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)

Lockheed Electronics Company, Inc.
A SUBSIDIARY OF LOCKHEED CORPORATION
1830 NASA Road 1, Houston, Texas 77058
Tel. 713-333-5411

JSC-14621
APR 5 1979
CR160162

Ref: 642-6919
Contract NAS 9-15800
Job Order 73-705
End Item PO 89947/SCPRV-3396

TECHNICAL MEMORANDUM

MATHEMATICAL DESCRIPTION AND PROGRAM DOCUMENTATION FOR CLASSY, AN ADAPTIVE MAXIMUM LIKELIHOOD CLUSTERING METHOD

BY

R. K. Lennington
and
M. E. Rassbach
Elogic, Inc.
Houston, Texas

Approved By: J. C. Minter
T. C. Minter, Supervisor
Techniques Development
Section

(NASA-CR-160162) MATHEMATICAL DESCRIPTION AND PROGRAM DOCUMENTATION FOR CLASSY, AN ADAPTIVE MAXIMUM LIKELIHOOD CLUSTERING METHOD (Lockheed Electronics Co.) 63 p HC A04/MF A01 CSCL 09B G3/61
N79-21802
Unclas
24526

RECEIVED
NASA STI FACILITY
ACCESS DEPT.
MAY 1979

April 1979

LEC-12177

CONTENTS

Section		Page
1.	INTRODUCTORY DESCRIPTION	1-1
	1.1 ASSUMPTIONS AND PROBLEM DEFINITION	1-1
	1.2 SOLUTION PROCEDURE	1-2
2.	MATHEMATICAL BACKGROUND	2-1
	2.1 FUNDAMENTAL EQUATIONS FOR CONTINUOUS STATISTICS	2-1
	2.2 MODE OF ITERATION	2-2
	2.2.1 UPDATING MODE: (INDEX(KL) > 0)	2-2
	2.2.2 ITERATING MODE: (INDEX(KL) < 0)	2-3
	2.3 TREE STRUCTURE	2-4
	2.4 EQUATIONS FOR THE PROPORTION CALCULATION	2-6
	2.5 LINEAR ALGEBRA AND GENERAL LINEAR TRANSFORMATIONS	2-11
	2.6 EQUATIONS FOR A MIXTURE OF TWO DISTRIBUTIONS	2-16
3.	DESCRIPTION OF SUBROUTINES	3-1
	3.1 STATIS	3-2
	3.2 ADJUST	3-6
	3.3 JOIN	3-11
	3.4 SPLIT	3-12
	3.5 DENCAL	3-24
	3.6 APRIOR	3-25
	3.7 ISPLIT	3-26
	3.8 EIGROT	3-26

Section Page

4. CONCLUSIONS. 4-1

4.1 TIMING AND OPTIMIZATION 4-1

4.2 MODIFICATIONS AND IMPROVEMENTS. 4-3

5. REFERENCE. 5-1

TABLES

Table		Page
I	MOMENTS OF THE MIXTURE OF TWO NORMAL DISTRIBUTIONS.	2-18
II	CLUSTER TREE TRANSFORMATION ROUTINES.	3-10
III	VARIABLES AND ACCUMULATORS USED IN SPLIT.	3-20
IV	INITIALIZATION VARIABLES USED IN SPLIT.	3-23

FIGURE

Figure		Page
1	Flow diagram for CLASSY algorithm	1-8

1. INTRODUCTORY DESCRIPTION

1.1 ASSUMPTIONS AND PROBLEM DEFINITION

The fundamental mathematical assumption underlying CLASSY is that the data may be usefully approximated by a mixture of multivariate normal densities. That is, if p is probability and x is an observation vector,

$$p(x|m,\pi) = \sum_{i=1}^{m} a_i p_i(x|\mu_i,\Sigma_i) \tag{1}$$

where

a_i = the *a priori* probability of occurrence of class i

$p_i(x|\mu_i,\Sigma_i)$ = the multivariate normal probability density function for class i

m = the total number of classes

π = the full set of parameters
= $\{a_1,\cdots,a_m, \mu_1,\cdots,\mu_m, \Sigma_1,\cdots,\Sigma_m\}$

Given a set of unlabeled sample vectors $\{x_j\}$, we may form the likelihood function in the following manner.

$$L(\{x_j\}|m,\pi) = \prod_{j=1}^{N} \left[\sum_{i=1}^{m} a_i p_i(x_j|\mu_i,\Sigma_i) \right] \tag{2}$$

where N = the total number of samples.

So far, the assumptions and equations parallel the usual maximum-likelihood development. CLASSY makes the additional assumption that each value of the parameters m and π occurs with an *a priori* probability $A(m,\pi)$. The objective of CLASSY is to determine the discrete parameter m and the continuous parameter vector π to maximize the following function.

$$L(\{x_j\},m,\pi) = A(m,\pi) \prod_{j=1}^{N} \left[\sum_{i=1}^{m} a_i p_i(x_j|\mu_i,\Sigma_i) \right] \tag{3}$$

$A(m,\pi_m)$ must be chosen so that it satisfies the normalization constraint

$$\sum_{m=1}^{\infty} \int A(m,\pi)d\pi = 1 \tag{4}$$

Typically, in the absence of other information, the *a priori* probabilities may be chosen as

$$A(m,\pi) = \prod_{i=1}^{m} C_i \tag{5}$$

where C_i = a constant containing normalization factors over π. With this choice for $A(m,\pi)$, the function to be maximized becomes

$$L(\{x_j\},m,\pi) = \left(\prod_{i=1}^{m} C_i \right) \prod_{j=1}^{N} \left\{ \sum_{i=1}^{m} \frac{a_i}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{(x_j - \mu_i)^T \Sigma_i^{-1} (x_j - \mu_i)}{2} \right] \right\} \tag{6}$$

where d = dimensionality of the samples.

1.2 SOLUTION PROCEDURE

Many approaches may be taken in maximizing eq. (4). The approach chosen in CLASSY is to interleave the maximum-likelihood iteration (designed to maximize $L(\{x_j\},m,\pi)$ with respect to the continuous parameter vector π) with a discrete split, join, and combine process (designed to maximize $L(\{x_j\},m,\pi)$ with respect to the discrete parameter m). It is expected that by alternating these two techniques, values of m and π corresponding to at least a local

maximum of $L(\{x_j\},m,\pi)$ will be determined. Since the splitting and combining techniques operate around each existing cluster, only in special cases will this local maximum fail to be global.

The data are first scrambled to ensure that a true random sample is obtained. This is especially important in the CLASSY algorithm since any correlation in the data may cause the maximum-likelihood procedure to converge very slowly or experience cyclic drifts. The initial values assumed are

$$\left.\begin{aligned} m &= 1 \\ a_1 &= 1 \\ \mu_1 &= \begin{bmatrix} 0.04 \\ \vdots \\ 0.04 \end{bmatrix} \\ \Sigma_1 &= \begin{bmatrix} 10 & & 0 \\ & \ddots & \\ 0 & & 10 \end{bmatrix} \end{aligned}\right\} \tag{7}$$

The data are then examined point by point, and the parameter vector π is updated according to the maximum likelihood equations which may be expressed as follows (ref. 1).

$$p(i|x_k,\pi) = \frac{a_i p_i(x_k|\mu_i,\Sigma_i)}{\sum_{i=1}^{m} a_i p_i(x_k|\mu_i,\Sigma_i)} \tag{8}$$

$$a_i' = \frac{1}{N}\sum_{k=1}^{N} p(i|x_k,\pi) \tag{9}$$

$$\mu_i' = \frac{\sum_{k=1}^{N} p(i|x_k,\pi)x_k}{\sum_{k=1}^{N} p(i|x_k,\pi)} \tag{10}$$

$$\Sigma_i' = \frac{\sum_{k=1}^{N} p(i|x_k,\pi)(x_k - \mu)(x_k - \mu)^T}{\sum_{k=1}^{N} p(i|x_k,\pi)} \tag{11}$$

where $p(i|x_k,\pi)$ is the posterior probability of class i, given the *kth* sample vector and value of the parameters, and the primes refer to new or updated values for the parameters.

This same technique is applied to the accumulation of the third- and fourth-order moments and the logarithm likelihood for each cluster. These statistics are used to test the fit of the hypothesized distribution to the data.

As each point is considered, the probability that it belongs to each class is computed. These probabilities, which may be thought of as the fractional part of each data point assigned to each cluster, are accumulated as the "weights" for each cluster (eq. (8)). When the weight for a given cluster exceeds a threshold value (which increases each time it is exceeded), the program checks the likelihood ratio and the fit of the normal distribution to the data for that cluster. Old data (accumulated using less accurate parameter values) is also subtracted from each parameter sum accumulation at this time.

The fit of the hypothesized normal distribution to the data for a cluster is evaluated by examining the third- and fourth-order moments about the mean for that cluster, which represent measures of skewness and kurtosis. The statistics which are generated are given by

$$S_1 = (S^T\Sigma^{-1}S) \tag{12}$$

where

S = the skewness vector (trace of the rank 3 skewness tensor using the inverse covariance)

S_1 = a scalar measure of skewness

S^T = transpose of S

Σ^{-1} = the inverse covariance matrix

$$K_1 = Tr(K\Sigma^{-1}) \tag{13}$$

where

K = matrix of kurtosis values (trace of the rank 4 kurtosis tensor)

$K_1, K_2, \cdots$ = scalar measures of kurtosis

$$K_2 = Tr(K\Sigma^{-1}K\Sigma^{-1}) - \frac{1}{d}[Tr(\Sigma^{-1}K)]^2 \tag{14}$$

In CLASSY, these three statistics are tested against their approximate sampling distributions computed under the hypothesis that the samples were drawn from the normal distribution specified by the current values of the parameters. If any one of these three statistics exceeds the threshold value, the hypothesis is generated that the cluster may be split into two parts. The parameters for each of the two new component clusters are estimated by minimizing the difference between the observed covariance matrix, the skewness vector, and the kurtosis matrix, and the corresponding quantities for the mixture distribution composed of the two new normal distributions.

Following the generation of a split hypothesis, the parent cluster is not discarded immediately. When the maximum-likelihood iteration cycle is begun again, it is carried out for the previously existing clusters, including the parent cluster and the new subclusters (with the new parameters and an initial weight, which is currently set to 40 points for each cluster). Thus, a hierarchial structure or cluster tree evolves as this process is repeated.

At the same time in the processing that a cluster is checked to see if it may need to be split, certain other tests are performed. If a cluster has subclusters (i.e., has been previously split), it is not split again; but the likelihood ratio of the daughter clusters to the parent cluster is examined. If this ratio is larger than a given threshold, the parent cluster is eliminated and the daughter clusters take its place (i.e., the hypothesis that the parent is split is accepted). On the other hand, if the ratio is too small, the daughter clusters are eliminated in favor of the parent (i.e., the hypothesis that the parent is split is rejected). In addition, a cluster may be eliminated if its prior probability becomes too small, which may occur if another cluster has "taken" all its points. The program also checks the degree of overlap between clusters at the same level in the cluster tree. If the degree of overlap is too great and the two clusters are not the only subclusters of a given parent cluster, the hypothesis that these are the same cluster is raised by joining the two similar clusters. The new cluster is given parameters which are a combination of the clusters joined to form it. All of these are tests restructuring the cluster tree at certain intervals; namely, when the weight (or number of points assigned to a given cluster on a fractional probabilistic basis) has accumulated to a certain point in the statistics accumulation portion of the program.

After tests have been made to determine if a cluster may need to be split or if the cluster tree may need to be restructured, the old data are subtracted from the cluster statistics previously accumulated, and the skewness vector and the kurtosis matrix for that cluster are reset to zero. The program then continues the statistical accumulation. If a complete pass through the data set is made before a cluster is tested for possible adjustment, the values of the means at that time are used in eq. (11) until another pass through the data set has been completed; that is, the program does full iteration for the cluster rather than continuous updating.

The present program cycles through the data a fixed number of time, as controlled by an external parameter. When the desired number of passes is complete, the program classifies the data by going through them point by point

and assigning each data point to the cluster in the cluster tree for which the probability of occurrence of this data point is the greatest. This is the only time in the program that points are assigned to clusters. When all of the points have been assigned, a cluster map showing the cluster symbol for each point is printed out. The program also prints out the final values for the parameters for each cluster in the cluster tree. A general flow diagram for the CLASSY program is shown in figure 1.

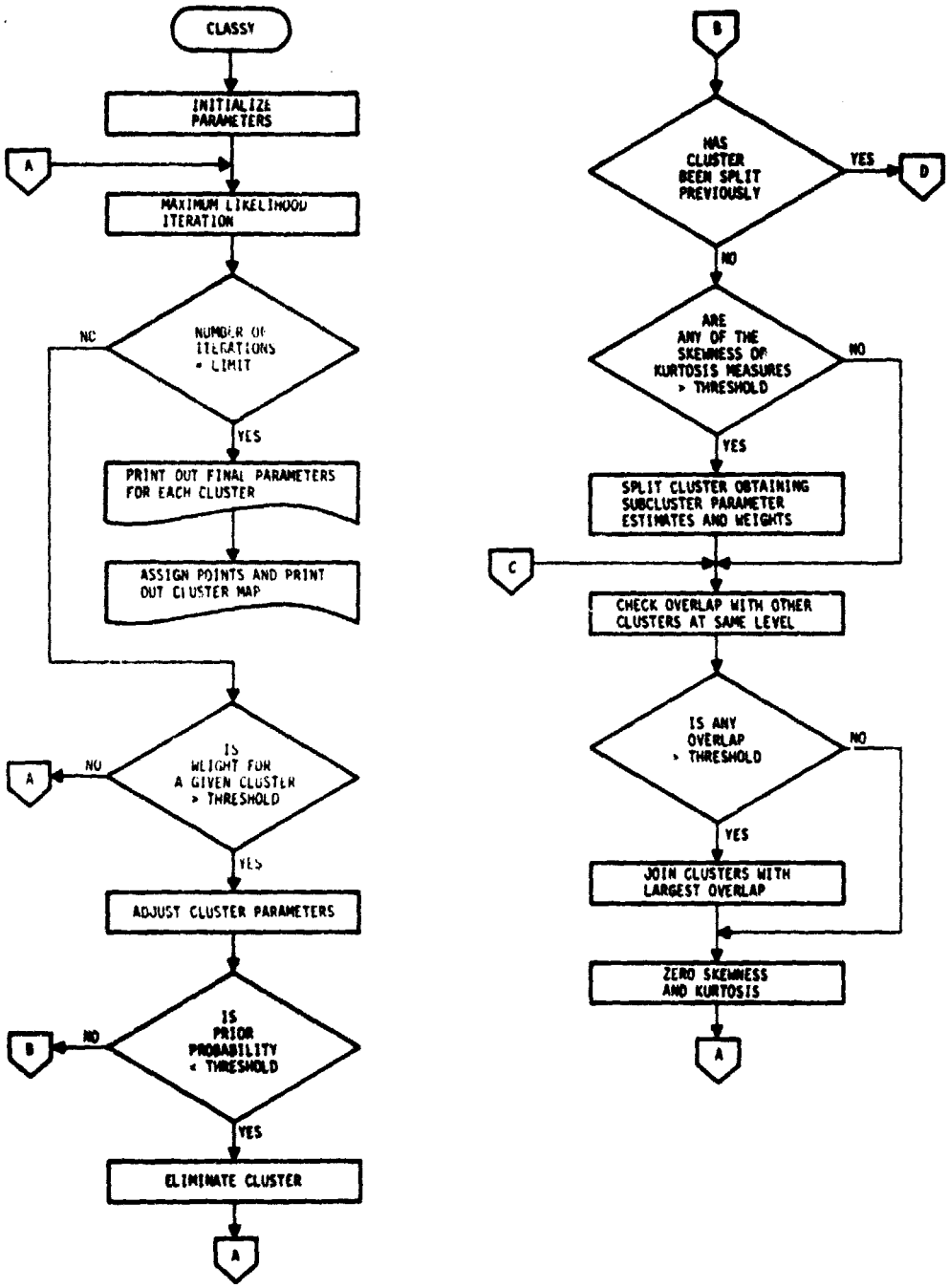

Figure 1.– Flow diagram for CLASSY algorithm.

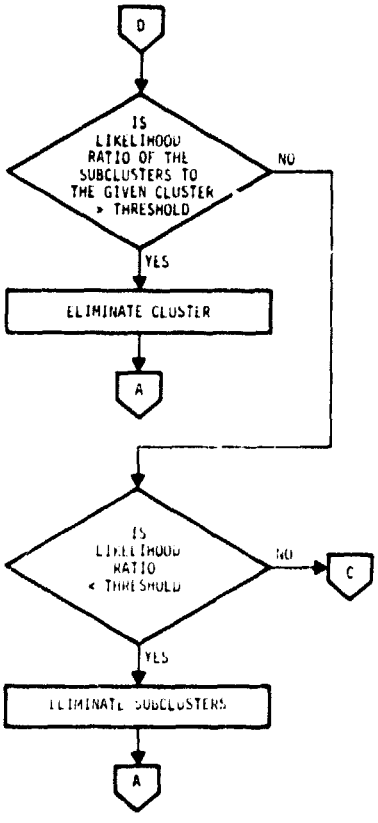

Figure 1.- Concluded.

2. MATHEMATICAL BACKGROUND

Descriptions of the detailed mathematics and statistics used in CLASSY are given in this section, noting variables which have a direct correspondence to theoretical quantities. In addition, preliminary remarks establishing equations used in more than one routine are included.

2.1 FUNDAMENTAL EQUATIONS FOR CONTINUOUS STATISTICS

The fundamental equations for the continuous statistical parameters follow:

$$\hat{a}_i = \frac{1}{N} \sum_{j=1}^{N} P(i|x_j) \tag{15}$$

$$\hat{\mu}_i = \frac{1}{a_i N} \sum_{j=1}^{N} x_j \, P(i|x_j) \tag{16}$$

$$\hat{\Sigma}_i = \frac{1}{a_i N} \sum_{j=1}^{N} x_j x_j^T \, P(i|x_j) - \mu_i \mu_i^T \tag{17}$$

where

$$P(i|x_j) = \frac{a_i p_i(x_j)}{\sum_{k=1}^{m} a_n p_n(x_j)} \tag{18}$$

where

N = total number of points

m = number of classes

a_i = proportion

μ_i = mean

Σ_i = covariance

and values marked with a circumflex (^) denote estimates.

These equations are the standard maximum-likelihood equations for a mixture of normal distributions with unknown parameters and proportions. (CLASSY actually solves the equations with the large-sample size approximation by substituting $\hat{\mu}$ for μ in eq. (17).)

There are obviously many numerical procedures for solving these equations, given a specific set of data points x_j. CLASSY uses direct functional iterations for eqs. (16) and (17) (substitution of the right-hand side into the left side), and a somewhat more complicated scheme for the proportion equation, eq. (15). Each iteration is subject to Aitken extrapolation, by fixed parameters contained in arrays PACCEL, VACCEL, and MACCEL which are currently all zero, corresponding to unaccelerated iteration.

2.2 MODE OF ITERATION

CLASSY does not use complete iterations on these equations at all times; when the estimates for a cluster are considered poor, statistics are calculated in a continuous or updated fashion. Mode selection is on a cluster-by-cluster basis. The absolute value of the variable INDEX(KL) is used to store the number or "name" of the cluster. The updating mode affects only the cluster parameters a, μ, and Σ, and does not change the processing of skewness, kurtosis, or likelihood ratio.

2.2.1 UPDATING MODE: (INDEX(KL) > 0)

Every newly created cluster is in the updating mode, the normal mode for a cluster whose statistics are not well defined. Also, a cluster processed through the routine ADJUST because its weight (W) exceeds its weight adjustment threshold (WADJ) is placed in the updating mode on finishing ADJUST.

Statistics are calculated on a current, or running, basis for clusters in this mode. The values of a, μ, and Σ appearing on the left of eqs. (15)-(17) change from each point processed to the next. That is, the sums on the left gain one additional point for each point processed, and this most recent value of the parameters is used for the next data point. Cluster parameters

are continuously varying, rather than corresponding to some particular iteration of the numerical scheme. Although this procedure may seem somewhat irregular, it follows the general dictum of numerical analysis, "use the most recent data." In fact, the values for a, μ, and Σ produced in this way should be better estimates of their true values, and thus lead to faster convergence of the numerical procedure. This technique, when combined with the short iteration procedure in which only a fraction of the points are processed between passes through ADJUST, allows the algorithm to find good coarse estimates of the cluster parameters relatively quickly, often within the first pass through the data rather than after several iterations. However, to be effective, the updating iteration mode requires that the data sample have no large-scale variation; i.e., that real data be scrambled or disordered to destroy point-to-point and regional correlation.

In the update mode, the cluster vector SUM is used to store $W\hat{\mu}$ (the current mean), and OSUM (Old SUM) is used to save the value of SUM when the cluster was last processed (created or last ADJUSTed). Similarly, the cluster array OVAR (Old VARiance) contains $W\hat{\Sigma}'$ (the covariance at last processing), and VRIN (VaRiance INverse) contains the inverse of $W\hat{\Sigma}$ (the current covariance matrix).

2.2.2 ITERATING MODE: (INDEX(KL) < 0)

A cluster is placed in iterating mode by ADJUST if it is processed by that routine due to a complete pass having been made through the data since its creation or last ADJUSTment (IDADJ(KL) $\leq$ NPTSO). A cluster in this mode observes the complete data sample between processing passes through ADJUST, and thus its numerical procedure can be considered to operate iteratively. (There is no fixed relation between the start of the iterative cycles of different clusters — one may start at point 5, a second at point 3000, and a third at point 12345. However, all the iterations for a given cluster start at the same pixel, unless (very rarely) the cluster reenters update mode.)

When a cluster is being processed in the iterating mode, the parameters a, μ, and Σ appearing on the left side of eqs. (15)-(17) are fixed at the

cluster's last pass through ADJUST and remain fixed until its next pass. Thus, the sums are calculated for an entire data sample using the same values of the parameters. This is necessary to actually solve the equations, since processing in the update mode for full passes through the data would lead to cyclic drifts in the parameters, which in turn would alter the value of the sums and the actual estimates. Although this should not be a large effect for scrambled or disordered data, the iterating mode is necessary to keep the program within the design objective of well-defined mathematics.

In the iterate mode, the cluster vectors SUM and OSUM have the same meaning as in the update mode, but OSUM is always used to calculate the mean used in the probability calculation. For the arrays OVAR and VRIN, the situation is different; VRIN contains the inverse of $W\hat{\Sigma}'$ (the old covariance) and OVAR contains $W\hat{\Sigma}$ (the current covariance). This is necessary because VRIN is used in numerous places as the inverse of the covariance matrix, both in the equations and as the metric for the space of data vectors.

2.3 TREE STRUCTURE

In addition to the continuous variables, CLASSY maintains a discrete tree structure of clusters and several continuous statistics used in updating this structure. This discrete structure differentiates CLASSY from a simple maximum-likelihood approximator using accelerated numerical techniques.

Under each cluster there may be two or more subclusters, each with a subcluster structure of its own. The sum of the distributions of the subclusters of a cluster is an alternative distribution to that represented by the cluster itself. As CLASSY is presently constituted, only the case of one cluster versus many can be represented; many versus many is not allowed. (One versus one is effectively handled by the continuous statistics section.)

Between each cycle through ADJUST, STATIS accumulates a likelihood ratio between the parent cluster and its subclusters. The natural logarithm of this ratio is maintained in the cluster variable SPFAC (SPlitting FACtor),

where SPFAC much less than 0 favors the parent cluster and SPFAC much greater than 0 favors the subclusters.

This likelihood ratio is used in ADJUST to make a final decision in favor of the parent cluster or its subclusters; the losing alternative is deleted by SEPER or by SUBLIM. SPFAC is also used in STATIS to choose the parent cluster probability or the sum of the subcluster resultant probabilities. The selection is done by a continuous changeover via $(P_{parent} + ZP_{subs})/(1 + Z)$, where Z = exp (SPFAC) is the likelihood ratio.

SPFAC is initialized when a cluster is created and each time it is ADJUSTed, and is set to the value returned by the subroutine APRIOR (currently a constant). This is the point of entry in the program for the necessary bias factor against large numbers of clusters, and for various volume normalization factors discussed in more detail under the APRIOR routine. The cluster variable OPRIOR (Old PRIOR) is also set to this initial value of SPFAC to retain the initial value for reference.

Another statistic accumulated by STATIS is the square of the normalized probability difference between a parent cluster and the sum of its subclusters, PQRAT.

$$\text{PQRAT} = \sum \left(\frac{P_{sub} - P_{parent}}{P_{sub} + P_{parent}} \right)^2 \tag{19}$$

This is a crude measure of how much a parent cluster differs from the mixture of its subclusters. If the subcluster total becomes nearly the same as the parent cluster density, and the likelihood ratio does not favor the subclusters, then ADJUST assumes that the parent is the "real" cluster, and eliminates the subclusters via SUBLIM. This is a common situation, since if the subclusters do not fit the data better (e.g., if the data are best fit by a single normal) then the subcluster parameters will change until the sum fits the single normal. Thus, subclusters are generally eliminated by becoming very similar to the parent cluster, rather than directly by an unfavorable likelihood ratio.

The other statistics maintained for each cluster are the traces of the skewness and kurtosis tensors (not divided by the total weight).

$$\mathrm{SKEW}_i = \sum \bar{x}_i \bar{x}^2 \tag{20}$$

$$\mathrm{KURT}_{ij} = \sum \bar{x}_i \bar{x}_j \bar{x}^2 \tag{21}$$

where

$$\bar{x}_i = x_i - \mu_i \quad ; \quad \bar{x}^2 = \bar{x}_i \bar{x}_j (\Sigma^{-1})^{ij} = \bar{x}^t \Sigma^{-1} \bar{x}$$

These are accumulated between ADJUSTments (or creation) of a cluster. (They are zeroed each time through ADJUST.) These statistics are summarized to make three scalar statistics in ADJUST and are then tested against thresholds calculated in CLINIT, to decide whether to consider splitting a cluster. The three summary statistics are derived in section 2.5 of this report. It is not necessary to calculate these statistics for a cluster with subclusters since such a cluster cannot be SPLIT again.

Three other variables maintained in CLASSY concern the normalization of the distributions and the volume integrals which are included in the normalization factors. These three are the cluster variables VOLIN, VOLRT, and DCON. The overall coefficient for a cluster probability density (excluding its proportion) is exp(-DCON/2)/VOLRT, VOLRT = $\mathrm{VOLIN}^{1/2}$, thus

$$\mathrm{VOLIN} * \exp(\mathrm{DCON}) = (2\pi)^{MQ} \det \Sigma$$

The splitting of the coefficient into two parts is made necessary by the fact that the Σ actually used includes W as a factor, and after a determinant is taken, this is raised to the power of the number of channels, which can lead to exponent overflow/underflow conditions. The relative division of the coefficient between VOLIN and DCON is made in a constant fashion in ADJUST.

2.4 EQUATIONS FOR THE PROPORTION CALCULATION

The mean and covariance parameters for a cluster are calculated using direct substitution of the left-hand side of eqs. (15) through (18) into the right-hand side.

However, this procedure is not very sensitive for the proportion equations. Because the proportion values enter very strongly into all the equations, it was deemed desirable to use a somewhat more complicated but faster converging system of equations to obtain the proportions.

It should be noted that there is no single iterative procedure for solving a given set of equations; there are an infinite number of systems with the same fixed point and differing rates and manners of convergence to this fixed point. In particular, direct substitution using eqs. (15)-(18) commonly taken to be the maximum-likelihood procedure is only one of many. While equations (15)-(18) are the maximum-likelihood equations, many differing techniques for solving them can be found in any standard text on numerical analysis; e.g., changes of variable, Newton's method, Aitken extrapolation, and methods derived from these. It is almost never true that the apparently simplest numerical technique is the best.

Although CLASSY was originally designed to use a sparse matrix variant of Newton's method, the only special techniques actually employed are the update mode, the extrapolations in ADJUST, a Monte Carlo method used in STATIS, and the following modification of the proportion calculation system.

Substituting eq. (18) into eq. (15) and deleting j-sum parameters and circumflexes:

$$a_i = \frac{1}{N}\sum \frac{a_i p_i}{\sum_{k=1}^{m} a_k p_k} = \frac{1}{N}\sum \frac{a_i p_i}{P} \tag{22}$$

where

$$P = \sum_{k=1}^{m} a_k p_k \tag{23}$$

Now a_i does not depend on j, so the two sides may be canceled, getting

$$1 = \frac{1}{N}\sum \frac{p_i}{P} \tag{24}$$

We may now define

$$q_i = \frac{1}{1 - a_i} \sum_{\substack{k=1 \\ k \neq i}}^{m} a_k p_k$$

so $P = ap + (1 - a)q$. Transposing eq. (24) and combining, we get

$$\left.\begin{aligned} 0 &= \frac{(1 - a_i)}{N} \sum \frac{p_i - q_i}{a_i p_i + (1 - a_i) q_i} \\ &= \frac{1 - a_i}{N} \sum D_i \end{aligned}\right\} \qquad (25)$$

where

$$D_i = \frac{p_i - q_i}{a_i p_i + (1 - a_i) q_i}$$

Now D_i just changes sign if p_i and q_i are switched, or equivalently, class i and not class i. It is the simplest variable in which to represent the proportion problem. In terms of D_i, the direct substitution proportion iteration is

$$a_i' = a_i + a_i(1 - a_i) \frac{1}{N} \sum D_i \qquad (26)$$

As expected, when eq. (24) is satisfied, $a' = a$.

It can be shown that for completely separated classes ($p_i = 0$ or $q_i = 0$ always) the direct substitution is exact in one iteration; that is $p_i = 0$ implies $D_i = \frac{-1}{1 - a_i}$ and $q_i = 0$ implies $D_i = \frac{1}{a_i}$. However, for points in between, the behavior of this iteration is soft and may coverge slowly, because (as seen in the derviation of eqs. (24) and (25))the intrinsic variability which allows the system of equations to be solved is in the denominator, $a_i p_i + (1 - a_i) q_i$, since it contains the only dependence on a_i. Because complete separation allows the a_i in the denominator to come out, it is possible for eq. (26) to

be exactly solved in this case. Mixed points always contribute part or all of their weight to holding the "status quo," or delaying the convergence of the equations; these points are always entered into the sums using the old value of a_i, and the form of the iteration does not take into account their changing contribution.

For example, if we consider a Newton's method iteration (assuming we are close to the solution), then we have

$$a_i' = a_i + \frac{1}{\frac{1}{N}\sum D_i^2} \frac{1}{N}\sum D_i \tag{27}$$

using

$$\frac{d}{da_i}\sum D_i = -\sum D_i^2$$

This approach substitutes $\frac{1}{\frac{1}{N}\sum D_i^2}$ for $a_i(1 - a_i)$ in eq. (26).

This is exact for complete separation, but otherwise Newton's method gives better convergence, because the points at $p_i = q_i$ do not contribute to the denominator at all.

The method used in CLASSY to calculate proportions is a simple system of this same general type which should not display any major problems.

By splitting the sum in eq. (25) into parts corresponding to $p_i > q_i$ and $p_i < q_i$ and dropping the class index i, we have

$$\sum_{p>q} D + \sum_{p<q} D = 0$$

where every part of the first term is positive, and the second term is negative. In line with the cancellation leading to eq. (24), this becomes

$$\frac{1}{a'}\left(a \sum_{p>q} D\right) + \frac{1}{1 - a'}\left[(1 - a) \sum_{p<q} D\right] = 0$$

where a' = a implies a solution. Solving for a',

$$\left.\begin{aligned} a' &= \frac{a \sum_{p>q} D}{a \sum_{p>q} D - (1 - a) \sum_{p<q} D} \\ &= \frac{\sum_{p>q} aD}{N - \sum_{p>q} \frac{q}{P} - \sum_{p<q} \frac{p}{P}} \\ &= a + a(1 - a) \frac{\sum D}{N - \sum_{p>q} \frac{q}{P} - \sum_{p<q} \frac{p}{P}} \end{aligned}\right\} \quad (28)$$

(recall P = ap + (1 - a)q). In the first form, both denominator terms are positive.

The second form corresponds to the variables used in CLASSY — the numerator is retained in the variable CIN, and the terms

$$\sum_{p>q} \frac{q}{P} + \sum_{p<q} \frac{p}{P}$$

used in the denominator are the variable CTOT. The actual proportions currently used in CLASSY are the proportion of a cluster relative to its parent cluster. The proportion as actually used is calculated for each pixel in the first loop of STATIS (in update mode), and in ADJUST using eq. (28), and is retained in the class variable PROP(KL). The routine DENCAL (Denominator CALculation) is used to readjust the values of CIN and CTOT when the tree is restructured. This is required by the use of proportions relative to the parent.

The third form shows the relation of this system to direct substitution, eq. (26). The motion is amplified if there are mixed points, and the denominator sums are 0 if p = 0 or q = 0. The relation of this amplification and that used by Newton's method (eq. (27)) has not been analyzed.

The proportion system in CLASSY could be improved. The use of proportions relative to the parent cluster appears to be a design fault, although the effects of this choice are fairly pervasive, and the chief benefit achieved by making the change to absolute proportions would be a cleaner program.

The actual proportion calculation system used was a first attempt at achieving better convergence than is allowed by direct substitution, and it is subject to all the problems of first attempts. It appears at present that a Newton's method approach, if combined with a change of variables or a model which prevents overshoot (violation of $0 \leq a \leq 1$) would be best. It is possible that such a scheme could be cast into a form sufficiently similar to the one used here that only the equations for updating CIN and CTOT in STATIS would need to be changed. Note that the separation between $p > q$ and $p < q$ is somewhat arbitrary; in fact, a fractional separation which appears close to eq. (27) and is a function of the current proportions could be used. In any such system, the designer, besides making sure that Newton's method did not overshoot, must also ensure that the system combines properly with the continuous changes present in update mode. Such a change should improve the convergence behavior of the program. In designing such a system, it is useful to work in terms of the symmetric variables $b = 2a - 1$ and $r = \frac{p - q}{p + q}$, where $-1 \leq b$ and $r \leq 1$. Then $D = \frac{r}{1 + br}$, and all the formulas pick up a useful symmetry under $b \to -b$, $r \to -r$. The overall proportion system must be invariant under this transformation.

2.5 LINEAR ALGEBRA AND GENERAL LINEAR TRANSFORMATIONS

The CLASSY algorithm was designed to be invariant under arbitrary linear transformations of the brightness space. That is, if we transform all the pixels or data points via

$$x'_J = Mx_J \tag{29}$$

where M is an arbitrary fixed nondegenerate matrix, then CLASSY should give the corresponding results: same cluster tree, same probabilities for each

point to be in a cluster, same proportions, etc. Means and covariances should be changed by

$$\left.\begin{aligned} \mu_i' &= M\mu_i \\ \Sigma_i' &= M\Sigma_i M^t \end{aligned}\right\} \qquad (30)$$

Thus CLASSY is transparent to general linear coordinate transformations: it does not "know" what coordinate system it is in.

There are two exceptions to this rule, both in ADJUST:

a. CLASSY assumes that the input data have been discretized with interval 1 along the coordinate axes. This must be compensated by a Shepherd's correction which is not generally invariant, since the discretization is not coordinate invariant. When the algorithm was operated without this correction it tended to find point clusters, or clusters with zero covariance along crystal planes of the discretization lattice, which are, after all, the true clusters in the data.

b. To limit computing time, the calculations used to select candidates for JOINing compare the diagonal elements of the covariance matrices for similarity. This is a noninvariant operation.

In d channels, the set of nondegenerate matrices form a mathematical group, called GL(d). Vectors form indivisible spaces under the group, loosely called representations. Breaking the system down into irreducible representations greatly simplifies the analysis of the clustering problem, since only certain operations and relations are proper. This type of insight was used throughout the design of CLASSY. In particular, the only way to take the product of two vectors is with the inverse covariance matrix.

A method of notation for systems of the type which are invariant under GL groups is called the "summation convention." In it, normal (column) vectors are represented by quantities with subscript indices (x_i). Vectors in the dual space (row vectors) are represented by superscript indices (not to be confused with exponents which almost never occur in this context), as (A^i).

A typical dual vector is the derivative with respect to a normal vector: $\partial^i = \partial/\partial x_i$. Other objects may have multiple upper or lower indices, or both. If an index appears twice in a term (as a superscript and a subscript), a summation over all values of the pair, called a contraction, is automatically implied. All the rest of the indices are called free indices, and must appear identically, including upper or lower position, in all terms added together, and on both sides of an equation. This notation is extremely convenient for use with vector/tensor systems such as those appearing in CLASSY, and will be invoked several places in this report, where it is irreplaceable. Examples follow:

T^i_j, M^i_j ordinary transformation matrices

$P^i_r = T^i_j M^j_r = M^j_r T^i_j$ their matrix product $P = MT$

$Q^i_r = T^j_r M^i_j = M^i_j T^j_r$ their product $P = TM$

(Note that order does not matter: the order information is contained in the index pairing.)

M^i_i the trace of M, Tr M

$T^i_j M^j_i$ the trace of TM, Tr(TM) = Tr(MT)

δ^i_j the Kronecker delta, = 1 if i = j, otherwise 0 (equivalent to the unit matrix; like all matrices must always have one upper and one lower index)

$\mu^i = \frac{1}{N} \Sigma x_i$ definition of the mean

$\Sigma_{ij} = \frac{1}{N} \Sigma x_i x_j - \mu_i \mu_j$ definition of the covariance (Note that Σ is not a matrix, but a rank 2 covariant tensor. No transpose operations appear using this notation.)

$(\Sigma^{-1})^{ij}$ its inverse; note the index motion, which is required

$(\Sigma^{-1})^{ij}\Sigma_{jr} = \delta^i_r$ the definition of the inverse

$\Sigma_{ij} = \Sigma_{ji}$ states that Σ is symmetric (cannot be said of a matrix)

$A_{ij} = -A_{ji}$ states that A is skew-symmetric (cannot be said of a matrix)

$x'_i = T^j_i \, x_j$ transformation of x by the matrix T

$\Sigma'_{ij} = T^k_i \, T^\ell_j \, \Sigma_{k\ell}$ transformation of the symmetric tensor (covariance) by the matrix T

$\Sigma^{-1'ij} = (T^{-1})^i_k (T^{-1})^j_\ell (\Sigma^{-1})^{k\ell}$ transformation of the inverse of Σ by T (Note the index differences from above.)

Much of the conventional nomenclature for multivariate statistics becomes incorrect in this viewpoint. The covariance is not a matrix, but a rank 2 covariance tensor, Σ_{ij}, and its inverse is a rank 2 contravariant tensor, $(\Sigma^{-1})^{ij}$. Correspondingly, the use of the "transpose" operation is invalid; it is only necessary in the standard nomenclature because of the attempt to represent the covariance as a matrix. Except in this section or as stated the standard nomenclature will be used to improve communication. The above notational convention is well known and fairly standard; the reader may find more detailed accounts in any elementary book on group theory, in texts on mathematical physics, or in books on multivariate analysis.

Occasionally in this report, the term "scalar" is used to refer to quantities that do not change under the transformation; e.g., ordinary numbers.

The measures of skewness and kurtosis used in CLASSY are the traces or contractions of the complete skewness and kurtosis tensors. The complete tensors are

$$S_{ijk} = \frac{1}{N} \sum \bar{x}_i \bar{x}_j \bar{x}_k \tag{31}$$

$$K_{ijk\ell} = \frac{1}{N} \sum \bar{x}_i \bar{x}_j \bar{x}_k \bar{x}_\ell \tag{32}$$

where $\bar{x} = x - \mu$. In 16 channels, S_{ijk} has 816 components, and $K_{ijk\ell}$ has 3876;

they are thus too large to be employed efficiently in a program such as CLASSY, and instead their traces are used:

$$S_i = S_{ijk}(\Sigma^{-1})^{jk} = \frac{1}{N}\sum \bar{x}_i\bar{x}_j\bar{x}_k(\Sigma^{-1})^{jk} = \frac{1}{n}\sum \bar{x}_i\bar{x}^2 \tag{33}$$

$$K_{ij} = K_{ijk\ell}(\Sigma^{-1})^{k\ell} = \frac{1}{N}\sum \bar{x}_i\bar{x}_j\bar{x}_k\bar{x}_\ell(\Sigma^{-1})^{k\ell} = \frac{1}{n}\sum \bar{x}_i\bar{x}_j\bar{x}^2 \tag{34}$$

where $\bar{x}^2$ is the distance from the cluster center: $\bar{x}^2 = \bar{x}\Sigma\bar{x}$.

The cluster vector SKEW is (W - OW) * S_i, and the array KURT is (W - OW) * K_{ij}. (The OW is present because SKEW and KURT are rezeroed each time out of ADJUST.)

S_i transforms as an irreducible vector under GL(d), but K_{ij} can be separated into two components:

$$K = K_{ij}(\Sigma^{-1})^{ij} \tag{35}$$

and

$$K^\circ_{ij} = K_{ij} - \frac{1}{d}\Sigma_{ij}K \tag{36}$$

Note that $K^\circ_{ij}(\Sigma^{-1})^{ij} = 0$, so that this operation is referred to as separating K_{ij} into its trace and its traceless components, both of which are incapable of further reduction. K°_{ij} has no associated scalar, therefore the lowest order scalar terms available from these contracted third and fourth moments, in addition to K, are:

$$S^2 = S_iS_j(\Sigma^{-1})^{ij} \tag{37}$$

$$\left.\begin{aligned}(K^\circ)^2 &= K^\circ_{ij}K^\circ_{i'j'}(\Sigma^{-1})^{ii'}(\Sigma^{-1})^{jj'} \\ &= K_{ij}K_{i'j'}(\Sigma^{-1})^{ii'}(\Sigma^{-1})^{jj'} - \frac{K^2}{d}\end{aligned}\right\} \tag{38}$$

These are the simplest forms; the rest are of higher order in both K, S and x (such as $S_i(\Sigma^{-1})^{ij}K_{jk}(\Sigma^{-1})^{k\ell}S_\ell$).

These three quantities are used in ADJUST to test whether a cluster appears normal or is a candidate for splitting. Currently all three are used; but if the algorithm were modified to allow nonnormal and skewed clusters, then only $(K^\circ)^2$ gives meaningful information about split clusters.

In ADJUST, TRK represents K (and later $[K - d(d + 2)]/\sqrt{W - OW}$), normalizing to the expected random variation). SK represents DW * S^2 and URK represents DW * $(K^\circ)^2$. SK, URK, and the later TRK are all normalized so that random fluctuations have a size independent of DW.

The three components each have a specific meaning:

a. S is a vector indicating how far and in which direction the base (high $\bar{x}^2$) of the distribution is shifted from the peak (low $\bar{x}^2$).

b. K has the value d(d + 2) for a normal distribution. Larger K's represent distributions which are more pointed than a normal distribution independent of direction (lepto-kurtosis); smaller K's represent distributions which are less pointed and with smaller "tails" (endo-kurtosis).

c. The third tensor, K°, represents the tendency for points in each set of opposite directions to be at a different distance from the center compared to some of other pair of directions. In other words, K° measures how "lumpy" the distribution is when observed on a sphere at some fixed distance from the center. Since this lumpiness is characteristic of multimodal distributions in several dimensions, K° is really the best measure of multimodality used in CLASSY. The other tests have been included to maintain consistency with the formal description of the program as fitting the given distribution with a mixture of normals. Also, unless other tests or a precise formal model were used, these tests could ultimately mask multimodal situations if ignored.

2.6 EQUATIONS FOR A MIXTURE OF TWO DISTRIBUTIONS

The following formulae are the first four moments of a general mixture of two normal distributions; essential use is made of them in the routines JOIN and SPLIT. The summation convention is used.

Consider a normal distribution in d variables with covariance σ^2, in two cases: (1) total weight 1, mean 0; and (2) total weight a, mean μ. In the following table, the portion in single quotes gives the basic character of the term, which must then be symmetrized on indices.

Moment	Weight 1, mean 0	Weight a, mean μ
0	1	a
1	0	$a\mu$
2	σ^2	$a\left(\mu_i\mu_j + \sigma_{ij}{}^2\right)$
3	0	$'a\mu(\mu^2 + 3\sigma^2)' = a\left(\mu_i\mu_j\mu_k + \mu_i\sigma^2_{jk} + \mu_j\sigma^2_{ik} + \mu_k\sigma^2_{ij}\right)$
4	$'3\sigma^4' = \sigma^2_{ij}\sigma^2_{k\ell} + \sigma^2_{ik}\sigma^2_{j\ell} + \sigma^2_{i\ell}\sigma^2_{jk}$	$'a(\mu^4 + 6\mu^2\sigma^2 + 3\sigma^4)' = a\left(\mu_i\mu_j\mu_k\mu_\ell + \mu_i\mu_j\sigma^2_{k\ell} + \mu_i\mu_k\sigma^2_{i\ell} + \mu_i\mu_\ell\sigma^2_{jk} + \mu_j\mu_k\sigma^2_{i\ell} + \mu_j\mu_\ell\sigma^2_{ik} + \mu_\ell\mu_k\sigma^2_{ij} + \sigma^2_{ij}\sigma^2_{k\ell} + \sigma^2_{ik}\sigma^2_{j\ell} + \sigma^2_{i\ell}\sigma^2_{jk}\right)$

A mixture of two such distributions will be used.

	Cluster 1	Cluster 2	Remark
Weight	a	b	$a + b = 1$; let $c = a - b$
Mean	μ	ν	Let $\delta = \mu - \nu$
Covariance	σ^2	τ^2	Let $D^2 = \sigma^2 - \tau^2$

These are viewed as a single distribution, with weight 1, mean μ, covariance Σ^2 (inverse Σ^{-2}), skewness S, and kurtosis K (with the $3\Sigma^4$ term also subtracted), as shown in table I.

TABLE I.— MOMENTS OF THE MIXTURE OF TWO NORMAL DISTRIBUTIONS

Moment	Remark
0	$1 = a + b$
1	$\mu = a\mu + b\nu$
2	$'\Sigma^2 = ab\delta^2 + a\sigma^2 + b\tau^2{}'$ $\Sigma^2_{ij} = ab\delta_i\delta_j + a\sigma^2_{ij} + b\tau^2_{ij}$
3	$S_i = (\Sigma^{-2})^{k\ell} S_{ik\ell}$ $= (\Sigma^{-2})^{k\ell}\left(\frac{1}{N}\Sigma x_i x_k x_\ell - \mu_i\Sigma^2_{k\ell} - \mu_k\Sigma^2_{i\ell} - \mu_\ell\Sigma^2_{ik} - \mu_i\mu_k\mu_\ell\right)$ $= '\mathrm{ab}(\Sigma^{-2})\delta(3D^2 - c\delta^2)'$ $= \mathrm{ab}(\Sigma^{-2})^{k\ell}\left(\delta_i D^2_{k\ell} + \delta_k D^2_{i\ell} + \delta_\ell D^2_{ik} - c\delta_i\delta_k\delta_\ell\right)$
4	$K_{ij} = (\Sigma^{-2})^{k\ell} K_{ijk\ell}$ $= '\Sigma^{-2}\left(\frac{1}{N}\Sigma^4 - 4\mu S_{(3)} - 6\Sigma^2\mu^2 - \mu^4 - 3\Sigma^4\right)'$ $= (\Sigma^{-2})^{k\ell}\left(\frac{1}{N}\Sigma x_i x_j x_k x_\ell - \mu_i S_{jk\ell} - \mu_j S_{ik\ell} - \mu_k S_{ij\ell} - \mu_\ell S_{ijk}\right.$ $- \mu_i\mu_j\Sigma^2_{k\ell} - \mu_i\mu_k\Sigma_{j\ell} - \mu_i\mu_\ell\Sigma^2_{ik} - \mu_j\mu_k\Sigma^2_{i\ell} - \mu_j\mu_\ell\Sigma^2_{ik} - \mu_k\mu_\ell\Sigma^2_{ij}$ $\left.- \mu_i\mu_j\mu_k\mu_\ell - \Sigma^2_{ij}\Sigma^2_{k\ell} - \Sigma^2_{ik}\Sigma^2_{j\ell} - \Sigma^2_{i\ell}\Sigma^2_{jk}\right)$ $= '\Sigma^{-2}(3\mathrm{ab}D^4 + \mathrm{ab}(1 - \mathrm{ab})\delta^4 + 6c\mathrm{ab}\delta^2 D^2)'$ $= (\Sigma^{-2})^{k\ell}\left[\mathrm{ab}\left(D^2_{ij}D^2_{k\ell} + D^2_{ik}D^2_{j\ell} + D^2_{i\ell}D^2_{jk}\right) + \mathrm{ab}(1 - \mathrm{ab})\delta_i\delta_j\delta_k\delta_\ell\right.$ $+ c\mathrm{ab}\left(\delta_i\delta_j D^2_{k\ell} + \delta_i\delta_k D^2_{j\ell} + \delta_i\delta_\ell D^2_{jk} + \delta_j\delta_k D^2_{i\ell} + \delta_j\delta_\ell D^2_{ik}\right.$ $\left.\left. + \delta_k\delta_\ell D^2_{ij}\right)\right]$

Using the definition of c,

$a = \frac{1 + c}{2}$, $b = \frac{1 - c}{2}$, $\mathrm{ab} = \frac{1 - c^2}{4}$, and $1 - 6\mathrm{ab} = \frac{3c^2 - 1}{2}$

3. DESCRIPTION OF SUBROUTINES

Most subroutines used in CLASSY have simple, nonmathematical descriptions; they do simple bookkeeping, structural manipulations, or simple functions such as printout or matrix algebra. These routines are not described in this section.

The following subroutines which have mathematical properties and mathematical descriptions are described in this section:

STATIS (Statistics)
ADJUST (Adjustment)
JOIN (Combines)
SPLIT (Splitting)
DENCAL (Denominator calculation)
APRIOR (A priori distribution)
ISPLIT (Is split)
EIGROT (Eigenrotation)

STATIS and ADJUST are the subroutines which control the processing in CLASSY. STATIS handles all the incremental statistics, essentially doing all the accumulation required by eqs. (15)-(18). STATIS also contains the code generating the DO loop over the data points (which could have been placed outside it). ADJUST is called by STATIS for each cluster on a specified basis (if either of two given thresholds are exceeded). ADJUST in turn does all action on a cluster which is done on a lumped basis: making tests for split clusters, separable clusters, joinable clusters, etc. ADJUST is also in charge of all extrapolation of continuous parameters, subtracting old data from the sums of eqs. (15)-(18) so that the system can update or iterate properly, not depending on bad data values from earlier data or iterations. In general, ADJUST handles any operation that is not executed every time a point is entered into a sum and is in charge of testing for and calling all the tree-restructuring operations. The structure of STATIS is fairly well dictated by the mathematics, while that of ADJUST is largely heuristic.

3.1 STATIS

STATIS, besides generating the pixels and triggering ADJUST, calculates the quantities in eqs. (15)-(18), together with a few other statistics such as skewness, kurtosis, and likelihood ratio which are used in the structural change tests in ADJUST. STATIS also contains the logic which modifies the calculations of these statistics, depending on whether a given cluster is in update or iterate mode. The decision as to which mode a cluster should be in is made in ADJUST.

STATIS is divided into two blocks, or loops, over the cluster tree. The first of these calculates the class conditional and posterior probabilities for each class, and the second updates all the statistical sums. The first loop consumes most of the execution time of CLASSY, since all clusters must be processed through each pixel, and for each a quadratic form must be evaluated. This loop starts at FORTRAN statement 31 and continues to just before statement 150 (see section 4.1). For each point, the first pass defines for each cluster KL:

$$\left.\begin{aligned} \text{PCUM(KL)} &= \text{(Probability acCUMulator)} \\ &= \sum_{i\varepsilon KL} a_i p_i(x) \end{aligned}\right\} \qquad (39)$$

where $i\varepsilon KL$ indicates i is a subcluster of KL, and PCUM is later normalized by PRIRCM.

$$\left.\begin{aligned} \text{PRIRCM(KL)} &= \text{(PRIORs acCUMulator)} \\ &= \sum_{i\varepsilon KL} a_i \end{aligned}\right\} \qquad (40)$$

$$\left.\begin{aligned} \text{PROP(KL)} &= \text{(PROPortion)} \\ &= \text{CIN(KL)}/[W_F - \text{CTOT(KL)}] \end{aligned}\right\} \qquad (41)$$

where W_F is the weight of the parent

$$\left.\begin{aligned} \text{PCOND(KL)} &= \text{(CONDitional Probability)} \\ = P_{KL}(x) &= \frac{1}{(2\pi)^{d/2}|\Sigma_{KL}|} e^{-1/2(x-\mu_{KL})^T \Sigma_{KL}^{-1}(x-\mu_{KL})} \end{aligned}\right\} \qquad (42)$$

A selection is made whether to use PCUM, the sum of the subcluster probabilities, or PCOND as the output probability for this cluster (PST). This is made in a continuous fashion,

$$PST(KL) = a_{KL} \frac{PCOND(KL) + e^{SPFAC}\, PCUM(KL)}{1 + e^{SPFAC}} \tag{43}$$

essentially selecting the cluster or its subclusters in proportion to the likelihood ratio between them. For SPFAC sufficiently large or small, PST(KL) is clamped to PCUM or PCOND, respectively, with thresholds XOVFLO and XUNFLO. PST(KL) = (P STored) is the final output for cluster KL.

The subroutine CORECT has been used to calculate $x - \mu$ for each cluster, storing the result in REL (RELative pixel), and the cluster variable DISS contains the value of the quadratic form $(x - \mu)^T \Sigma^{-1} (x - \mu)$ including DCON. When this form is too large, the cluster is not processed for this point.

The second loop of STATIS updates all the statistics being accumulated for the cluster: direct statistics, skewness (SKEW) and kurtosis (KURT), the log likelihood ratios between a parent cluster and its subclusters (SPFAC), and of the probability difference between the parent and its subclusters (PQRAT). This latter variable is used to determine if the parent cluster and its subclusters have come to be practically the same distribution. The temporary ZQ used here is $r = \frac{p - q}{p + q}$ as described in section 2.4, and a short approximation to the log is used in computing SPFAC.

All the basic variables updated (W, SUM, VRIN, CIN, CTOT, SKEW, KURT, SPFAC, and PQRAT) were described in section 2.

Three dependent variables are also maintained by STATIS: VOLIN, VOLRT, and DCON, described in section 2.3; with VRIN, these have special updating requirements which must be described.

VRIN is the inverse of the covariance a_{ij}, which is not directly maintained.

$$a_{ij} = \sum \bar{x}_i \bar{x}_j$$

where $\bar{x}_i = x_i - \mu_i$ or

$$a = \sum \bar{x}\, \bar{x}^T$$

using matrix notation. Updating a

$$a' = a + \bar{x}\, \bar{x}^t$$

Then

$$\left.\begin{aligned} VRIN' &= (a + \bar{x}\,\bar{x}^t)^{-1} \\ &= a^{-1} - a^{-1}\bar{x}\,\bar{x}^t a^{-1} + a^{-1}\bar{x}\,\bar{x}^t a^{-1}\bar{x}\,\bar{x}^t a^{-1} - \bar{x} \cdots \\ &= VRIN - (a^{-1}\bar{x})(\bar{x}^t a^{-1})(1 - \bar{x}^t a^{-1}\bar{x} + (\bar{x}^t a^{-1}\bar{x})^2 \cdots \\ &= VRIN - \frac{(a^{-1}\bar{x})(a^{-1}\bar{x})^t}{1 + \bar{x}^t a^{-1}\bar{x}} \end{aligned}\right\} \quad (44)$$

which may be verified by direct multiplication. Note that $\bar{x}^t a^{-1}\bar{x}$ is the exponential argument from the multivariate normal distribution.

Similarly, if V = det(a), the determinant of a, then

$$\left.\begin{aligned} V' &= \det(a + \bar{x}\,\bar{x}^t) \\ &= \det(a)\,(1 + \bar{x}^t a^{-1}\bar{x}) \\ &= V(1 + \bar{x}^t a^{-1}\bar{x}) \end{aligned}\right\} \quad (45)$$

This may be obtained by expanding $\bar{x}\,\bar{x}^t$ terms by minors, and using the definition of the inverse in terms of cofactors. VOLIN is the same as det(a) up to a constant factor, and is updated by the same formula.

VOLRT is updated by one cycle of Newton's method to follow $(VOLIN)^{1/2}$, and DCON is updated (using an approximation to the log), to compensate for the

factor of W**AMQ which appears in VOLIN and affects VOLRT. This factor appears because the terms in the covariance are added before the determinant is taken, but without dividing by the total weight to get the actual covariance. This is unavoidable and is easily handled using the DCON variable. The formulas actually used in the program are generalizations to the case with variable weights.

After updating all the statistics, STATIS checks to see if a variable is over threshold (W $\geq$ WADJ or NPTSO $\geq$ IDADJ) and saves the index of at most one cluster for passing to ADJUST after the current data point has been fully processed. The delay is necessary to avoid overlapping usages of certain EQUIVALENCE cluster variables and to otherwise ensure that processing is complete in spite of the restucturings.

In iterate mode, everything but VRIN, VOLIN, VOLRT, and DCON is updated similarly; PROP is not updated from CIN and CTOT in the first loop in this case, and OSUM is used rather than SUM in calculating the mean. In addition, OVAR, which in this case represents the current rather than the old values of the variance, is updated directly. The vector COVEC is $\frac{1}{W}(a^{-1}\bar{x})$, ALOW = $\frac{P}{W'}$, ALPHA = $\frac{W}{W'}P$, and

$$\text{COFI} = \frac{-WP}{W'\left[1 + P\,\frac{(\bar{x}^t a^{-1}\bar{x})}{W}\right]}$$

where

W = the old weight

W' = the new weight for the cluster

COFI = the coefficient used to update VRIN

Although eqs. (15)-(18) indicate that every point is added to every cluster, certain shortcuts were taken in CLASSY to speed processing. Normally, each data point will have fairly large probabilities for one or a few clusters. The remainder will be far out of range, with their probabilities damped by a large exponential argument. These terms could be deleted without causing any problem and would usually be effectively eliminated by the machine's floating-point hardware. Classy handles this situation in STATIS at the Monte Carlo loop starting at statement 132.

To avoid introducing a bias, a small number of the low-probability points are given a correspondingly increased probability and processed normally. The procedure follows: first, any point with probability greater than the threshold parameter PLIM (defined in CBLO) is automatically processed. For the remainder, an integer from 0 through MONTE-1 is selected randomly by the random number routine DISC. If that number is a specified value (1), then the probability is multiplied by AMONTE. (AMONTE=MONTE is also a parameter.) The program then returns to the PLIM threshold check and continues to loop until P > PLIM or the random integer misses the specified value. If the random integer ever misses, then the current point is not processed through this cluster. Thus, if the probability for a given point in a given cluster falls below PLIM, the probability has a $\frac{1}{\text{MONTE}}$ chance to be multiplied by MONTE and considered further, and a $\frac{\text{MONTE-1}}{\text{MONTE}}$ chance to be ignored. The bias which easily crops up in a tail-truncation procedure is thus eliminated.

Presently, the values of PLIM and MONTE (AMONTE) are fixed throughout a run. If the second half of STATIS should ever consume excessive time, a modification could allow a large value of PLIM during early processing to be followed by a drop to a small value for the last few passes.

3.2 ADJUST

ADJUST is entered periodically to adjust a cluster via extrapolation of data, and elimination of old data from the continuous statistical parameters of a cluster, and to make the tests required to decide on discrete transformations of the cluster tree. Most of the separate operations occurring in ADJUST are unrelated. ADJUST also gets and frees storage for temporary matrices used by itself and by the discrete transformation subroutines it calls, particularly SPLIT.

Before returning, ADJUST sets the old values of all the statistics to their current values and calculates the thresholds WADJ (Weight ADJust) and IDADJ (ID ADJust) for the next call to ADJUST. WADJ is set to a quantity which exceeds W by an amount which is the increase allowed in W before the next

ADJUST. This increase is currently a fixed parameter DWFAC > 1 times W, but could be set to a value dependent on the stability of the cluster. The IDADJ threshold is included so that data is not double-counted; that is, to ensure that every cluster is ADJUSTed at least once for every pass through the data. ADJUST sets IDADJ to the current point number (NPTSO) plus the total number of pixels in the data set (TOTPIX). Before returning, ADJUST also determines the mode of the cluster (update or iterate), depending on whether ADJUST was entered due to WADJ or IDADJ, respectively.

In processing the continuous statistics, ADJUST first subcontracts the old values of the accumulated statistics from the current values to ensure that no data older than the previous call to ADJUST (or the cluster creation) is included in the new statistics. In addition, the motion of the parameters since the last call to ADJUST is calculated, and the new parameters are set to overshoot their current, subtracted values by this motion times certain acceleration factors.

The acceleration factors (currently 0) are a function of whether the cluster was in update or iterate mode, and are in the arrays PACCEL (Proportion Acceleration), VACCEL (mean (Vector) Acceleration), and MACCEL (covariance (Matrix) Acceleration). These are indexed by the internal temporary KADTY (KADTY = 1 for update mode, = 2 for iterate mode). These extrapolations are done using the temporary EXF, which contains various weight factors via the temporary WINFC. The scheme is an ordinary extrapolation or accelerated convergence scheme for a set of equations in a set of variables, and will not be discussed further since presently the parameters are zero. The CTOT's of the subclusters and sibling clusters must be modified during the updating, due to the use of relative proportions.

The discrete changes in the cluster tree are made whenever a cluster being adjusted passes a test for some particular change. There are five such cluster tree transformations; they are listed in table II with the routine making the transformation, an abstract of the test used, and parameters from common, upon which each test depends. Parameters used in the WADJ calculations are also included.

The statistics and tests used to determine if a trial SPLIT is to be made are described in sections 1.1 and 2.5. The likelihood ratio tests and PQRAT test which govern the calls to SEPER, SUBLIM, and ELIM are also discussed in sections 1.1, 2.3, and 2.5.

The test to determine whether a trial join is advisable is based on a heuristic criterion which compares the mean vectors for two clusters and the diagonal elements of the covariance matrices. This criterion is given by

$$R_{ij} = \frac{(\mu_i - \mu_j)^T \left(\frac{W_i \Sigma_i^{-1} + W_j \Sigma_j^{-1}}{W_i + W_j} \right) (\mu_i - \mu_j) + A \sum_{k=1}^{d} \left(\ln \left| \sigma_{kk_i} \right| - \ln \left| \sigma_{kk_j} \right| \right)}{B \left(\frac{W_j}{W_i} + \frac{W_i}{W_j} \right)^2 + 1} \tag{46}$$

where

W_i = current weight for cluster i

A and B = arbitrary constants (currently, A = 0.3 and B = 0.18)

The first term in the numerator is the distance between the mean vectors of clusters i and j, weighted by an average computed from the inverse covariance matrix for clusters i and j. The second term in the numerator is a measure of the difference in the diagonal elements of the two covariance matrices. The diagonal elements rather than the full covariance matrices are used for computational simplicity. A more complete expression involving all covariance terms is $\ln \det \Sigma_1 \Sigma_2^{-1}$. The denominator is designed to discriminate against small clusters in the sense that R_{ij} will be artifically reduced if the weight of one cluster is small relative to the weight of the other cluster. This factor is designed to give large clusters an opportunity to absorb small clusters if such a join does not substantially affect the statistics of the larger cluster.

The R_{ij} criterion is computed for certain clusters having the same parent as cluster i; the clusters to be checked are selected on a Monte Carlo basis.

If the cluster j for which R_{ij} is a minimum is less than a fixed threshold, the hypothesis is raised so that clusters i and j are really the same cluster. In practice, a new cluster at the next higher level in the cluster tree is created, with parameters and other statistics obtained by combining the values for the two similar clusters. This is accomplished by calling the JOIN subroutine.

TABLE II.– CLUSTER TREE TRANSFORMATION ROUTINES

Transformation	Routine	Test	Parameters
Generate two subclusters	SPLIT	Trace kurtosis, skewness, or traceless part of kurtosis too large. Must not yet have subclusters	TRBND, SKBND, URKBND, TRCHI, SKCHI, URKCHI, WAIT
Eliminate this cluster in favor of its subclusters	SEPER	Likelihood ratio strongly favors subclusters	SEPTH
Eliminate all subclusters of this cluster	SUBLIM	Likelihood ratio strongly favors parent clusters, or likelihood ratio is mediocre and subclusters very similar to parent cluster (using PQRAT)	SBLTH, PQRATH, SPMVTH
Eliminate this cluster and any subclusters	ELIM	This cluster proportion too small (and NOELIM switch is off)	ELIMTH, NOELIM
Make this cluster and a sibling cluster subclusters of a new cluster	JOIN	Sibling most similar to this in mean and covariance diagonal elements is sufficiently similar. Siblings to be checked are selected on a Monte Carlo basis. (Procedure is quite heuristic)	WDJOIN, PJOIN, VRJOIN, RLIM
Next adjustment point	WADJ		DWFAC, WSIM, WDELSM

3.3 JOIN

The subroutine JOIN takes two clusters which are subclusters of the same parent and combines them into one cluster, which is a subcluster of the original parent and has as subclusters the two original clusters. Aside from the bookkeeping necessary to modify the cluster tree, it must also calculate the statistics of the new cluster.

The weight and adjustment threshold of the new cluster are defined by parameters. Its likelihood ratio is given the APRIOR value, and PQRAT, SKEW, and KURT are zeroed. Its proportion is the sum of the proportions of its components, with the numerator CIN having the weighted average of the subcluster CIN's. DENCAL is called to give the subclusters the same relative weights they had previously, but relative to the new parent cluster.

The mean and covariance of the new cluster are μ and Σ^2 from table I, calculated directly in the DO loops ending at FORTRAN statement 21. Referring to table I, the relevant variables have the values

CA = a

CB = b (new statistics - update mode)

CBV = b (old statistics - iterate mode)

(Two values are necessary because of the different handling of some variables.)

CF = temporary (contains ab)

FA = temporary for weight adjustments

DELTA = $ab\delta\left(\frac{W_J}{W_B}\right)$

W_B = weight of the second subcluster

W_J = weight of the new cluster

The new statistics are stored in the standard form, and the mandatory calculations for VOLIN, VOLRT, and DCON are made, along with the mandatory storage of "old" statistical values.

3.4 SPLIT

SPLIT is a subroutine which guesses the optimal axis on which to split an existing cluster. SPLIT is called after ADJUST ascertains that the cluster should be split; SPLIT has available the skewness and kurtosis data for the cluster as additional information to use in finding the proper split. After SPLIT has calculated the proper statistics for the component clusters, it builds the clusters and links them into the cluster tree. The parent cluster is not changed in any way, but the new clusters are linked to it as subclusters. SPLIT does not actually split a cluster, but determines the best way to consider splitting it. The actual decision to split is made on the basis of likelihood ratio information. If final splitting is required, it is carried out by the routine SEPER. The two new clusters formed by SPLIT are created with small values of W and WADJ to allow them to move rapidly to fit the actual distribution, and to be ADJUSTed quickly. The new clusters are considered to be guesses, and are treated accordingly.

The equations to be solved by SPLIT are those for the mixture of two distributions (table I). SPLIT first puts everything in a frame with an overall mean M of zero, and where the overall covariance is the unit matrix. Under this convention, rank 2 tensors may be called matrices, or reference may be made to the inner product of two vectors, etc. Whenever such a usage is made, the covariance or its inverse intervenes; e.g., $\delta \cdot \delta$ for vector δ really means $\delta_i \Sigma^{-1ij} \delta_j$. Such requirements can always be tracked by the index summation convention, as the only way an upper index may be converted to a lower index or vice versa is with the covariance or its inverse. It is easiest to think of the covariance as being a unit matrix, in which case the group GL(d) of general linear transformations on spectral space is restricted to its subgroup O(d) of orthogonal transformations, which leave the covariance unchanged.

In order to define the two new clusters, SPLIT must find two new covariance matrices, the difference between the two new cluster means, and the difference between the two new proportions for a total of $2(d^2 + d)/2 + d + 1 = (d + 1)^2$ variables. Covariance matrices are represented by their square roots, the "standard deviations," to preserve positive difiniteness.

The overall statistics of this combined distribution must match those of the given cluster, including the skewness and kurtosis. The mean and proportion of the given cluster are taken care of automatically, so there are $d(d + 1)/2$ equations each for the covariance (matching it to 1) and the kurtosis, and d equations for the skewness, for a total of $d^2 + 2d$ equations, or one more unknown than there are equations.

The method used to solve these equations in the current version of SPLIT is rather crude, and could be greatly improved. After a first approximation is made, the subroutine tries to minimize a quadratic form of the squared difference between the three classes of statistics and their values for the current assignments of the independent variables, by using a steepest-descent type of algorithm. The quadratic form to be minimized is thus

$$\text{OBCOV}\ \|\hat{\Sigma} - \Sigma\|^2 + \text{OBSKEW}\ \|\hat{S} - S\|^2 + \text{OBKURT}\ \|\hat{K} - K\|^2 \tag{47}$$

where the circumflexed values refer to those calculated from the current values of the independent variables using the equations from table I, and the unmarked values are those derived from the statistics of the cluster to be SPLIT. When K is referred to in SPLIT, the value is used with the normal distribution offset of $[(2 + d)\ \Sigma]$ subtracted; this offset is the trace on one pair of indices of '$3\Sigma^4$'. Using the summation convention

$$'3\Sigma^4' = \Sigma_{ij}\Sigma_{k\ell} + \Sigma_{ik}\Sigma_{j\ell} + \Sigma_{i\ell}\Sigma_{jk}$$

$$('3\Sigma^4')\Sigma^{-1ij} = \Sigma_{j\ell} + d\Sigma_{j\ell} + \Sigma_{j\ell} = (d + 2)\Sigma_{j\ell}$$

OBCOV, OBSKEW, and OBKURT are arbitrary objective function coefficients defined in common. Written out, this objective function is:

$$\begin{aligned}
\text{obj} = \text{OBCOV}\ &\left(g\delta\delta^t + \frac{1 + c}{2}\sigma^2 + \frac{1 - c}{2}\tau^2 - I\right)^2 \\
&+ \text{OBSKEW}[g(\delta\,\text{Tr}D^2 + 2D^2\delta - c\delta^2\delta) - S]^2 \\
&+ \text{OBKURT}\Big\{2gD^2\text{Tr}D^2 + g(D^2)^2 + g\left(\frac{3c^2 - 1}{2}\right)\delta\delta^t \\
&- cg[\delta^2D^2 + (\text{Tr}D^2)\delta\delta^t + 2\delta\,(D^2\delta)^t + 2(D^2\delta)\delta^t] - K\Big\}^2
\end{aligned} \tag{48}$$

where σ, τ, D^2, c, and δ are parameters of the two clusters as defined in table I and in section 2.6; g is $(1 - c^2)/4$, S is skewness, and K is subtracted kurtosis. The above expression is written in matrix notation in a coordinate frame where Σ is the unit matrix (note the appearance of the unit matrix in the covariance term), and all matrix products, traces, vector inner products, etc., are taken using Σ as the metric. δ^2 is a scalar $= \delta^t\delta$ and $\delta\delta^t$ is a matrix.

As a steepest descent technique is used, the derivatives of the objective function with respect to the independent variables are required. Note that the derivative with respect to a vector is a vector, and that the derivative with respect to a matrix is a matrix. Also, the notation $\{A,B\} = AB + BA$ for matrices A and B is used to denote the anticommutator calculated by routine ACOM. The error terms in the objective function are written:
$E = \hat{\Sigma} - \Sigma = \hat{\Sigma} - I$; $T = \hat{S} - S$; $V = \hat{K} - K$; E and V are matrices and T is a vector. For brevity, A for OBCOV, B for OBSKEW, and C for OBKURT are written.

For the derivative,

$$\begin{aligned}\frac{1}{2}\frac{\partial obj}{\partial c} = {} & A\left[-\frac{c}{2}\delta^t E\delta + \frac{1}{2}\mathrm{Tr}(ED^2)\right] \\ & + BT\left(-\frac{c}{2}\frac{\hat{S}}{a} - a\delta^2\delta\right) \\ & + C\left\{-\frac{c}{2}\left[2\mathrm{Tr}(VD^2)\mathrm{Tr}D^2 + \mathrm{Tr}(VD^4)\right]\right. \\ & + \left(3cg - \frac{c}{2}b\right)\delta^t V\delta + \left(\frac{c^2}{2} - g\right)\left[\delta^2\mathrm{Tr}(VD^2)\right. \\ & \left.\left. + \delta^t V\delta\,\mathrm{Tr}D^2 + 4(V\delta)^t(D^2\delta)\right]\right\}\end{aligned} \tag{49}$$

where

$$(V\delta)^t(D^2\delta) = 2\delta^t\{V, D^2\}\delta \tag{50}$$

$$\frac{1}{2}\frac{\partial obj}{\partial \delta} = 2AgE\delta + gB[-2cT\delta\delta + (TrD^2 - c\delta^2 + 2D^2)I]$$

$$+ 2gC[(h - 2cTrD^2)V\delta - cTr(VD^2)\delta - 2c\{V,D^2\}\delta] \qquad (51)$$

where

$$h = \frac{3c^2 - 1}{2}$$

If we write

$$\frac{1}{2}\frac{\partial obj}{\partial \delta} = A\frac{c}{2}E + gB(T\delta I + T\delta^t + \delta T^t)$$

$$+ gC\Big\{(2TrD^2 - c\delta^2)V + (2TrD^2 - c\delta^t V\delta)I$$

$$+ \{V,D^2\} - 2c\Big[(V\delta)\delta^t + \delta(V\delta)^t\Big]\Big\} \qquad (52)$$

then

$$\frac{\partial obj}{\partial \sigma} = \left\{\sigma, \frac{\partial obj}{\partial D^2} + \frac{E}{2}\right\}$$

$$\frac{\partial obj}{\partial \tau} = \left\{\tau, \frac{-\partial obj}{\partial D^2} + \frac{E}{2}\right\} \qquad (53)$$

The body of SPLIT is taken up with calculating the above objective function and its derivative. To handle the shortage of one available equation, the system is allowed to approach a minimum naturally, but the objective function is multiplied by $(1 + c^2 \cdot GAMCEN)$*, which tends to make the weights of the two clusters equal.

Internal variable names, temporary variables, and accumulators associated with variables used in this part of SPLIT are listed in table III.

*GAMCEN is a control parameter defined in a BLOCK DATA subroutine.

The actual steepest-decents procedure is not complicated. There is a certain ambiguity in any steepest-descents procedure in that the derivative vector is of a different type from the dX needed as an increment. This is an example of the mathematics elaborated in section 2.5; essentially, any steepest-descents procedure yields different results in different coordinate systems. In the case of the SPLIT routine, there are natural coordinate systems within the vectors and within the matrices (due to the transformation of Σ to 1), so the only ambiguity concerns the relative sizes of the scalar, vector, and matrix terms. These are handled by means of control parameters GAMMET, DELMET, and SGTMET, so that the square of the full derivative is

$$\text{GRADSQ} = \text{GAMMET}\left(\frac{\partial \text{obj}}{\partial c}\right)^2 + \text{DELMET}*\left(\frac{\partial \text{obj}}{\partial \delta}\right)^2 + \text{SGTMET}\left[\left(\frac{\partial \text{obj}}{\partial \sigma}\right)^2 + \left(\frac{\partial \text{obj}}{\partial \tau}\right)^2\right] \tag{54}$$

In the steepest-descents procedure, SPLIT first calculates the objective function at its new test point. If this is an improvement over the last test point's objective function, the derivatives are calculated and a steepest-descent step is taken to a new point using these derivatives. (This is always done at the first point.) If the new point is not an improvement, the derivative is not calculated, and the step size is made smaller than the step just made, and points backward along the old step. Conceptually, the new point is rejected in this case, and the program tries a new point in the same direction but closer to the previous origin point.

The step size is controlled by the variable SSIZ, using the temporary SHRINK. If the new point is an improvement (which is the change ratio of SSIZ), the step size is increased by an amount dependent on the expected and actual changes in the objective function, and bounded below by $\sqrt{\text{EXMNSQ}}$ and above by EXMAX, where EXMNSQ and EXMAX are control parameters. If the test point is not an improvement, the new point is taken at the minimum of the parabola defined by the test and old objective function values, and its derivative at the old point.

The iteration is terminated by a system using the variables PCTIMP, THIMP, and DOBFAC, and the control parameters DAMP, TIM0, and TIM1. Briefly, PCTIMP is a running average estimate of the fractional improvement in the objective function per iteration, where DAMP controls the length of the running average. THIMP is the THeoretical IMProvement for the step (not greater than the new objective function). After each iteration, the average improvement, PCTIMP*OBJ is compared with DOBPMS (Derivative of OBjective Per MilliSecond), and if it falls short of the required value, iteration is terminated. DOBPMS is calculated early in the program from the timing factors TIM0 and TIM1 and from the number of channels, since the cost of each iteration depends on the number of channels. The purpose of this procedure is to balance off computer time against the value of a better solution to SPLIT. There is also a direct limit to the number of iterations using the control parameter ITERMX.

After the iteration is complete, SPLIT rotates the solution back to the original coordinate system and builds the two new clusters. It was found necessary to enlarge the covariances of the two output subclusters to enable them to find the true distributions they were intended to match. This spreading of the clusters is necessary due to the "guess" character of SPLIT; otherwise, the generated clusters would be so far off the actual clusters they were intended to model that they would get no points, and would be eliminated eventually as having too small a proportion. Thus SPLIT adds to the covariance of each cluster an amount SPRED $(\Sigma_{ij} + 0.2\delta_i\delta_j)$, where SPRED is a control parameter.

The initialization of the steepest-descent procedure is based on a crude solution to the splitting problem. The system is transformed to a coordinate frame where the overall covariance Σ is the unit matrix. This is done by performing an eigen decomposition on Σ to diagonalize it, and then stretching or shrinking along each of the coordinate axes by the square root of the necessarily positive eigenvalues to make them 1. A rotation can diagonalize the kurtosis, leaving a unit covariance and a diagonal kurtosis. SPLIT uses this kurtosis-diagonal frame of reference.

For initialization, C is set to zero (actually 10^{-5}) and the skewness is disregarded to first order. The direction of splitting is along the most negative eigenvalue of KURT. (If there is none, the splitting is in covariance rather than in mean, e.g., the cluster has a sharp peak and a broad base.) The length of the vector in this direction is obtained by solving a cubic equation using both the skewness and kurtosis along that direction; the skewness equation is necessary to fix D^2 in the given direction. The remaining components of the displacement vector are calculated from the skewness components, and the covariance diagonals from them. The detailed equations used for the initialization do not appear to be available at present other than in the code itself; however, they were derived from table I directly as described. Some variables used in the initialization and their meanings are given in table IV.

SPLIT is a very crude routine, and could be much improved. It may also be a little slow for 16 channels, since many of the operations involved are cubic in the number of channels. This cubic behavior multiplied by the number of iterations can potentially be fairly expensive even though SPLIT is typically called only a few times during the execution of CLASSY. This can be regulated by the control parameters TIM0 and TIM1.

Since SPLIT only generates a guess, it is possible that a much wider solution will suffice. In fact, the initial guess used by SPLIT may be adequate, which could be tested by making comparison runs with ITERMX = -1. Tests with other small values might be profitable as well.

In any event, it should be possible to solve the equations used in SPLIT by a more ordinary, direct approach, rather than the somewhat roundabout steepest-descents method actually employed. The current version used this method only to make the coding direct; an earlier version of SPLIT was written using a more direct method of solution, but was judged too difficult to debug. It was essentially an extension of the initialization method used in the current SPLIT.

SPLIT currently divides a cluster into two components. However, analysis indicates that if the kurtosis has more than one negative eigenvalue, then the distribution, if made of normals, must have three or more components. No recognizance of this is made in the current code. Any later version should address this, or at least gather statistics to determine if the case is important.

TABLE III.— VARIABLES AND ACCUMULATORS USED IN SPLIT

FORTRAN variable	Meaning
GAM	c
GP	$(1 + c)/2$
GM	$(1 - c)/2$
AA	$g = \frac{1 - c^2}{4}$
BB	$h = \frac{3c^2 - 1}{2}$
TRD	$Tr(D^2)$
DELSQ	δ^2
R	$D^2\delta$
DUM	D^4 (temporary)
TMG	$Tr(D^2) - c\delta^2$
BBP	$h\delta^2 - cTr(D^2)$
GAM2	$2c$
GAMDEL	$c\delta^2$
ERCOV	E^2
ERSKEW	T^2
ERKURT	V^2
OBJ	objective = ERCOV*OBCOV + ERSKEW*OBSKEW + ERKURT*OBKURT
SPROA	$(TrD^2)\delta + 2D^2\delta - c\delta^2\delta$
DEL	δ
SG	σ
TAU	τ
ERE	σ^2 (temporary)
VER	τ^2 (temporary)
T	T

TABLE III.– Continued.

FORTRAN variable	Meaning
ERE	E (temporary)
VER	V (temporary)
GMCF	$1 + c^2$*GAMCEN
DKURT	g*OBKURT
DKRTGM	cg*OBKURT
DSKEW	g*OBSKEW
DD5	-2cg * OBKURT
ERED	$E\delta$
DSQT	D^2T
VDEL	$V\delta$
DUM	$\{V, D^2\}$ (temporary)
VDSQD	$\{V, D^2\}\ \delta$
TDEL	$T^t\delta$
DVDEL	$\delta^t V\delta$
TSPROA	T * SPROA
TVDSQ2	$Tr(VD^2)$
TPVD	gT * OBSKEW - 2cg$V\delta$ * OBKURT
DCOV2	2g OBCOV
D2	$2g\left[h\delta^t V\delta - \frac{c}{2} Tr(VD^2)\right]$ OBKURT
	- 2cg $T\delta$ OBSKEW
D3	$g(TrD^2 - c\delta^2)$ OBSKEW
D5	$2g(h\delta^2 - cTrD^2)$ OBKURT
D6	-4gc OBKURT
SG1	$\frac{1 + c}{2}$ OBCOV
TAU1	$\frac{1 - c}{2}$ OBCOV
UNIDSQ	$gT^t\delta$ OBSKEW + $g\left[\frac{1}{2} Tr(VD^2) - c\delta^t V\delta\right]$ OBKURT

TABLE III.– Concluded.

FORTRAN variable	Meaning
DD3	$g(TrD^2 - c\delta^2)$ OBKURT
DERED	$\delta^t E \delta$
DVD2D2	$\delta^t \{V, D^2\} \delta$
DDEL	$\partial obj/\partial \delta$
TEREDQ	$Tr(ED^2)$
TR2VD4	$2Tr(VD^4)$
VER	$\partial obj/\partial \sigma^2$ (temporary, note square)
ERE	$\partial obj/\partial \tau^2$ (temporary, note square)
DSG	$\partial obj/\partial \sigma$
DTAU	$\partial obj/\partial T$
DGAM	$\partial obj/\partial c$
SUMV	$(\partial obj/\partial \delta)^2$
SUMM	$(\partial obj/\partial \sigma)^2 + (\partial obj/\partial \tau)^2$
GRADSQ	(square of total derivative - see text)
GRADRT	$\sqrt{GRADSQ}$

TABLE IV.– INITIALIZATION VARIABLES USED IN SPLIT

Variable	Meaning
EVURT	Kurtosis eigenvalues
IBES	Index of most negative eigenvalue
AMXVAL	Most negative eigenvalue
TRN	$Tr(D^2)$
TRSQ	TRN**2
RT	Essentially $\sqrt{32\ EVURT}$
RTSM	$\Sigma\ RT + \left(\frac{16}{3}\ \frac{\lvert S_{IBES}\rvert}{\delta_{IBES}} - TRN\right)$
TCOF	d + 4 + (1/3)
ORTSM	$\Sigma\ \frac{1}{RT} - (\frac{1}{TRN})$ where in RTSM, TCOF, and ORTSM the term in parentheses appears only if there is a most negative eigenvalue
FRT	$\sqrt{\frac{32}{3}}$ AMXVAL
DELIN	The displacement along the most negative eigenvalue
DBES	D^2 along most negative eigenvalue
ERT	A new approximation to RT
BTR	Temporary coefficient
DELFAC	Temporary coefficient

3.5 DENCAL

This routine adjusts CIN and CTOT for a cluster to change the proportion for that cluster by a given ratio. This adjustment is necessary because of the notational convention that makes a cluster's proportion the product of its parent cluster's proportion with its proportion as calculated from its own proportion system (CIN and CTOT). If the proportion system design were changed to work with absolute proportions, DENCAL would be unnecessary. DENCAL also makes the mandatory calculations of the dependent and old variables DROP, OPROP, and ODEN.

The proportion of a cluster relative to its parent is calculated via

$$\text{PROP} = \frac{\text{CIN}}{W_F - \text{CTOT}}$$

where W_F is the weight of the parent cluster. This form is necessary to keep the proportions correct even if a cluster is skipped via the Monte Carlo system in STATIS.

DENCAL keeps CIN constant in making the proportion change, making only the changes required by the change in parent cluster and W_F.

If RATIO is the change in proportion, we have PROP' = RATIO*PROP or

$$\frac{\text{CIN}}{W_F' - \text{CTOT}'} = \text{RATIO}\ \frac{\text{CIN}}{W_F' - \text{CTOT}} \tag{55}$$

which becomes

$$\text{CTOT}' = W_F' - (W_F - \text{CTOT})/\text{RATIO} \tag{56}$$

All the variables correspond to those in the program, except W(KF) = W_F' , OLW = W_F.

3.6 APRIOR

APRIOR is a short routine which returns the natural logarithm of the additional cutoff factor needed for one more class, times any volumetric factors needed to normalize the integrals over the continuous parameters. The cutoff factor is a function C(m) of m, the number of classes, such that

$$\sum_{l=m}^{\infty} C(m) = 1 \tag{57}$$

It is necessary to multiply the overall probability by such a factor so it can be normalized even if the number of classes is unknown. Additional factors may be necessary to normalize various integrals over the continuous parameter space.

The results given by CLASSY do not appear overly sensitive to the value given to APRIOR, and this should be generally true except in the case of very statistically sensitive problems. As of this writing, no controlled study has been made of the effects of changing the values returned by APRIOR. It is possible that redefining the clustering problem handled by CLASSY without the use of the Bayesian model approach would clarify the range of values allowed for APRIOR without getting the one cluster per point (or n clusters for n^2 points) divergences in the clustering behavior of the program.

At present, APRIOR returns a value of VFAC*MQ + BIAS, where VFAC is a control parameter giving the dimensionality dependent volumetric factors (as a logarithm), and BIAS gives the logarithm of the overall cluster cutoff factor:

$$C(m) = (\text{constant})\ \exp(-m*\text{BIAS}) \tag{58}$$

It is possible that VFAC, the dominant term, could be made as high as -log 2, or even -(log 2)/MQ, since when a cluster is split, the subclusters have

effectively twice their own volume for their means to move around in. It would certainly be useful to test CLASSY with some fixed and presumably artificial data set, and find what values of VFAC and BIAS are required for the program to give anomalous behavior. Running close to these limits might make the algorithm more sensitive. It should be noted that the terms represented by APRIOR are quite small compared to those represented in the product probability over all the points, simply because there are so many more points than classes in most cases. Thus it is probable that the results are almost entirely independent of APRIOR, as they generally should be.

3.7 ISPLIT

ISPLIT is a short logical function used during the mapping and output stages of the program. Although CLASSY naturally uses fractional assignment of each point to a number of classes, during the mapping stage a decision must be made as to which single class each pixel should be assigned. A decision must also be made as to whether to use a parent cluster or its subclusters; this latter decision is the function of ISPLIT.

ISPLIT returns .TRUE. for a cluster if the cluster has subclusters, and either the likelihood ratio favors the subclusters, or the subclusters are older than the parent cluster. Otherwise ISPLIT returns .FALSE. The second proviso concerning the age of the cluster is necessary to avoid selecting newly JOINed parent clusters over their subclusters. Such parents have an advantage over their subclusters due to the APRIORi factor in SPFAC, and would be automatically selected even if the subclusters were a much better fit to the data. Thus the second proviso in ISPLIT forces a decision in favor of the subclusters of a JOIN until the new parent cluster has succeeded in eliminating the subclusters.

3.8 EIGROT

EIGROT is a general eigenvector-eigenvalue routine for symmetric matrices used primarily by SPLIT. It calls system routines, and is thus computing-system dependent. The routines used for EIGROT must handle the case of equal eigenvalues correctly.

4. CONCLUSIONS

4.1 TIMING AND OPTIMIZATION

Although no detailed work has yet been done on the timing and optimization characteristics of CLASSY, a few of the main features are obvious. This analysis will not examine the time consumed in scrambling the data, but only the timing of the CLASSY algorithm. The scrambling time should be measured, but any reasonable coding should make it much less than that consumed by CLASSY itself. The main timing at present should be the product of number of points times number of channels squared times number of clusters times number of iterations. The quadratic dependence on the number of channels should be noted in particular. CLASSY was originally designed as a research program, therefore many optimizing features were omitted.

The first loop of STATIS is executed for every point times every cluster. The second loop is executed probably two or three times per data point. ADJUST is called once per WADJ interval, typically every 100 or 200 points, and the discrete transformation subroutines are called even less often, typically a few times per run. The dominant factors in both the first and second loops of STATIS are quadratic in the number of channels. ADJUST contains some operations which are cubic in the number of channels (matrix inversion, etc.) but these are executed infrequently and therefore should consume little machine time. With the possible exception of SPLIT, the discrete transformations consume negligible machine time.

It can be seen then, that perhaps 80 percent of more of the execution time for CLASSY is spent in the first loop, and of that time perhaps one-half (more for 16 channels) is spent in the routine DOTSQ. Rewriting that routine as a straight line (no indexing) assembly routine could speed up the algorithm by 30 percent or more. Expanding all the vector routines and writing a low-precision assembly language routine for the function XP should double the speed of the system.

It is apparent that CLASSY is expending a great deal of effort in calculating the probabilities of points in classes in which they have very low probabilities. If these clusters could be eliminated from consideration without introducing a bias, the performance of CLASSY should be improved by a large factor. The second-loop processing of these clusters is eliminated by the Monte Carlo system. By moving parts of that system into the first loop, or making similar changes, timing could be considerably improved.

One simple method of eliminating the quadratic form evaluation for every cluster is to maintain a lower bound of the form relative to some standard quadratic form which is evaluated only once. Or, better, the system could be rotated to the frame where this form is unity. The only alteration that this would require would be to the Shepherd's correction terms in ADJUST; the non-invariant JOIN selection would be improved. Then, since the distance to a point would be always greater than $(x - \mu)^2$ relative to the standard form, most of the expensive quadratic form evaluation could be eliminated. This timing could be improved even more if techniques dividing the feature space were used. Additionally, the bounds for a cluster could take into account those of its subclusters, so that they could be eliminated with the main cluster. In any event, the extra overhead for maintaining the bound value should be small, and probably could be included in ADJUST rather than the updating portion of STATIS. Together with the optimized coding of the inner routines, this quick classification scheme might accelerate CLASSY by as much as a factor of 10. The other principal way to speed up CLASSY is to reduce the number of iterations necessary to reach a satisfactory solution.

CLASSY currently takes a large, single cluster of low weight for its initial condition. If a run sufficiently similar to the given run is available, a spread out version of that run's final clusters could be used. However, in the absence of a good signature extension this might not lead to significant improvement. A more general procedure might be to make several iterations using a subset or linear combination of the channels, and then switching to the full set for perhaps two passes. The two modes would be bridged by a

pass or a fraction of a pass using the old set of features to calculate the probabilities, and the new set in updating the class statistics, basically a variant of iterate mode. This technique should be short and easy to implement and should nearly double CLASSY's speed on the 16-channel problem. (Note that WADJ must grow quite rapidly to go around the whole data set by the second pass.)

The three major techniques just described would be quite simple to implement, and should speed CLASSY up by a factor of 15 to 20. Minor techniques for optimization might improve it by 40 percent more.

Some minor techniques for optimization include:

1. Making WADJ a function of cluster stability or rate of change
2. Variation of PLIM and MONTE in the Monte Carlo system
3. Finding good values for the acceleration parameters in ADJUST
4. Improving the proportion system
5. Newton's method for the convergence of the basic likelihood equations or Newton's method corrections might be applied for overlapping clusters (This is probably not warranted in the present version.)
6. Better methods for guessing the component distributions after a SPLIT
7. The selection of points highly dependent on the parameters to be placed on a temporary file and reprocessed intensively

4.2 MODIFICATIONS AND IMPROVEMENTS

Qualitative changes which might be made in the underlying model of CLASSY are described in this section. CLASSY imposes on the data an underlying structure of a mixture of normal distributions, probably not completely valid for Landsat data. In some cases CLASSY has given overlapping clusters as output, which, if they correspond to the same class, are a result of skewed or kurtotic distributions. Two simple modifications, allowing a skewness parameter and a radial function or "form factor" for each class as additional parameters

would eliminate these cases. Only the traceless part of the kurtosis would then be available to detect and organize the splitting of clusters. Coming closer to the real class of distributions appearing in Landsat data should improve the resolution of CLASSY and subsequent classification. The skewness might be fixed without modifying CLASSY itself by imposing a nonlinear coordinate transformation on the brightness space.

Another change in the model for CLASSY would allow "bridges" between clusters, to contain points which are themselves mixed. This technique might combine well with the Newton's method system, but is probably less cost-effective than simply doing a field recognition.

Numerous other modifications could be made to CLASSY. More complex models requiring fewer parameters per cluster are among the most general. Here clusters would be assumed to have equal covariances, sparse covariances, zero skewness, etc., until the statistics prove otherwise. However, the question of which type of model to use is complex and system dependent, and will not be explored further in this report.

Finally, the actual classification, or assignment of points to clusters, used in the final stage of CLASSY was added only for the purpose of obtaining readable maps. This point assignment procedure could be improved, if training data were available to label the clusters. In this case, the proper procedure would be to add the probabilities for clusters having a common label before selecting the label with the maximum probability.

5. REFERENCE

1. Duda, R. O.; and Hart, P. E.: Pattern Classification and Scene Analysis. John Wiley and Sons (New York), 1973.