03210  
DVR03220  
DVR03230  
DVR03240  
DVR03250  
DVR03260  
DVR03270  
DVR03280  
DVR03290  
DVR03300  
DVR03310  
DVR03320  
DVR03330  
DVR03340  
DVR03350  
DVR03360  
DVR03370  
DVR03380  
DVR03390  
DVR03400  
DVR03410  
DVR03420  
DVR03430  
DVR03440  
DVR03450  
DVR03460  
DVR03470  
DVR03480

(v)

ORIGINAL PAGE IS  
OF POOR QUALITY



```

SUBROUTINE GEDATA (X, NP, CHAN, KK, *)
REAL*4 SG(136,30), MU(16,30)
REAL*4 X(16,1500), GT(15)
INTEGER*4 CHAN(1)
INTEGER*2 CL, CLS(15), PTS(15)/15*0/, ITER/0/, SCLS(15)
INTEGER*2 LPTS(15)/15*0/, IPTS(15)/15*0/
LOGICAL*1 FRST/.TRUE./
LOGICAL*1 OUTPP, OUTPX
LOGICAL*1 TRUEP

```

```
FOR CUBERLY'S DATA
```

```

COMMON /PASS/ SG, MU, M
COMMON /RSFT/ LINE, GT
COMMON /OROR/ SCLS
COMMON /GEPTS/ NL
IF (.NOT. FRST) GO TO 10
FRST=.FALSE.
LINE=0
K=0

```

```

NP = NUMBER OF POINTS PER 'LINE' (.LE.1500)
NL = NUMBER OF LINES
OUTPP=T - PRINT RUNNING TRUE PROPORTIONS
OUTPX=T - PRINT DATA VECTORS
TRUEP=T - READ IN TRUE PROPORTIONS

```

```

READ (5,1) NP, NL, OUTPP, OUTPX, TRUEP
1 FORMAT (2I5, 3L1)
NPS=NP
WRITE (6,2) NP, NL, OUTPP, OUTPX, TRUEP
2 FORMAT (' NP=', I5, ' NL=', I5, ' OUTPP=', L1, ' OUTPX=', L1, ' TRUEP=',
1 L1)
IF (TRUEP) READ (5,3) (CLS(J), GT(J), J=1, M)
3 FORMAT (8(A2, G8.6))
10 LINE=LINE+1
NP=NPS
IF (LINE.LE.NL) GO TO 20

```

```
FINISHED WITH THIS PASS OF THE DATA
```

```

REWIND 11
ITER=ITER+1
IF (ITER.GT.1) RETURN 1

```

```
COMPUTE TRUE PROPORTIONS & REARRANGE CLASSES TO HHOSE 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
```

```

57 CONTINUE
XJ=JJ
WRITE (6,51)

```

DVR03490  
DVR03500  
DVR03510  
DVR03520  
DVR03530  
DVR03540  
DVR03550  
DVR03560  
DVR03570  
DVR03580  
DVR03590  
DVR03600  
DVR03610  
DVR03620  
DVR03630  
DVR03640  
DVR03650  
DVR03660  
DVR03670  
DVR03680  
DVR03690  
DVR03700  
DVR03710  
DVR03720  
DVR03730  
DVR03740  
DVR03750  
DVR03760  
DVR03770  
DVR03780  
DVR03790  
DVR03800  
DVR03810  
DVR03820  
DVR03830  
DVR03840  
DVR03850  
DVR03860  
DVR03870  
DVR03880  
DVR03890  
DVR03900  
DVR03910  
DVR03920  
DVR03930  
DVR03940  
DVR03950  
DVR03960  
DVR03970  
DVR03980  
DVR03990  
DVR04000  
DVR04010  
DVR04020  
DVR04030  
DVR04040  
DVR04050  
DVR04060  
DVR04070  
DVR04080  
DVR04090  
DVR04100  
DVR04110  
DVR04120  
DVR04130  
DVR04140  
DVR04150  
DVR04160  
DVR04170  
DVR04180  
DVR04190  
DVR04200  
DVR04210  
DVR04220  
DVR04230  
DVR04240

ORIGINAL PAGE IS  
OF POOR QUALITY

```

51 FORMAT ('1 DATA OBSERVED'/' CLASS',T10,'POINTS',T20,'PROPORTIONS') DVR04250
   DO 60 J=1,K DVR04260
   IF (.TRUE.P) GO TO 60 DVR04270
   XC=PTS(J)/XJ DVR04280
   GT(J)=XC DVR04290
60 WRITE (6,61) CLS(J),PTS(J),GT(J) DVR04300
61 FORMAT (A3,T10,I6,T20,G16.8) DVR04310
   RETURN 1 DVR04320
C ***** DVR04330
20 CONTINUE DVR04340
   DO 30 I=1,MP DVR04350
C DVR04360
C READ OBSERVATION VECTOR DVR04370
C DVR04380
C READ (11,3),FND=35,FRR=35) CL, (X(J,I),J=1,16) DVR04390
C DVR04400
31 FORMAT (8X,A1,6X,16F4.0) DVR04410
   SELECT SUBSET OF CHANNELS DESIRED DVR04420
C DVR04430
C DVR04440
   DO 37 L=1,KK DVR04450
37 Y(L,I)=X(CHAN(L),I) DVR04460
   IF (.TRUE.P) GO TO 30 DVR04470
   IF (.ITER.GE.1) GO TO 30 DVR04480
   IF (K.EQ.0) GO TO 42 DVR04490
C DVR04500
C TALLY FOR COMPUTING TRUE PROPORTIONS DVR04510
   DO 40 J=1,K DVR04520
   IF (CL.EQ.CLS(J)) GO TO 45 DVR04530
40 CONTINUE DVR04540
42 K=K+1 DVR04550
   J=K DVR04560
   CLS(K)=CL DVR04570
45 PTS(J)=PTS(J)+1 DVR04580
30 CONTINUE DVR04590
   IF (.ITER.GE.1) RETURN DVR04600
64 CONTINUE DVR04610
   IF (.NOT.(OUTPP.OR.TRUE.P)) GO TO 87 DVR04620
C DVR04630
C COMPUTE PROPORTIONS OF DATA FOR THIS LINE AND TO DATE DVR04640
C DVR04650
   L=0 DVR04660
   LT=0 DVR04670
   DO 80 J=1,K DVR04680
   LT=LT+PTS(J) DVR04690
   LPTS(J)=PTS(J)-1PTS(J) DVR04700
   IPTS(J)=PTS(J) DVR04710
80 L=L+LPTS(J) DVR04720
   XX=L DVR04730
   DO 85 J=1,K DVR04740
   PROP=LPTS(J)/XX DVR04750
   TPROP=PTS(J)/FLOAT(LT) DVR04760
85 WRITE (6,86) LINE,CLS(J),PROP,TPROP DVR04770
85 FORMAT (' LINE',I5,' CLASS ',A1,' TRUE PROPORTIONS FOR THIS LINE = DVR04780
   ' ,G16.8,' TOTAL TRUE PROPOR. TO DATE =',G16.8) DVR04790
C DVR04800
C WRITE OUT OBSERVATION VECTORS DVR04810
C DVR04820
87 CONTINUE DVR04830
   IF (.NOT.(OUTPX)) RETURN DVR04840
   WRITE (6,66) DVR04850
66 FORMAT ('1 X=') DVR04860
   DO 70 I=1,MP DVR04870
   L=I+(LINE-1)*MP DVR04880
70 WRITE (6,62) L,(X(J,I),J=1,KK) DVR04890
62 FORMAT (I5,16F5.0) DVR04900
   RETURN DVR04910
35 WRITE (6,36) LINE,I DVR04920
36 FORMAT ('1 END OF DATA ON LINE',I5,' AND PIXEL',I5) DVR04930
   REWIND 11 DVR04940
   MP=I-1 DVR04950
   LINE=NL DVR04960
   IF (MP.LE.0) GO TO 10 DVR04970
   IF (.ITER.LT.1) GO TO 64 DVR04980
   RETURN DVR04990
   END DVR05000

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

SUBROUTINE MCHLSK(KK,NV,DUM,DET)
*****
THIS ROUTINE COMPUTES THE MODIFIED CHOLESKY DECOMPOSITION OF
THE COVARIANCE MATRIX. THE DECOMPOSITIONS OVERLAY THE ELEMENTS
OF THE COVARIANCE MATRIX.
KK=L D L*
*****
KK - THE COVARIANCE MATRIX STORED IN SYMMETRIC STORAGE MODE.
NV - THE NUMBER OF CHANNELS USED
DUM - A WORK AREA OF SIZE NV-1
DET - THE DETERMINANT OF THE COVARIANCE MATRIX.

REAL KK(1),DUM(1)
LOGICAL*1 JF1
JF1=.TRUE.
J1=0
J0=0
DET=1.

      LOOP OVER ALL CHANNELS

DO 10 J=1,NV
KL=J-1
L=J+1
J0=J1
J1=J1+J
TF=KK(J1)
IF (JF1) GO TO 12
KL=0

      COMPUTE THE DIAGONAL ELEMENTS OF D AND STORE IN KK
      TEMPORARILY STORE THE PRODUCT KK(I,I)*KK(J,I) IN DUM(I)

DO 15 I=1,KL
R=KK(J0+I)
K1=K1+I
R1=KK(K1)*R
TF=TF-R1*R
DUM(I)=R1
15 CONTINUE
KK(J1)=TF
12 CONTINUE
DET=DET*TF
IF (L.GT.NV) GO TO 10
IRD=J1-L+1

      COMPUTE THE R,J-TH ELEMENT OF L USING T1

DO 20 IR=L,NV
IRD=IRD+IR-1
T1=KK(IRD+J)
IF (JF1) GO TO 16
DO 25 I=1,KL
T1=T1-DUM(I)*KK(IRD+I)
25 CONTINUE
16 KK(IRD+J)=T1/TF
20 CONTINUE
JF1=.FALSE.
10 CONTINUE
J1=0

      STORE THE ELEMENTS OF D IN THIS FORM FOR USE IN SUBROUTINE
      CLASS

DO 30 J=1,NV
J1=J1+J
30 KK(J1)= 1./KK(J1)
RETURN
END

```

DVR05010  
DVR05020  
DVR05030  
DVR05040  
DVR05050  
DVR05060  
DVR05070  
DVR05080  
DVR05090  
DVR05100  
DVR05110  
DVR05120  
DVR05130  
DVR05140  
DVR05150  
DVR05160  
DVR05170  
DVR05180  
DVR05190  
DVR05200  
DVR05210  
DVR05220  
DVR05230  
DVR05240  
DVR05250  
DVR05260  
DVR05270  
DVR05280  
DVR05290  
DVR05300  
DVR05310  
DVR05320  
DVR05330  
DVR05340  
DVR05350  
DVR05360  
DVR05370  
DVR05380  
DVR05390  
DVR05400  
DVR05410  
DVR05420  
DVR05430  
DVR05440  
DVR05450  
DVR05460  
DVR05470  
DVR05480  
DVR05490  
DVR05500  
DVR05510  
DVR05520  
DVR05530  
DVR05540  
DVR05550  
DVR05560  
DVR05570  
DVR05580  
DVR05590  
DVR05600  
DVR05610  
DVR05620  
DVR05630  
DVR05640  
DVR05650  
DVR05660  
DVR05670  
DVR05680  
DVR05690  
DVR05700  
DVR05710  
DVR05720  
DVR05730

ORIGINAL PAGE IS  
OF POOR QUALITY

```

COMPILER OPTIONS - NAME= MAIN,OPT=CQ,LINECNT=60,SIZE=CCCCCK,
SOURCE,LCUDIC,NCLIST,NOLCK,LCAD,MAP,NREDIT,NOID,XREF
C
C THIS PROGRAM SCRAMBLES NP RECORDS (.LE.10000) EACH OF LENGTH 19
C *CRDS CONTAINED ON FILE 11 AND PUTS THE RESULTS ON FILE 12
C
ISN 0C02' INTEGER*2 INT(10000),DAT(19)
ISN 0C03 DEFINE FILE B(10000,10,U,IJ)
ISN 0C04 ISEED=1314159793
ISN 0C05 READ (5,11) NP
ISN 0C06 11 FORMAT (15)
C
C GENERATE THE INTEGERS 1,2,...,NP AND STORE IN ARRAY INT
C
ISN 0C07 DC 40 I=1,NP
ISN 0C08 20 INT(I)=I
C
C SCRAMBLE ARRAY INT
C
ISN 0C09 J=NP
C
C GGUBF GENERATES A RANDOM NUMBER FROM U(0,1)
C
ISN 0C10 25 R=GGUBF(ISEED)
ISN 0C11 IY=J*NR+1
ISN 0C12 I=INT(J)
ISN 0C13 INT(I)=INT(IX)
ISN 0C14 INT(IX)=I
ISN 0C15 J=J-1
ISN 0C16 IF (J.GT.1) GO TO 25
ISN 0C17 *WRITE (6,7)
ISN 0C18 7 FORMAT (: SHFLD')
ISN 0C19
C
C LOOP TO PUT I-TH RECORD IN INT(I)-TH POSITION ON FILE 3 (A TEMP.
C DIRECT ACCESS FILE)
C
ISN 0C20 DC 30 I=1,NP
ISN 0C21 L=INT(I)
ISN 0C22 FIND(B,L)
ISN 0C23 READ (11,1) DAT
ISN 0C24 1 FORMAT (2I3,2A1,10I3)
ISN 0C25 30 *WRITE (8,L) DAT
ISN 0C26 *WRITE (5,8)
ISN 0C27 8 FORMAT (: ON 8')
C
C COPY FILE 8 TO FILE 12
C
ISN 0C28 DC 40 I=1,NP
ISN 0C29 I1=I+1
ISN 0C30 READ (8,I1) DAT
ISN 0C31 FIND(B,I1)
ISN 0C32 IF (1/50*50.EQ.1) *WRITE (6,3) I
ISN 0C33 3 FORMAT (I8)
ISN 0C34 40 *WRITE (12,I) DAT
ISN 0C35 *STOP
ISN 0C36 END
ISN 0C37

```

ORIGINAL PAGE IS  
OF POOR QUALITY