CAC Technical Memorandum No. 54

AN ILLIAC IV ALGORITHM FOR STATISTICAL CLASSIFICATION OF 8-CHANNEL MULTISPECTRAL IMAGE DATA

by

Martin Ozga and John Thomas

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

June 1975

Reported research was conducted in collaboration with staff of the Laboratory for Applications of Remote Sensing, Purdue University, Lafayette, Indiana 47907.

This work was supported by the National Aeronautics and Space Administration through Grant NGR 14-005-202 in cooperation with the EROS Program of the U. S. Department of Interior. Collaborative support was also provided by the Statistical Reporting Service of the U. S. Department of Agriculture.

1.0 Introduction

This paper is a verbal description of an ILLIAC IV program which classifies 8-channel multispectral image data into spectrally distinct categories. The program was written in ASK in 32-bit mode and resembles the 4-channel ILLIAC classifier as described in (1) with necessary changes to handle 8-channel data.

2.0 The Algorithm

The algorithm used follows closely the implementation of the maximum-likelihood Gaussian classification rule within the LARSYS software of the Laboratory for Applications of Remote Sensing (LARS) of Purdue University.[2,3]

In performing classification, the discriminant function for each of the N classes desired is evaluated for each data point. The point is classified into the category for which this evaluation yields the highest value.

The discriminant functions G_i are in the standard form for multivariate normal distributions.[2,3,4] For category i the function is

$$G_i(x) = b_i - \frac{1}{2} [(X-M_i)^T \Sigma_i^{-1} (X-M_i)]$$

where:

X = an eight-component vector of data

M_i = an eight-component mean vector whose elements are the mean values of intensities in the eight channels for category i

Σ_i = an 8 x 8 (symmetric) matrix whose elements are the variance and covariance of intensities in the eight channels for category i

$b_i = -\frac{1}{2} \log_e |\Sigma_i|$.

The vectors M_i and matrices Σ_i are produced by clustering programs which run on the ILLIAC IV[5] or, for small amounts of data, under the EDITOR[6] system on TENEX.

3.0 Files

The input and output files for the 8-channel classify program will be described here. For more information on file types and contents, see also (7).

3.1 Input Statistics File

The input statistics file contains the usual header of two long (ILLIAC) rows followed by the statistics. The type in ILLIAC word 0 is 5 to indicate that the variance-covariance matrices for the categories have been inverted. The number of categories out (in ILLIAC word 2) determines the number of categories to be used for classification. The statistics are stored in ILLIAC 32-bit floating-point format, two values per word. The statistics are stored by category and, within each category, consist of the eight mean values followed by the 36 values of the inverted variance-covariance matrix followed by the value of b_i (see Section 2 above). Only 36 values are needed to completely specify the matrix since it is 8 x 8 symmetric. The upper-triangular portion of the matrix is stored in order by rows. The statistics for two categories are stored in one ILLIAC row with the first category starting in word 0 and the second in word 24. The last 16 words of each row are not used. Thus, we have in each ILLIAC row:

ILLIAC words	Usage
0 - 3	mean values for category i
4 - 21	upper-triangular portion of inverted variance-covariance matrix for category i
22	value of b_i in inner part, outer part not used
23	not used
24 - 27	mean values for category i + 1
28 - 45	upper-triangular portion of inverted variance-covariance matrix for category i + 1
46	value of b_i in inner part, outer part not used
47 - 63	not used

Each inverted variance-covariance matrix V is stored in the order

$$V_{11}V_{12} \cdots V_{18}V_{22}\,{}_{23} \cdots V_{28}V_{33}V_{34} \cdots V_{38} \cdots V_{88}$$

where within a single word the order is inner, outer or that, for example, V_{11} is stored in the inner part of the word and V_{12} in the outer part.

3.2 Input Raw Data File

The input raw data file WINDOW consists of the usual two-row header followed by eight-channel raw data. The file may be either type 0 or type 1. In any case, the number of categories in the header of WINDOW is ignored since the number of categories is obtained from the input file COEFFS of statistics. WINDOW may be as large as necessary, constrained only by the size of the ILLIAC disk. The size of the file is determined by the program from the number of long rows, stored in ILLIAC word 4.

3.3 Output Classified File

The output classified file consists of the usual two-row header, with the file type set to 2, followed by the classified points. Each classified point occupies 16 bits with the category of the point, an integer from 1 to N (the number of categories), being right-adjusted in the field. A 16-bit field is used since future plans call for storage of chi-square probabilities (of correct classification) in the upper eight bits of the field.

4.0 Internal Data Structures

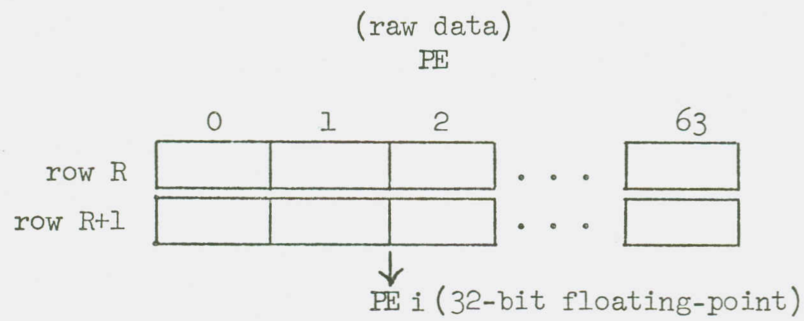
4.1 The Statistics, COEFF

The statistics are stored in the area COEFF. Within COEFF, each PE has the same value at the same address. Since operation is in 32-bit mode, the inner and outer parts of each word have the same 32-bit floating-point value. All statistics are stored in all PEs allowing elimination of routing during computation of the category to which a point belongs. Down each PE, the statistics for each category occupy 48 words. Assume that the first word for category i is at PE address A, then for each PE we have, in both the inner and outer parts of the word:

Word	Usage
A - A+7	mean values for category i
A+8 - A+43	upper-triangular portion of inverted variance-covariance matrix for category i
A+44	value of b_i
A+45 - A+47	not used

4.2 The Converted Raw Data, X

X is an eight-row vector of 32-bit floating point values. Since each point of raw 8-channel data occupies 64 bits, the converted data for two rows is stored in X. Only two rows at a time need be processed since, in classification, the category assigned to each point is independent of the value of each point and there are no iterations through the data, each discriminant function being evaluated once for each point. The scheme for converting raw data from rows R and R + 1 is then:



	Inner	Outer
X =	channel 1 of row R in PE i	channel 1 of row R+1 in PE i
	channel 2 of row R in PE i	channel 2 of row R+1 in PE i
	⋮	⋮
	channel 8 of row R in PE i	channel 8 of row R+1 in PE i

4.3 The Unpacked Output Categories, MINI

The category for each point is stored in MINI which occupies the same area of memory as the input raw data. The raw data is no longer needed once it has been converted to 32-bit floating-point. Each category is stored as a 32-bit integer so that one row of categories is needed for two rows of raw data. Before being written to disk, the categories are packed in the format described in Section 3.3.

5.0 Program Execution

5.1 Overview

First, the statistics are read in from disk, reformatted, and stored in PE memory. Next, comes a series of cycles until all the input data has been processed. In each cycle, some raw data is read in, converted to 32-bit floating-point, classified by application of the discriminant functions, and packed and written to the output file.

5.2 Reading in the Statistics

The first page of the statistics file COEFFS is read. The type is checked to be sure it is 5. If there are more than 28 categories, the second page is read. The statistics are broadcast to all the PEs to be stored as described in Section 4.2 by bringing eight words at a time into the CU, broadcasting them to all PEs, and then storing these into the appropriate locations in the memories of each PE.

5.3 Reading in the Raw Data and Writing Out the Classified Data

Since the amount of data which may be classified is only limited by the size of the ILLIAC disk and since the data need be only processed once (no iterations on the data being involved), the input is read and processed in segments of 28 ILLIAC pages. The 28-page size of a segment is determined by the memory available. Since each raw data point occupies 64 bits and each classified point only 16 bits, the output is only 1/4 as long as the input so only 7 ILLIAC pages are written on each cycle. A complication arises since the first ILLIAC page contains a two-row header which is not subject to the 1/4 reduction. Therefore, the first group of 28 pages contains 28 x 16 - 2 = 446 rows of raw data and the first group of 7 output pages contains 7 x 16 - 2 = 110 rows of classified data. Since 110 x 4 = 440, six rows of raw data are retained for the next cycle. At the end of each cycle except the last, the seven output pages are written to disk, the last six rows of raw data are moved to the top of the buffer, and the next 28 input pages are read into the buffer, following those six rows. On the last cycle, all remaining data is processed and written out to disk.

5.4 Performing the Classification

The discriminant functions for all categories as described in Section 2 are applied to all points, 128 points at a time. The points are expanded to vectors of 32-bit floating-point values and the category of the point is that for which the discriminant function yields the highest result. A slight complication arises in performing the matrix multiply since only the upper-triangular parts of the symmetric inverted variance-covariance matrices are stored. The problem is solved by using index sets, stored as a row of memory INDXSET. Eight values are needed for each row of the matrix for a total of 64. The values indicate the location of the element relative to the beginning of the statistics for that category. The values are brought into the CU as needed and then each is used as an index in all PEs to perform the computation.

5.5 Packing the Classified Points

Before they can be written to disk, the classified points must be packed so each occupies 16 bits and so that they are in the same order as in the original data. After the classification has been performed, the classified points are stored as 32-bit integers with the corresponding points from two rows stored in one 64-bit word, as mentioned in Section 4.3. To pack them, the classified points are first placed in their own rows. Then, in each row, a route of one to the left combines two elements in one word (with the elements in every other PE not useful, however). Finally, four rows must be combined into one. Each of the last three rows are routed to the starting position for data from that row (positions 16, 32, 48). Then, doing the four rows at a time, routing is used to position elements in the right PEs so they can be shifted and ORed to form the packed data.

Thus, starting with the four unpacked rows, we have:

	16	32	48
0			63
0			63
0			63
0			63

The last three are routed right by 16, 32, 48 respectively, thus:

	16	32	48
0			63
48 63	0		47
32	63	0	31
16		63	0 15

And finally all four are packed into one row:

0 15	16 31	32 47	48 63
Packed data for first row	Packed data for second row	Packed data for third row	Packed data for fourth row

by routing the same distance from each row on each of 16 cycles. Although this packing process is rather cumbersome, the data reduction attained is very important, particularly if the output must be retained on disk for a period of time for further analysis or must be sent over the ARPA Network to another site.

6.0 Operations

6.1 Files

As described in Section 3, there are three disk files required for the classification, the input files WINDOW (raw data) and COEFFS (statistics) and the output file WINDOX (classified data). WINDOW and COEFFS must occupy separate areas of disk, of course. WINDOX may occupy the same area of disk as WINDOW, but only under the restriction that WINDOX be written into areas from which raw data (WINDOW) has already been read. This can be assured by having both files start at the same location since only 7 pages of WINDOX are written for each 28 pages of WINDOW read.

The size of WINDOX in pages should be 1/4 the size of WINDOW (in pages) rounded up to the next higher integer plus one extra page to allow for header. Thus, if P is the length of WINDOW in pages, the length of WINDOX in pages is $(P + 3)/4 + 1$ where "/" indicates integer division with truncation. A more accurate computation (possibly leading to fewer pages) may be performed if the number of long rows R in WINDOW is known. The number of pages in WINDOX is then

$$(((R + 3)/4) + 2 + 15)/16$$

where the extra 2 is added in to account for the header and "/" once again indicates integer division with truncation.

6.2 Program Completion and Error Codes

A completion code is usually available by inspecting the contents of register TRO as displayed in the POF for the run. If the value is "A", the run was successful. If the value is "BAD" followed by some digit, an error is indicated. Current values and meanings of this final digit are:

Value	Error indicated
0	unable to read the first page of COEFFS, the statistics file
1	file type of COEFFS not equal to 5
2	unable to read the second page of COEFFS
3	unable to read the first page of WINDOW, the raw data file
4	file type of WINDOW other than 0 or 1
5	unable to read succeeding pages (after the first) of WINDOW
6	unable to write WINDOX, the output file of classified points.

In unusual cases, other errors may occur, usually leading to immediate termination, which will cause TRO to have values not in the pattern described above. The most common such error is not allocating enough ILLIAC disk space for a file. Other causes may be program error or hardware failure.

6.3 Timing

The elapsed time for the entire run is computed and displayed in the POF as the value of register IIA if the run terminated normally (with TRO = A). If the run terminated in an error, timing is not computed. The value of IIA displayed in the POF is a hexadecimal integer which is the total elapsed time in tenths of milliseconds (seconds x 10^4).

References

1. Thomas, J., "An ILLIAC IV Algorithm for Statistical Classification of ERTS-1 Imagery", CAC Technical Memorandum No. 18, Center for Advanced Computation, University of Illinois at Urbana-Champaign, Urbana, Illinois, May 1974.

2. Fu, K. S., Landgrebe, D. A., and Phillips, T. L., "Information Processing of Remotely Sensed Agricultural Data", Proceedings IEEE, Vol. 57, No. 4, April 1969, pp. 639-653.

3. Swain, P. H., "Pattern Recognition: A Basis for Remote Sensing Data Analysis", LARS Information Note 11572, The Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana, 1972.

4. Nilsson, Nils, Learning Machines, McGraw-Hill, New York, 1965.

5. Ozga, M., "An ILLIAC IV Algorithm for Cluster Analysis of 8-Channel Multispectral Image Data", CAC Technical Memorandum No. 53, Center for Advanced Computation, University of Illinois at Urbana-Champaign, Urbana, Illinois, May 1975.

6. Ray, R., Thomas, J., Donovan, W., Ozga, M., and Graham, M., "EDITOR, An Interactive ARPA Network - TENEX Interface to ILLIAC IV Multispectral Image Processing Systems", CAC Document No. 114, Center for Advanced Computation, University of Illinois at Urbana-Champaign, Urbana, Illinois, June 1974.

7. Thomas, J., "ERTS Data File Formats for ILLIAC IV - TENEX Processing", CAC Technical Memorandum No. 19, Center for Advanced Computation, University of Illinois at Urbana-Champaign, Urbana, Illinois, April 1974.