T2

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

JSC-14871
MAY 1979

NASA CR-
160239

Ref: 642-7583
Contract NAS 9-15800
Job Orber 73-705

TECHNICAL MEMORANDUM

SOFTWARE REQUIREMENTS FOR THE STUDY OF CONTEXTUAL CLASSIFIERS AND LABEL IMPERFECTIONS

By

C. B. Chittineni

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

(NASA-CR-160239) SOFTWARE REQUIREMENTS FOR THE STUDY OF CONTEXTUAL CLASSIFIERS AND LABEL IMPERFECTIONS (Lockheed Electronics Co.) 18 p HC A02/MF A01 CSCL 09B

N79-27875
Unclas
G3/61 30470

May 1979

1. DESIGN OF CONTEXTUAL POSTPROCESSORS

This section describes the requirements for the design of contextual post-processors. Section 1.2 contains the requirements for spatially uniform context, and section 1.3 contains the requirements for sequential context. The following routine, which is common to both contexts, should be developed separately.

1.1 COMPUTATION OF POSTERIORI PROBABILITIES

From the discriminant functions, the posteriori probabilities needed by the contextual algorithms are computed as given below.

- Inputs:

  M — number of pattern classes

  ε — user-specified constant

  W_i — weight vector for class i

- Computation:

  $$g_i(X) = W_i^T X, \; i = 1, 2, \cdots, M$$

  $$g_{min} = \min_i g_i(X)$$

  $$p(\omega_i|X) = \frac{g_i(X) - g_{min} + \varepsilon}{\sum_{i=1}^{M} [g_i(X) - g_{min} + \varepsilon]}, \; i = 1, 2, \cdots, M$$

- Outputs:

  $p(\omega_i|X), \; i = 1, 2, \cdots, M$

1.2 SPATIALLY UNIFORM CONTEXT

The local neighborhood for the spatially uniform context is shown in the following figure.

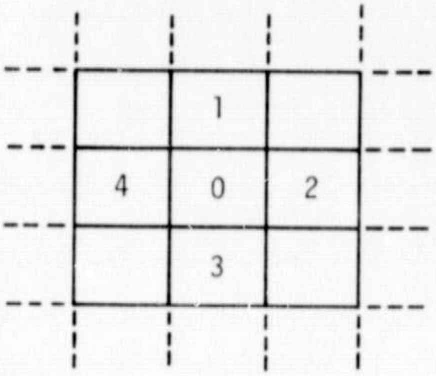

Figure 1.– Four neighbors of picture element (pixel) 0.

The posteriori probabilities of pixel 0 are updated using information from pixels 0, 1, 2, 3, and 4.

- Inputs:
  1. Merged data file
  2. Weight vectors file
  3. Number used in dividing the features
  4. Number of classes
  5. Number of features
  6. Locations of features
  7. Number and locations of training patterns in each class
  8. Number and locations of test patterns in each class
- Computation:

  a. Updating posteriori probabilities:

$$p(\omega_0 = i_0 | X_0, X_1, X_2, X_3, X_4)$$

$$= \frac{p(\omega_0 = i_0 | X_0)\left\{\prod_{j=1}^{4}\left[(1 - \Theta) + \Theta\,\frac{p(\omega_j = i_0 | X_j)}{P(\omega_j = i_0)}\right]\right\}}{\sum_{i_0=1}^{M} p(\omega_0 = i_0 | X_0)\left\{\prod_{j=1}^{4}\left[(1 - \Theta) + \Theta\,\frac{p(\omega_j = i_0 | X_j)}{P(\omega_j = i_0)}\right]\right\}},$$

$$i_0 = 1, 2, \cdots, M$$

b. Classification: Decide $X_0 \varepsilon \omega_0 = i_0$ if

$$p(\omega_0 = i_0 | X_0, X_1, \cdots, X_4) = \max_{\substack{i \\ i=1,\cdots,M}} p(\omega_0 = i | X_0, X_1, \cdots, X_4)$$

c. Crossvalidation with context: The code is developed for an error correction classifier and should be prepared as a subroutine. The necessary equations for the Fisher classifier are given in the Fisher classifier section and should be prepared as a subroutine.

d. Computation of Θ:

$$L(\Theta) = (1 - \Theta)^4 + \Theta(1 - \Theta)^3 A + \Theta^2(1 - \Theta)^2 B + \Theta^3(1 - \Theta) C + \Theta^4 D$$

$$A = \sum_{i_0=1}^{M} \frac{p(\omega = i_0 | X_0)}{P(\omega = i_0)} [p(\omega = i_0 | X_1) + p(\omega = i_0 | X_2) + p(\omega = i_0 | X_3)$$
$$+ p(\omega = i_0 | X_4)]$$

$$B = \sum_{i_0=1}^{M} \frac{p(\omega = i_0 | X_0)}{P^2(\omega = i_0)} [p(\omega = i_0 | X_1) p(\omega = i_0 | X_2) + p(\omega = i_0 | X_1) p(\omega = i_0 | X_3)$$
$$+ p(\omega = i_0 | X_1) p(\omega = i_0 | X_4) + p(\omega = i_0 | X_2) p(\omega = i_0 | X_3)$$
$$+ p(\omega = i_0 | X_2) p(\omega = i_0 | X_4) + p(\omega = i_0 | X_3) p(\omega = i_0 | X_4)]$$

$$C = \sum_{i_0=1}^{M} \frac{p(\omega = i_0 | X_0)}{P^3(\omega = i_0)} [p(\omega = i_0 | X_1) p(\omega = i_0 | X_2) p(\omega = i_0 | X_3)$$
$$+ p(\omega = i_0 | X_1) p(\omega = i_0 | X_2) p(\omega = i_0 | X_4)$$
$$+ p(\omega = i_0 | X_1) p(\omega = i_0 | X_3) p(\omega = i_0 | X_4)$$
$$+ p(\omega = i_0 | X_2) p(\omega = i_0 | X_3) p(\omega = i_0 | X_4)]$$

$$D = \sum_{i_0=1}^{4} \frac{p(\omega = i_0|X_0)}{P^4(\omega = i_0)} [p(\omega = i_0|X_1)p(\omega = i_0|X_2)p(\omega = i_0|X_3)p(\omega = i_0|X_4)]$$

Compute the coefficients A, B, C, and D and the roots of the equation $L'(\Theta) = 0$, where

$$L'(\Theta) = a\Theta^3 + b\Theta^2 + c\Theta + d$$

$$a = 4 - 4A + 4B - 4C + 4D$$

$$b = -12 + 9A - 6B + 3C$$

$$c = 12 - 6A + 2B$$

$$d = -4 + A$$

A subroutine the author has written for finding the roots of a polynomial should be used. For finding the optimal Θ, a subroutine should be developed from the flowchart given in figure 2 (from ref. 1).

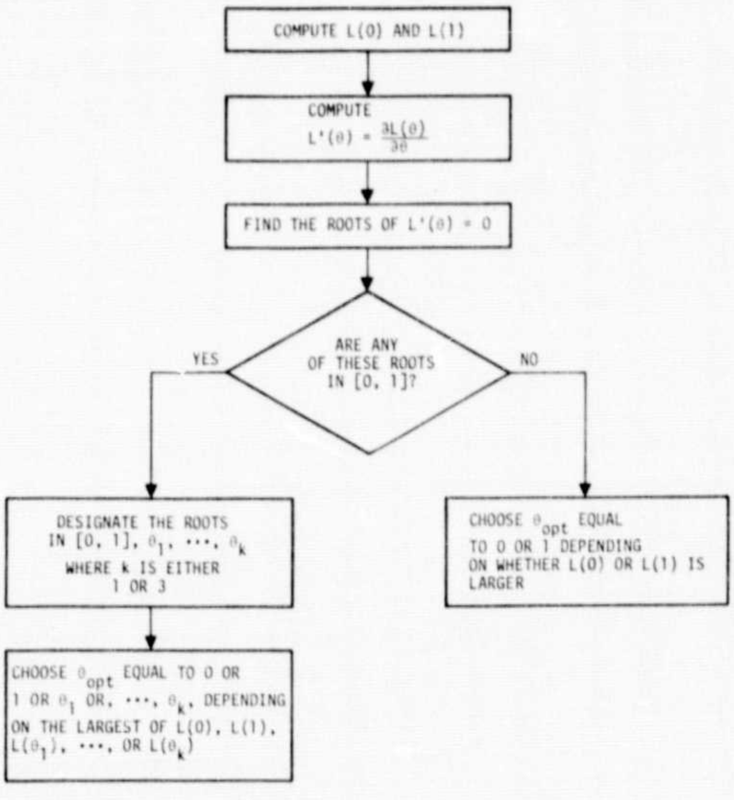

Figure 2.— Procedure for finding Θ_{opt} in the range $0 \leq \Theta_{opt} \leq 1$, which gives the global maximum of $L(\Theta)$. (From ref. 1.)

Classify the end pixels without using the context.

- Outputs:

   1. Classification map
   2. Proportion of classes in the segment from the classification
   3. Confusion matrix on the total dot set
   4. Confusion matrix on the test dot set
   5. Confusion matrix of crossvalidation with context

1.3 SEQUENTIAL CONTEXT

The local neighborhood of the pixel 0, for the use of sequential context, is shown in the following figure.

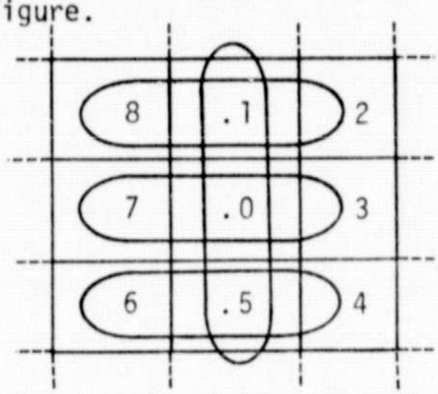

Figure 3.— Local neighborhood of pixel 0.

The posteriori probabilities of each central pixel in a circled group are updated using the neighborhood information. The exact equations are given below. First, the posteriori probabilities are updated for horizontal groups, and then the pixels of the vertical center column are used for updating those of pixel 0.

- Inputs:

   1. Merged data file
   2. Weight vectors file
   3. Number used in dividing the features
   4. Number of classes

5. Number of features

6. Locations of features

7. Number and locations of training patterns in each class

8. Number and locations of test patterns in each class

- Computation:

a. Updating posteriori probabilities

X_{n-1}, ω_{n-1}	X_n, ω_n	X_{n+1}, ω_{n+1}

Figure 4.— Horizontal neighbors of pixel n.

$$p(\omega_n = k|X_{n-1}, X_n) = \frac{(1 - \Theta)p(\omega_n = k|X_n) + \Theta \dfrac{p(\omega_n = k|X_n)p(\omega_{n-1} = k|X_{n-1})}{P(\omega_{n-1} = k)}}{\left[(1 - \Theta) + \Theta \displaystyle\sum_{i=1}^{M} \dfrac{p(\omega_n = i|X_n)p(\omega_{n-1} = i|X_{n-1})}{P(\omega_{n-1} = i)}\right]}$$

$$p(\omega_n = k|X_{n-1}, X_n, X_{n+1}) = \frac{p(\omega_n = k|X_{n-1}, X_n)\left[(1 - \Theta) + \Theta \dfrac{p(\omega_{n+1} = k|X_{n+1})}{P(\omega_{n+1} = k)}\right]}{\left[(1 - \Theta) + \Theta \displaystyle\sum_{i=1}^{M} \dfrac{p(\omega_{n+1} = i|X_{n+1})}{P(\omega_{n+1} = i)} p(\omega_n = i|X_{n-1}, X_n)\right]}$$

b. Classification: Same as in section 1.2

c. Crossvalidation: Same as in section 1.2

d. Computation of Θ: A separate Θ is computed for each row or column of pixels and used in updating the formulas. The computations required to be performed in computing the value of Θ for the neighborhood shown in figure 3 are given below:

$$L(\Theta) = (1 - \Theta)^2 + \Theta(1 - \Theta)A + \Theta^2 B$$

where

$$A = \sum_{i_n=1}^{M} \frac{p(\omega = i_n|X_n)}{P(\omega = i_n)} [p(\omega = i_n|X_{n-1}) + p(\omega = i_n|X_{n+1})]$$

$$B = \sum_{i_n=1}^{M} \frac{p(\omega = i_n|X_n)}{P^2(\omega = i_n)} [p(\omega = i_n|X_{n-1})p(\omega = i_n|X_{n+1})]$$

$$\Theta_1 = \frac{1}{2}\left[\frac{2 - A}{1 - A + B}\right]$$

For finding the optimal Θ, a subroutine should be developed from the flowchart given in figure 2.

- Outputs: Same as in section 1.2.
- Applicable document: reference 1

2. DESIGN OF THE FISHER CLASSIFIER

This section contains the requirements for the design of the Fisher classifier and the use of crossvalidation with this classifier. It is assumed that the patterns in the classes are arranged as in figure 5.

- Inputs:
  1. Name of data file
  2. Name of weight vectors file
  3. Number of classes
  4. Number of features
  5. Locations of features
  6. Number of training patterns in each class

- Computation:

a. Means and covariance matrices:

X_j^i = jth training pattern of class i

N_i = number of training patterns of class i

$$\left.\begin{aligned} \hat{m}_i &= \frac{1}{N_i} \sum_{j=1}^{N_i} X_j^i \\ \hat{\Sigma}_i &= \frac{1}{(N_i - 1)} \sum_{j=1}^{N_i} (X_j^i - \hat{m}_i)(X_j^i - \hat{m}_i)^T \end{aligned}\right\} i = 1, 2, \cdots, M$$

b. Weight vectors:

$g_i(X)$ = discriminant function of class i

$$g_i(X) = V_i^T X + v_i$$

$$\left.\begin{aligned} V_i &= \hat{S}_W^{-1} \hat{m}_i \\ v_i &= -\hat{m}_i^T \hat{S}_W^{-1} \frac{(\hat{m}_1 + \hat{m}_2 + \cdots + \hat{m}_M)}{M} \end{aligned}\right\} i = 1, 2, \cdots, M$$

$$\hat{S}_W = \hat{\Sigma}_1 + \hat{\Sigma}_2 + \cdots + \hat{\Sigma}_M$$

c. Classification: Same as section 1.2, b.

d. Formulas for crossvalidation: This section gives the expressions which are required to be implemented for crossvalidation. Expressions are given for the case in which a pattern X_k^1 from class ω_1 is left out. this computation is to be performed in a similar loop when a pattern comes on from the other class.

$$\hat{\Sigma}_1 = \frac{1}{(N_1 - 2)} \sum_{j=1}^{N_1} (X_j^1 - \hat{m}_1)(X_j^1 - \hat{m}_1)^T$$

$\hat{m}_i$, i = 1, 2, $\cdots$, M and $\hat{\Sigma}_i$, i = 2, $\cdots$, M are computed as in 2a.

$$\hat{S}_W = \hat{\Sigma}_1 + \hat{\Sigma}_2 + \cdots + \hat{\Sigma}_M$$

$$\left.\begin{aligned} V_i &= \hat{S}_W^{-1}\hat{m}_i \\ v_i &= -\hat{m}_i^T\hat{S}_W^{-1}\,\frac{(\hat{m}_1 + \hat{m}_2 + \cdots + \hat{m}_M)}{M} \end{aligned}\right\}, \; i = 1, 2, \cdots, M$$

$$\alpha = \frac{N_1}{(N_1 - 1)(N_1 - 2)}$$

$$Y(X_k^1) = \hat{S}_W^{-1}(X_k^1 - \hat{m}_1)$$

$$\beta(X_k^1) = (X_k^1 - \hat{m}_1)^T\hat{S}_W^{-1}(X_k^1 - \hat{m}_1)$$

$$\nu(X_k^1) = 1 - \alpha\beta(X_k^1)$$

$$d_1 = Y^T(X_k^1)\hat{m}_1$$

$$\begin{aligned} d_2 &= Y^T(X_k^1)\,\frac{(\hat{m}_1 + \hat{m}_2 + \cdots + \hat{m}_M)}{M} \\ &= Y^T(X_k^1)\hat{m} \end{aligned}$$

$$e_i = Y^T(X_k^1)\hat{m}_i, \; i = 2, \cdots, M$$

$$V_1(X_k^1) = V_1 + \frac{\alpha d_1}{\nu(X_k^1)}\,Y(X_k^1) - \frac{1}{(N_1 - 1)\nu(X_k^1)}\,Y(X_k^1)$$

$$\begin{aligned} v_1(X_k^1) = v_1 &- \frac{\alpha}{\nu(X_k^1)}\,d_1 d_2 + \frac{d_2}{(N_1 - 1)\nu(X_k^1)} + \frac{d_1}{2(N_1 - 1)} \\ &+ \frac{\alpha\beta(X_k^1)}{2(N_1 - 1)\nu(X_k^1)}\,d_1 - \frac{1}{2(N_1 - 1)^2}\,\frac{\beta(X_k^1)}{\nu(X_k^1)} \end{aligned}$$

$$V_i(X_k^1) = V_i + \frac{\alpha e_i}{\nu(X_k^1)}\,Y(X_k^1), \; i = 2, \cdots, M$$

$$v_i(X_k^1) = v_i - \frac{\alpha e_i d_2}{\nu(X_k^1)} + \frac{e_i}{2(N_1 - 1)\nu(X_k^1)}, \quad i = 2, \cdots, M$$

- Outputs:
  1. Weight vectors file
  2. Discriminant functions for the total pattern set
  3. Confusion matrix for the total dot set
  4. Confusion matrix for the training dot set
  5. Confusion matrix for the test dot set
  6. Crossvalidation discriminant functions
  7. Crossvalidation confusion matrix
- Applicable document: reference 2

3. CORRECTION OF LABEL IMPERFECTIONS WITH A LINEAR CLASSIFIER

This section describes the requirements for the identification of imperfectly labeled patterns with the linear classifiers.

- Inputs:
  1. Name of data file
  2. Name of weight vectors file
  3. Number of classes
  4. Number of features
  5. Parameters of imperfections — β_{11}, β_{12}, β_{21}, β_{22}
  6. Confidence level — α
- Computation:

$$W = W_1 - W_2$$

$$w_0 = w_{01} - w_{02}$$

$$\left.\begin{aligned} \hat{M}_i &= \frac{1}{N_i} \sum_{j=1}^{N_i} X_j^i \\ \hat{\Sigma}_i &= \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (X_j^i - \hat{M}_i)(X_j^i - \hat{M}_i)^T \end{aligned}\right\}, \; i = 1, 2$$

$$\left.\begin{aligned} m_i &= W^T\hat{M}_i + \omega_0 \\ \sigma_i^2 &= W^T\hat{\Sigma}_i W \end{aligned}\right\}, \; i = 1, 2$$

Find t_1 and t_2, such that

$$\alpha = \frac{1}{(\beta_{11}\beta_{22} - \beta_{12}\beta_{21})}\left[\beta_{22}P(\hat{\omega} = 1)\int_{-\infty}^{\frac{-t_2-m_1}{\sigma_1}} N(0, 1)d\zeta - \beta_{21}P(\hat{\omega} = 2)\int_{-\infty}^{\frac{-t_2-m_2}{\sigma_2}} N(0, 1)d\zeta + \beta_{11}P(\hat{\omega} = 2)\int_{-\infty}^{\frac{-t_1+m_2}{\sigma_2}} N(0, 1)d\zeta - \beta_{12}P(\hat{\omega} = 1)\int_{-\infty}^{\frac{-t_1+m_1}{\sigma_1}} N(0, 1)d\zeta\right]$$

Store the values of cumulative density of one-dimensional normal variables and find t_1 and t_2 by a one-dimensional search. For each X, implement the label correction scheme, $g(X) = W^TX + w_0$. Change the label of X to $\omega = 1$ if $g(X) > t_1$. Change the label of X to ω_2 if $g(X) < -t_2$. Do not change the label of X if $-t_2 \leq g(X) \leq t_1$.

- Outputs:
  1. Values t_1, t_2
  2. Original and corrected labels of the patterns
- Applicable document: reference 3

4. COMPUTATION OF THE PROPORTION OF A CROP OF INTEREST, VARIANCE OF PROPORTION OF A CROP OF INTEREST, AND THE VARIANCE REDUCTION FACTOR

This section defines the requirements for developing a program to compute the proportion of a crop of interest, variance of proportion of a crop of interest, and the variance reduction factor.

- Inputs:
  1. Number of classes
  2. Proportions of classes of a whole segment, if available, as classified by the classifier
  3. Confusion matrix of the total dot set
  4. Confusion matrix of the test dot set
- Computation:
  a. Take the first class as a crop of interest and then the proportion of the class of interest.

$$\hat{P}(W) = \sum_{i=1}^{M} P(W|i)P(i)$$

$\hat{P}(W)$ = estimated probability of the occurrence of wheat

$P(i)$ = classifier-estimated probability of occurrence of class i, either obtained from the total dot set or from the whole segment

$P(W|i)$ = given the classifier decision as class i, probability of occurrence of wheat obtained from the confusion matrix of the evaluation set (illustrated below)

  b. Variance of the proportion of the class of interest

$$\mathrm{Var}[\hat{P}(W)] = \frac{\sum_{i=1}^{M} P(i)P(W|i)[1 - P(W|i)]}{m}$$

m = number of samples in the evaluation set

c. Variance reduction factor

$$R = \frac{\sum_{i=1}^{M} P(i)P(W|i)[1 - P(W|i)]}{\hat{P}(W)[1 - \hat{P}(W)]}$$

d. Percentage of correct classification of the total dot set

e. Percentage of correct classification of the test dot set

f. Crossvalidation percentage of correct classification. Example:

Evaluation dot results						
Actual class (i)	Classified category (j)					Number belonging to each class
	1	2	3	...	M	
1	m_{11}	m_{12}	m_{13}	...	m_{1M}	$m_{1\cdot}$
2	m_{21}	m_{22}	m_{23}	...	m_{2M}	$m_{2\cdot}$
3	m_{31}	m_{32}	m_{33}	...	m_{3M}	$m_{3\cdot}$
⋮	⋮	⋮	⋮		⋮	⋮
M	m_{M1}	m_{M2}	m_{M3}	...	m_{MM}	$m_{M\cdot}$
Number classified into category	$m_{\cdot 1}$	$m_{\cdot 2}$	$m_{\cdot 3}$	...	$m_{\cdot M}$	$m_{\cdot\cdot} = m$

$$P(W|i) = \frac{m_{1i}}{m_{\cdot i}}$$

- Applicable document: reference 4

5. RANDOM SPLITTING OF DATA INTO TRAINING AND TEST SETS

This section contains the requirements for the random splitting of data into training and test sets according to user-specified proportions.

- Inputs:

  1. Number of classes

  2. Data set file

  3. Number of features

  4. Proportion of each class for the training set

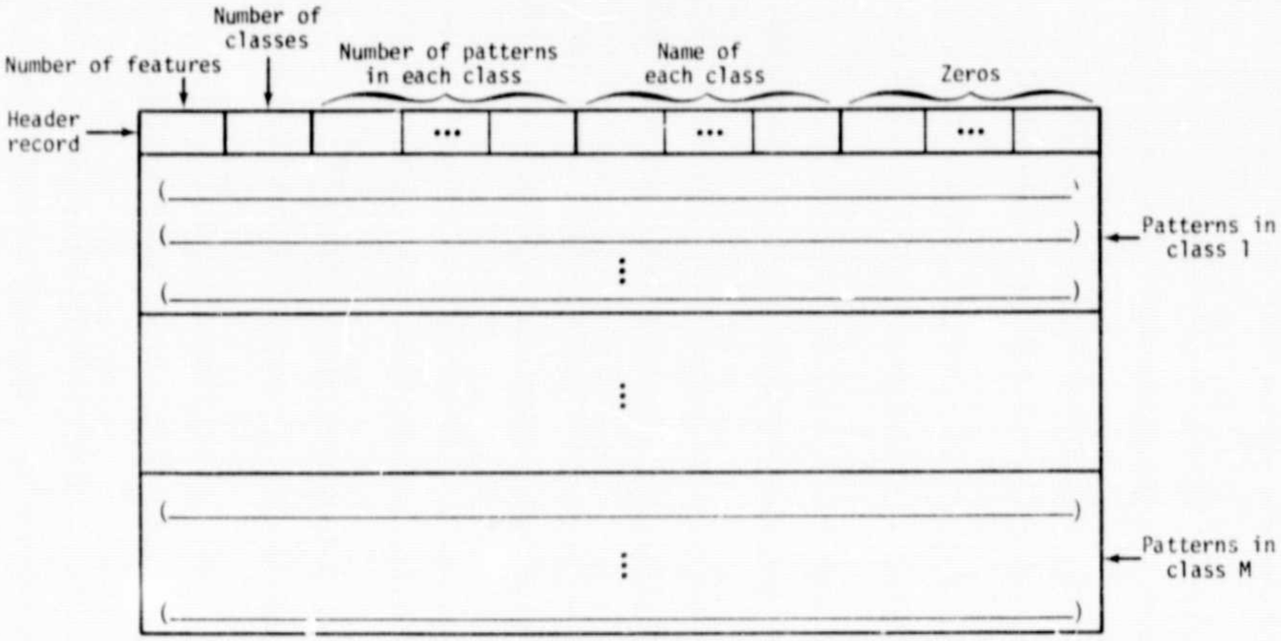

Figure 4.— Input data file.

- Outputs:

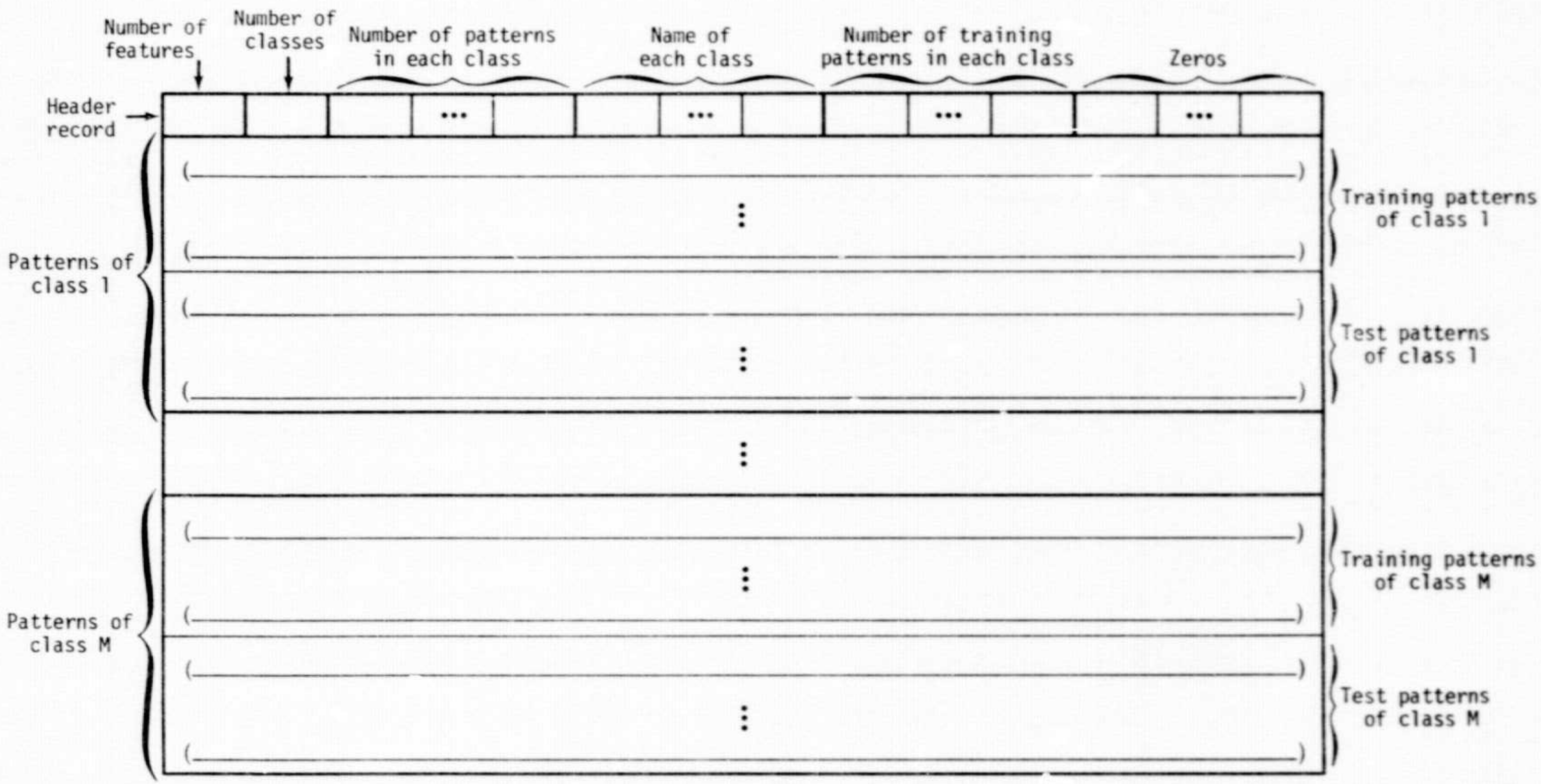

Figure 5.— Output data file.

6. REFERENCES

1. Chittineni, C. B.: Local Neighborhood Transition Probability Estimation and Its Use in Contextual Classification. LEC-13344, to be published.

2. Chittineni, C. B.: Fisher Classifier and Its Probability of Error Estimation. LEC-13301, to be published.

3. Chittineni, C. B.: Learning with Imperfectly Labeled Patterns. LEC-13068 (JSC-14867), Apr. 1979.

4. Chittineni, C. B.: Research Plan for Developing and Evaluating Classifiers. LEC-13300, to be published.