CR-128989

DEPARTMENT OF MATHEMATICS

UNIVERSITY OF HOUSTON HOUSTON, TEXAS

(NASA-CR-128989) [STATISTICAL STUDIES OF PATTERN CLASSIFICATION AND RECOGNITION, VOLUME 1] Final Report, 1 May 1972 - 30 Apr. 1973 (Houston Univ.) 164 p HC $10.25 CSCL 12A G3/19

N73-29566 THRU N73-29577 Unclas 07806

VOLUME I
FINAL REPORT NAS-9-12777
EARTH OBSERVATIONS DIVISION, JSC
MAY 1, 1972 - APRIL 30, 1973

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

Final Report[*]

NAS-9-12777

May 1, 1972 - April 30, 1973

Prepared for:

Earth Observations Division
Johnson Spacecraft Center
Houston, Texas

[*] Reports 1, 4, 15, 16, 25, 27 are not included in the final report. These reports will appear in the Final Report on NAS-9-12777, Mod 1S (with the exception of the Seminar Notes and SIEDS Recommendations).

UNIVERSITY OF HOUSTON
CULLEN BOULEVARD
HOUSTON, TEXAS 77004

DEPARTMENT OF MATHEMATICS

List of reports prepared under contract NAS-9-12777

1. Henry P. Decell, Jr. - Seminar Notes on Classification. May 1972

✓ 2. Henry P. Decell, Jr. - HYMPS-Numerical Techniques. May 1972

✓ 3. Henry P. Decell, Jr. and F. M. Speed - Differential Correction Schemes in Nonlinear Regression. Sept. 1972

4. Henry P. Decell, Jr. and C. L. Wiginton - SIEDS Recommendations. May 1972

✓1 & 2✓ ✓ 5. John Quirein - Divergence Considerations. Sept. 1972

✓ 6. Mary Ann Roberts - Pattern Recognition and the Potential Function. Sept. 1972

✓ 7. Terry Wilson - The Fuzzy Sets Approach to Pattern Recognition. Sept. 1972

✓ 8. L. H. Finch - Pattern Recognition and the Linear Discreminant Function. Sept. 1972

✓ 9. M. J. O'Malley - Linear Programming and Its Application to Pattern Classification. Sept. 1972

✓ 10. B. J. Barr - Cluster Seeking Techniques in Pattern Classification. June 1972

✓ 11. John Jurgensen - An Evaluation of An Algorithm for Linear Inequalities and Its Applications to Pattern Classification. Sept. 1972

✓ 12. John Quirein - Divergence and Necessary Conditions for Extremums. Nov. 1972

13. John Quirein - Sufficient Statistics: An Example. Jan. 1973

14. John Quirein - Sufficient Statistics for Divergence and the Probability of Misclassification. Nov. 1972

15. John Quirein - Admissible Linear Procedures and Threshholding. Jan. 1973

16. Robert Torres - On the Estimation of the Mean and Variance of Normal Populations from Cumulative Data. Sept. 1972

17. Wm. Morris, C. L. Wiginton, D. K. Lowell - SYMAT, COVAR - Test Procedures for Matrix Calculations. Oct. 1972

18. Jose O. Barrios - Nearest Neighbor Algorithms for Pattern Classification. Sept. 1972

19. Mary Ann Roberts - Computational Forms for the Transformed Covariance Matrix of Multivariate Normal Populations. Nov. 1972

20. James Leroy Hall - Perturbation and Sensitivity Inequalities in Divergence Calculations. March 1973

21. Henry P. Decell, Jr. - Rank-k Maximal Statistics for Divergence and Probability of Misclassification. Nov. 1972

22. Henry P. Decell, Jr. - On the Derivative of the Generalized Inverse of a Matrix. May 1972

23. Henry P. Decell, Jr. - Equivalence Classes of Constant Divergence and Related Results. Nov. 1972

24. Henry P. Decell, Jr. - An Expression for the Transformed Covariance Matrix of Multivariate Normal Populations. Nov. 1972

25. Dennison R. Brown - Matrix Representations of Semigroups (Title may be slightly changed on report) March 1973

26. Henry P. Decell, Jr., J. A. Quirein - An Iterative Approach to the Feature Selection Problem. March 1973

27. Mary Ann Roberts - Divergence and Householder Transformations. April 1973

N73-29567 1

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF HOUSTON HOUSTON, TEXAS

HYMPS-NUMERICAL TECHNIQUES

HENRY P. DECELL, JR.
MAY 1972

PREPARED FOR
EARTH OBSERVATION DIVISION , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

App. 1

HYMPS

Numerican Techniques

Report # 2

Contract NAS-9-12777

Henry P. Decell, Jr.
Department of Mathematics
University of Houston

May 1972

A-1-a

HYMPS

Numerical Techniques

Henry P. Decell, Jr.

General University of Houston

In the digital calculations that drive the classification portion of the Hybrid Pattern Recognition System (HYMPS) there are three items that warrant "tuning". They are:

I. Matrix Inversion

II. Det calculations & Singularity

III. Covariance Factorization

Although I,II,III are essentially viewed separately in the HYMPS writeup, they are, in fact, related and some improvement in numerical accuracy and computational speed can be gained by simply deleting redundant matrix manulipations.

Introduction

In what follows we will show that it is more economical to first factor the covariance matrix and, by so doing, delete the matrix inversion MINV. Necessary Det calculations can, moreover, be more easily realized by use of simple theoretical facts about the factorization.

Basically the reasons for doing the factorization first are:

1. MINV (or any other inversion routine, for that matter) can be eliminated in the current calculations

2. When MINV is deleted, errors in computation will be directly related to the factor routine and not to a combination of inversion factorization (unknown) errors.

3. All required information for classification is contained in the factorization.
4. The upper triangular form required in the analog classification scheme is preserved in these operations.
5. The B_k matrix calculations in their present form are no longer required.

Factorization

We recommend that the covariance matrix Σ be factored into "upper triangular form" i.e.

$$\Sigma = AA^T \quad \text{where}$$

A is a matrix with all zeros below the main diagonal (this is now being done to Σ^{-1} in HYMPS, after applying MINV to Σ). The results of this factorization will be as good as those obtained in factoring Σ^{-1} since we propose that the same factorization routine be utilized. In fact, the conditioning of Σ would produce factorization error since Σ^{-1} may well be garbage.

Now if $\Sigma = AA^T$ then

$$\Sigma^{-1} = (A^{-1})^T A^{-1} \quad \text{and}$$

since A is upper triangular so is A^{-1}.

We wish to compute the value of the classifier

$$f(x) = \frac{1}{(2\pi)^3 |\Sigma|^{1/2}} \exp - \frac{1}{2}(X-\bar{X})^T \Sigma^{-1} (X-\bar{X}).$$

If we let $Y = A^{-1}(X-\bar{X})$ then it is easy to see that the exponent Q is

$$Q = -\frac{1}{2}\{A^{-1}(X-\bar{X})\}^{T}\{A^{-1}(X-\bar{X})\}$$

$$= -\frac{1}{2} Y^{T}Y = -\frac{1}{2}\sum_{i=1}^{6} y_i^{2}$$

Hence if $Y = A^{-1}X$ then $AY = X$ and since A is upper triangular we can write the recursion formula for the y_i as follows. We do it in general, however, for HYMPS M=6

$$A_{mm}Y_m = X_m - \bar{X}_m$$

$$A_{m-1m-1}Y_{m-1} + A_{m-1m}Y_m = X_{m-1} - \bar{X}_{m-1}$$

$$A_{m-2m-2}Y_{m-2} + A_{m-2m-1}Y_{m-1} + A_{m-2m}Y_m = X_{m-2} - \bar{X}_{m-2}$$

$$\vdots$$

$$A_{11}Y_1 + A_{12}Y_2 + \cdots\cdots + A_{1m-1}Y_{m-1} + A_{1m}Y_m = X_1 - \bar{X}_1$$

In another form

$$Y_m = X_m - \bar{X}_m / A_{mm}$$

$$Y_{m-1} = \frac{X_{m-1} - \bar{X}_{m-1}}{A_{m-1m-1}} - \frac{A_{m-1m}}{A_{m-1m-1}} = \frac{1}{A_{m-1m-1}}\{X_{m-1} - \bar{X}_{m-1} - A_{m-1m}Y_m\}$$

$$Y_{m-2} = \frac{1}{A_{m-2m-2}}\{X_{m-2} - \bar{X}_{m-2} - A_{m-2m}Y_m - A_{m-2m-1}Y_{m-1}\}$$

$$\vdots$$

$$Y_1 = \frac{1}{A_{11}}\{X_1 - \bar{X}_1 - (A_{12}Y_2 + \cdots + A_{1m}Y_m)\}$$

In general,

$$Y_{m-k} = \frac{1}{A_{m-km-k}} \left\{ (X_{m-k} - \bar{X}_{m-k}) --- \sum_{j=1}^{k} A_{m-k,m-(j-1)} Y_{m-(j-1)} \right\}$$

Det Calculations

Since $\Sigma = AA^T$ it follows that:

$$\begin{aligned} \det\Sigma = \det(AA^T) &= (\det A)(\det A^T) \\ &= (\det A)(\det A) \\ &= (\det A)^2 \end{aligned}$$

Since A is upper triangular, its eigenvalues are the diagonal elements of A. Moreover, the det of <u>any</u> matrix is the product of its eigenvalues so that

$$\det A = \prod_{i=1}^{m} A_{ii}$$

Hence

$$\det\Sigma = (\det A)^2 = \left(\prod_{i=1}^{m} A_{ii} \right)^2$$

$$\therefore \det\Sigma = \prod_{i=1}^{m} A_{ii}{}^2,$$

an easy by-product of the factorization independent of MINV.

Singularity Evaluation

In the divergence calculations one should avoid concluding that Σ is "near singular" if $\det\Sigma \doteq 0$.

This is a classical misunderstanding of the theorem which states:

"Σ is singular if and only if det$\Sigma = 0$"

The misunderstanding arises by assuming a similar (however meaningless) theorem, namely,

"Σ is near singular if and only if det$\Sigma \doteq 0$"

The fact of the matter is that there does not exist a concept of "near singular" in matrix theory. The term "near singular" applies to numerical difficulties one may encounter in inverting matrices and is in no way related to whether or not Σ is in fact singular.

Consider the example (3 × 3)

$$A = \begin{pmatrix} 10^{-6} & 10^{-6} & 10^{-6} \\ 0 & 10^{-6} & 10^{-6} \\ 0 & 0 & 10^{-6} \end{pmatrix}$$

$$\det A = (10^{-6})^3 = 10^{-18} \doteq 0$$

Yet A is neither singular nor numerically difficult to invert.

Note: The fact that the example is "upper triangular is of no particular consequence except that det A is easy to calculate by inspection.

In fact for this A

$\Sigma = AA^T$ is symmetric and positive definite (See page 6-7)

yet det$\Sigma = 10^{-36} \doteq 0$.

N73-29568

1

UNIVERSITY OF HOUSTON · Founded 1927 ·

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF HOUSTON HOUSTON, TEXAS

DIFFERENTIAL CORRECTION SCHEMES IN NONLINEAR REGRESSION

HENRY P. DECELL, JR. AND
F. M. SPEED
SEPT. 1972

PREPARED FOR
EARTH OBSERVATION DIVISICN , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

B-1.

DIFFERENTIAL CORRECTION SCHEMES IN NONLINEAR REGRESSION

Report # 3

Contract NAS-9-12777

by

Henry P. Decell, Jr.
Department of Mathematics
University of Houston

F. M. Speed
Department of Mathematics
Texas A & I University

September 1972

B-1-a

DIFFERENTIAL CORRECTION SCHEMES IN NONLINEAR REGRESSION

Henry P. Decell, Jr.
University of Houston, Houston, Texas

F. M. Speed
Texas A & I University

Abstract

This paper briefly reviews and improves upon classical iterative methods in nonlinear regression. This is accomplished by discussion of the geometrical and theoretical motivation for introducing modifications using generalized matrix inversion, other than but in the same general vein as those discussed by Fletcher [6]. Examples having inherent pitfalls described in [8], [12] and others are presented and compared in terms of results obtained using classical and modified techniques. The modification is shown to be useful alone or in conjunction with other modifications appearing in the literature.

Introduction

Following for convenience the notation of [8], let y_t denote a set of n responces of the form

$$y_t = f_t(\theta) + e_t , \qquad t = 1,\dots,n$$

where the response function $f_t(\theta)$ is a known function of t and an undetermined vector $\theta = (\theta_1,\dots,\theta_p)$. We will call the vector $\hat{\theta}$ a least-squares estimate (given the n responses) of θ provided $\hat{\theta}$ minimizes

$$Q(\theta) = \sum_{t=1}^{n} (y_t - f_t(\theta))^2 .$$

The vectors are defined

$$Q'(\theta) = \frac{\partial(Q(\theta))}{\partial\theta_i}$$

$$R(\theta) = (y_t - f_t(\theta))$$

and the matrices

$$f'(\theta) = \left(\frac{\partial(f_t(\theta))}{\partial\theta_i} \right)^T$$

$$Q''(\theta) = \frac{\partial(\frac{\partial Q(\theta)}{\partial\theta_j})}{\partial\theta_i} .$$

Three of the most common differential correction schemes for estimating the parameter vector $\hat{\theta}$ are the steepest descent method, the quadratic approximation, and the Gauss-Newton method, with corrections respectively given by

$$\Delta\theta = -\alpha Q'(\theta) \ , \qquad \alpha > 0$$

$$\Delta\theta = -(Q''(\theta))^{-1}Q'(\theta)$$

$$\Delta\theta = -1/2(f'(\theta)^T f'(\theta))^{-1}Q'(\theta) \ .$$

These methods have their advantages and disadvantages. Of the three, the Gauss-Newton method is probably most popular.

The authors of [8] present a modification of a classical method and state that "The step $\Delta\theta$ will in general be distinct in both length and direction for each of the three methods." This is not necessarily the case from a computational point of view since the matrices to be inverted may be, for all practical computational purposes, singular; yet the system of equations may have infinitely many solutions. For example, the Gauss-Newton correction requires the solution of the equation

$$f'(\theta)^T f'(\theta)\Delta\theta = f'(\theta)^T R(\theta)$$

since

$$-1/2Q'(\theta) = f'(\theta)^T R(\theta) \ .$$

It is known that any equation of this form (i.e., of the form $A^T Ax = A^T z$, the normal equations of the least-squares problem: minimize $(Ax-z)^T(Ax-z)$ given A and z) always has at least one solution and perhaps infinitely many. We will try to point out the significance and consequences of these solutions in terms of their relationship to differential correction schemes.

The Generalized Inverse

A few basic concepts regarding generalized inverses important to the development follow.

Theorem 1. The four equations $AXA = A$, $XAX = X$, $(AX)^* = AX$, and $(XA)^* = XA$ have a unique solution X for each complex m×n matrix A. This solution X is called the generalized inverse of A and is denoted by $X = A^+$.

This theorem is due to Penrose [10] and is equivalent to the apparently more geometric characterization of the generalized inverse of A which follows.

Theorem 2. The generalized inverse A^+ of A is the unique solution of the equations

$$AX = P_{R(A)}$$

$$XA = P_{R(X)}$$

where $P_{R(A)}$ and $P_{R(X)}$, respectively, denote the perpendicular projection operators on the range spaces (column spaces) of A and X.

In any case, it is easy to see that if A is square and nonsingular, then A^+ is the ordinary inverse of A. Much work has been done recently in the area of generalized matrix inversion, including theoretical developments and computational techniques, rendering it a very useful tool in matrix theory and applications. A rather exhaustive bibliography concerning applications of generalized inverses can be found in [2], [3], and [13]. We will not develop the details of the basic concepts, but rather state an important theorem regarding the solution of matrix equations in general.

Theorem 3. The matrix equation $AXB = C$ has a solution X if and only if $AA^+CB^+B = C$, in which case all solutions are given by

$$X = A^+CB^+ + S - A^+ASBB^+$$

where S is an arbitrary matrix having the dimensions of X.

The Equation $A^TAx = A^Tz$

As stated earlier, the Gauss-Newton method involves the solution of an equation of this type at each iteration. The following corollary to Theorem 3 will give some insight to a possible course of action one could take at those times during the iteration process when the matrix $f'(\theta)^Tf'(\theta)$ (or perhaps even a matrix such as $Q''(\theta)$ in another method requiring inversion for the calculation of the correction $\Delta\theta$) is actually or nearly singular. For the purpose of this paper, we will describe how generalized inversion can be useful in iterative techniques requiring the solution of equations of the form $A^TAx = A^Tz$.

Corollary 1. If A is any $m\times n$ matrix and z is any $m\times 1$ vector, then the equation $A^TAx = A^Tz$ has at least one solution and all solutions are given by

$$X = A^+z + (I - A^+A)y$$

where y is arbitrary having the dimensions of x.

The proof of Corollary 1 is an immediate consequence of Theorem 3 and fact that $(A^TA)^+A^T = A^+$ [10].

Corollary 2. Among the solutions of $A^TAx = A^Tz$, the solution $x = A^+z$ has the smallest Euclidean norm (henceforth "norm" will be denoted $||\cdot||$).

The proof of Corollary 2 follows from the facts that $I - A^+A$ is the orthogonal projection operator on the orthogonal compliment of the range space of A^+ and hence that A^+z and $(I - A^+A)y$ are orthogonal for every y. In fact,

$$||A^+z + (I - A^+A)y||^2 = ||A^+z||^2 + ||(I - A^+A)y||^2$$
$$\geq ||A^+z||^2 \quad .$$

The significance of Corollary 1 is that there may be infinitely many possible corrections $\Delta\theta$ satisfying an equation defining a differential correction scheme in the presence of a singular or, in the computational sense, nearly singular coefficient matrix. There is a tendency to disregard or remain unaware of these solutions and, with the inability to invert the coefficient matrix, to look for new or modified techniques such as those found in [1], [5], [8], [9], and [12]. For example, in [7] Jennrich and Sampson modify the coefficient matrix by selected rows and columns. In [8], Marquardt changes the diagonal of the coefficient matrix. It has been our experience that these solutions should be given careful attention in the case of what will hereafter be called an apparent (i.e., actual or computational) singularity.

Fletcher [6] points out that in the generalized least-squares (Gauss-Newton) or Newton methods "... A most important property of the generalized inverse formulation is that in all circumstances (i.e., full rank or not), even when the generalized least-squares method would fail, the directions of search generated are downhill and so an imporvement can always be made to the sum of squares (assuming the approximation is not already a stationary point)." In this connection, the significance of Corollary 2 is that there is a reasonable way to choose a correction $\Delta\theta$ satisfying the defining equations of the scheme whenever an apparent singularity occurs. We propose to choose the minimum Euclidean norm correction $A^{+}z$ (i.e., the correction of shortest length consistent with the correction equation). It has been our experience that in nonlinear equations other solutions can result in failure of convergence.

The suggested correction certainly depends upon the algorithm used to calculate A^+ and the actual computational way in which the algorithm establishes that A is not of full rank (i.e., A^TA singular). Of course, this is intimately connected with near-zero tests in the algorithm, sensitivity to dependent columns or rows, conditioning, and so forth. We should further point out that, for a general differential correction scheme of the form $M(\theta)\Delta\theta = z(\theta)$, the choice of the correction should be $\Delta\theta = M(\theta)^+z(\theta)$ if there is at least one solution for $\Delta\theta$. Of course, according to Theorem 3 there will be at least one and possibly infinitely many solutions $\Delta\theta$ if and only if $M(\theta)M(\theta)^+z(\theta) = z(\theta)$. Moreover, if there is one and only one solution, then that solution is indeed given by $\Delta\theta = M(\theta)^+z(\theta)$.

For example, in the Gauss-Newton method, $M(\theta) = f'(\theta)^Tf'(\theta)$ and $z(\theta) = f'(\theta)^TR(\theta)$ so that $\Delta\theta = M(\theta)^+z(\theta) = (f'(\theta)^Tf'(\theta))^+f'(\theta)^TR(\theta) = f'(\theta)^+R(\theta)$. Even if $M(\theta)$ is nonsingular, then $(f'(\theta)^Tf'(\theta))^+ = f'(\theta)^Tf'(\theta))^{-1}$, and either form of $\Delta\theta$ may be used in calculations:

$$\Delta\theta = (f'(\theta)^Tf'(\theta))^{-1}f'(\theta)^TR(\theta) = f'(\theta)^+R(\theta) .$$

In other words if $M(\theta)$ is square and computationally nonsingular, the classical correction is, in fact, the minimum norm correction. We will not discuss the comparative aspects of computing $\Delta\theta$ in a correction scheme such as the Gauss-Newton method by one or the other of the theoretically equivalent formulas:

$$(1)\quad \Delta\theta = (f'(\theta)^T f'(\theta))^+ f'(\theta)^T R(\theta)$$

$$(2)\quad \Delta\theta = f'(\theta)^+ R(\theta)$$

Calculations in our examples use (2).

We have had unusual success with this technique in many practical problems too numerous to mention here. In many cases, one definite advantage seems to be the ability to continue making corrections of reasonable length and perhaps, as in the Gauss-Newton case, reasonable direction through regions in which the coefficient matrix $M(\theta)$ behaves badly. We do not propose this technique as a cure-all but rather that it should be included among other useful techniques in nonlinear regression. A few examples having known pitfalls will be presented in the next section.

Examples.

In the following examples, the residual sum of squares $Q(\theta)$ will be presented in tables by iteration number. The values of $Q(\theta)$ for the methods cited will be those values tabulated in the references cited. Some authors divide $Q(\theta)$ by the degrees of freedom. For clarity and easy comparison we indicate this division in the tables when necessary. Finally, the residual sum of squares given by the method of this paper (minimum norm correction) will be noted MN; $Q(\theta)$.

Results of the method of this paper compared with those of the Modified Davidon Method (MDM) used in [12] to find the parameters of an exponential model discussed by Hartley in [7] are given in Table 1.

Table 1

Exponential Model (Hartley)

Iteration	MN; Q(θ)	MDM; Q(θ)
0	27376	27376
1	14586	20127
2	13779	15412
3	13408	13552
4	13394	13485
5	13390	13449
6		13425
7		13394
8		13393
9		13390

A second exponential model given by the authors of [8] points out a failure of Hartley's method [7] due to a singular partial derivative matrix. In [8] a stepwise regression scheme (SR) is successfully utilized for this example. The results of the (SR) scheme compared with those of the method of this paper are given in Table 2.

Table 2

Exponential Model - Singular Partials

Iteration	MN; $Q(\theta)/8$	SR; $Q(\theta)/8$
0	521.41	521.41
1	429.84	429.84
2	39.11	88.15
3	15.765	83.74
4	15.545	*
10		21.33
30		15.545

*The value of SR: $Q(\theta)$ was not tabulated in [8] for this iteration.

Another six-parameter exponential model having inherent singularity problems is presented in [12] using the Modified Davidon Method (MDM). A comparison of the results using the technique of this paper is given in Table 3.

Table 3

Six Parameter Exponential Model - Singular Partials

Iteration	MN; Q(θ)	MDM; Q(θ)
0	21.38	21.38
10	.873	2.39
20	.792	1.99
30	.396	1.77
40		1.59
50		1.41
60		.90
70		.41
80		.407

Concluding Remarks

We have taken the liberty to exclude a reproduction of the detailed description of our example models. These models are thoroughly treated in [7], [8] and [12]. The tables give some indication of rates of convergence and a comparison of residuals only. We do not wish to leave the impression that iteration counts are comparable. For example, one Gauss-Newton iteration could have been equivalent to p conjugate direction steps for the matrix inversion employing the Davidon method.

Acknowledgments

The authors wish to express thanks to F. C. Delaney [4] for providing the detailed calculations presented in the tables. We would also like to express our thanks to G. E. Taylor [12] for making available to us the data used in his computations for MDM Q(θ) in Table 3.

REFERENCES

1. W. C. Davidon, Variable metric method for minimization, A.E.C. Research and Development Report ANL-5990 (Rev.), (1959).

2. H. P. Decell, An alternate form of the generalized inverse of an arbitrary complex matrix, SIAM Rev., (3) 7 (1965), 356-358.

3. ____________ , An application of generalized matrix inversion to sequential least squares parameter estimation, NASA TN D-2830, (1965).

4. F. C. Delaney, Generalized inverse calculation subroutine, (GINV2), Catalog 125, Lockheed Electronics Col, Houston, Texas, (1968).

5. R. Fletcher and J. J. D. Powell, A rapidly convergent descent method for minimization, The Computer Journal, 6 (1963), 163.

6. ____________ , Generalized inverse methods for the best least squares solutions of systems of nonlinear equations, The Computer Journal, (1968), 392-399.

7. H. O. Hartley, The modified Gauss-Newton method for the fitting of nonlinear regression functions by least squares, Technometrics, 3 (1961), 269-280.

8. R. I. Jennrich and P. F. Sampson, Application of stepwise regression to nonlinear estimation, Technometrics, 10 (1968), 63-72.

9. D. W. Marquardt, An algorithm for the estimation of nonlinear parameters, Society for Industrial and Applied Mathematics Journal, 11 (1963), 431-441.

10. R. Penrose, A generalized inverse for matrices, Proc. Camb. Philos. Soc., 51 (1955), 406-413.

11. J. B. Rosen, The gradient projection method for nonlinear programming, Part I. Linear Constraints, Society for Industrial and Applied Mathematics Journal, 8 (1960), 181-217.

12. P. Vitale and G. Taylor, A note on the application of Davidon's method to nonlinear regression problems, Technometrics, 10 (1968), 843-849.

13. Thomas L. Boullion and Patrick L. Odell, Proceedings of the symposium on theory and application of generalized inverses of matrices, Mathematics Series No. 4, Texas Technological College, Lubbock, Texas (1968).

14. G. Taylor, Data for example stated in [11], Private Communication.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF HOUSTON HOUSTON, TEXAS

DIVERGENCE CONSIDERATICNS I
JOHN QUIREIN
SEPT. 1972

PREPARED FOR
EARTH OBSERVATICN CIVISICN , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

N73-29569 C-1-

DIVERGENCE CONSIDERATIONS I

by

J. A. Quirein
University of Houston
Department of Mathematics

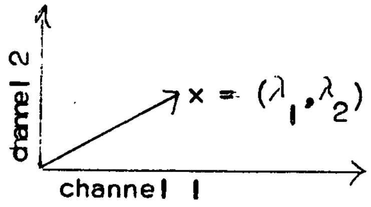
Report # 5

Contract NAS-9-12777

September 1972

DIVERGENCE CONSIDERATIONS

Problem Statement: Let $\pi_1, \pi_2, \ldots, \pi_n$ be n distinct, normally distributed classes or populations of two dimensional response vectors $x = (\lambda_1, \lambda_2)$, where λ_i is a measurement of the relative response of x along channel i.



The problem is to determine the "best channel" in the sense of divergence and in the sense of minimizing the probability of misclassification.

Let Σ_i denote the sample covariance matrix for the ith class and suppose, after training, we find that $\Sigma_i = I$, $i = 1,2, \ldots, n$. Let μ_i be the mean associated with the ith population; then it is easily verified that the interclass divergence is:

$$D(i,j) = (\mu_i - \mu_j)^2 \geq 0$$

The density function for the ith class is:

$$P_i(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x - \mu_i)^2}$$

It is useful at this time to consider the partitions of a given channel axis determined by the maximum likelihood solution of the Bayes discriminate problem. Recall in this case x is assigned to π_k if:

$$\ln P_k(x) = \max \left\{ \ln P_i(x),\ i = 1,2, \ldots, n \right\}$$

Under our assumptions that $\Sigma_i = I$, it is easily verified this becomes x is assigned to π_k if:

$$(x - \mu_k)^2 = \min \left\{ (x - \mu_i)^2,\ i = 1,2, \ldots, n \right\}$$

We shall assume $\mu_i < \mu_{i+1}$ for $i = 1, \ldots, n - 1$.

Now note

$$(x - \mu_{i+1})^2 = \left[(x - \mu_i) + (\mu_i - \mu_{i+1})\right]^2$$
$$= (x - \mu_i)^2 + 2(\mu_i - \mu_{i+1})(x - \mu_i) + (\mu_i - \mu_{i+1})^2$$

so that $(x - \mu_{i+1})^2 \geq (x - \mu_i)^2$ whenever $2(\mu_i - \mu_{i+1})(x - \mu_i) + (\mu_i - \mu_{i+1})^2 \geq 0$, that is, whenever $x \geq \frac{1}{2}(\mu_i + \mu_{i+1})$.

Thus associated with each mean μ_i, we have a region R_i such that if $x \in R_i$, x is assigned to the π_i population where the R_i's are defined as follows:

$$R_1 = \left\{x:\ x \leq \tfrac{1}{2}(\mu_1 + \mu_2)\right\}$$

$$R_i = \left\{x:\ \tfrac{1}{2}(\mu_{i-1} + \mu_i) \leq x \leq \tfrac{1}{2}(\mu_i + \mu_{i+1})\right\} \quad i = 2, \ldots, n-1$$

$$R_n = \left\{x:\ \tfrac{1}{2}(\mu_{n-1} + \mu_n) \leq x\right\}$$

Now consider the n-class problem with equal a priori probabilities $q_i = \frac{1}{n}$, the cost of misclassifying an individual from population π_j as being from population π_i, $C(i|j) = 1$, and the probability that x belong to R_j given that the individual is from π_i, $P(j|i,R) = \int_{R_j} P_i(x)dx$, then the cost of misclassification to be expected totally is:

$$Q(R) = \sum_{i=1}^{n} q_i \left\{ \sum_{\substack{j=1 \\ j\neq i}}^{n} C(j|i)P(j|i,R) \right\} = \frac{1}{n} \sum_{i=1}^{n} \left[\sum_{\substack{j=1 \\ j\neq i}}^{n} \int_{R_j} P_i \, dx \right]$$

For $i = 1$, $$\sum_{j=2}^{n} \int_{R_j} P_i dx = \int_{\bigcup_{j=2}^{n} R_j} P_1 \, dx = \frac{1}{\sqrt{2\pi}} \int_{\frac{1}{2}(\mu_1+\mu_2)}^{\infty} e^{-\frac{1}{2}(x-\mu_i)^2} dx$$

$$= \frac{1}{\sqrt{2\pi}} \int_{\frac{1}{2}(\mu_2-\mu_1)}^{\infty} e^{-\frac{1}{2}y^2} dy$$

$$= \frac{1}{\sqrt{2\pi}} \int_0^{\infty} e^{-\frac{1}{2}y^2} dy - \frac{1}{\sqrt{2\pi}} \int_0^{\frac{1}{2}(\mu_2-\mu_1)} e^{-\frac{1}{2}y^2} dy$$

$$= \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \int_0^{\frac{1}{2}\sqrt{D(2,1)}} e^{-\frac{1}{2}y^2} dy$$ since by symmetry of $e^{-\frac{1}{2}y^2}$

we have $$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}y^2} dy = 2 \frac{1}{\sqrt{2\pi}} \int_0^{\infty} e^{-\frac{1}{2}y^2} dy = 1.$$

When i = n we again use the symmetry of the function so that

$$\sum_{j=1}^{n-1}\int_{R_j} P_i\,dx = \int_{\overset{n-1}{\underset{j=1}{\bigcup}} R_j} P_i\,dx = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\frac{1}{2}(\mu_n+\mu_{n-1})} e^{-\frac{1}{2}(x-\mu_n)^2}dx$$

$$= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\frac{1}{2}(\mu_{n-1}-\mu_n)} e^{-\frac{1}{2}y^2}\,dy$$

$$= \frac{1}{\sqrt{2\pi}}\int_{\frac{1}{2}(\mu_n-\mu_{n-1})}^{\infty} e^{-\frac{1}{2}y^2}\,dy$$

$$= \frac{1}{2} - \frac{1}{\sqrt{2\pi}}\int_{0}^{\frac{1}{2}\sqrt{D(n,n-1)}} e^{-\frac{1}{2}y^2}\,dy$$

When 1 < i < n

$$\sum_{\substack{j=1\\ j\neq i}}^{n}\int_{R_j} P_i\,dx = \sum_{j=1}^{i-1}\int_{R_j} P_i\,dx + \sum_{j=i+1}^{n}\int_{R_j} P_i\,dx$$

$$= \int_{\overset{i-1}{\underset{j=1}{\bigcup}} R_j} P_i\,dx + \int_{\overset{n}{\underset{j=i+1}{\bigcup}} R_j} P_i\,dx$$

$$= \frac{1}{\sqrt{2\pi}}\left[\int_{-\infty}^{\frac{1}{2}(\mu_{i-1}+\mu_i)} e^{-\frac{1}{2}(x-\mu_i)}\,dx + \int_{\frac{1}{2}(\mu_i+\mu_{i+1})}^{\infty} e^{-\frac{1}{2}(x-\mu_i)}\,dx\right]$$

$$= \frac{1}{\sqrt{2\pi}}\left[\int_{-\infty}^{\frac{1}{2}(\mu_{i-1}-\mu_i)} e^{-\frac{1}{2}y^2}\,dy + \int_{\frac{1}{2}(\mu_{i+1}-\mu_i)}^{\infty} e^{-\frac{1}{2}y^2}\,dy\right]$$

$$= \frac{1}{\sqrt{2\pi}}\left[\int_{\frac{1}{2}(\mu_i-\mu_{i-1})}^{\infty} e^{-\frac{1}{2}y^2}\,dy + \int_{\frac{1}{2}(\mu_{i+1}-\mu_i)}^{\infty} e^{-\frac{1}{2}y^2}\,dy\right]$$

$$= 1 - \frac{1}{\sqrt{2\pi}} \int_0^{\frac{1}{2}\sqrt{D(i,i-1)}} e^{-\frac{1}{2}y^2}\, dy - \int_0^{\frac{1}{2}\sqrt{D(i+1,i)}} e^{-\frac{1}{2}y^2}\, dy$$

Finally we have the total cost of misclassification is

$$Q(R) = \frac{1}{n}\Bigg[\frac{1}{2} - \frac{1}{\sqrt{2\pi}}\int_0^{\frac{1}{2}\sqrt{D(2,1)}} e^{-\frac{1}{2}y^2}\, dy + \sum_{i=2}^{n-1}\Bigg(1 - \frac{1}{\sqrt{2\pi}}\int_0^{\frac{1}{2}\sqrt{D(i,i-1)}} e^{-\frac{1}{2}y^2}\, dy$$

$$- \frac{1}{\sqrt{2\pi}}\int_0^{\frac{1}{2}\sqrt{D(i+1,i)}} e^{-\frac{1}{2}y^2}\, dy\Bigg) + \frac{1}{2} - \frac{1}{\sqrt{2\pi}}\int_0^{\frac{1}{2}\sqrt{D(n,n-1)}} e^{-\frac{1}{2}y^2}\, dy\Bigg]$$

$$= \frac{1}{n}\Bigg[(n-1) - \frac{1}{\sqrt{2\pi}}\Bigg\{\sum_{i=2}^{n}\int_0^{\frac{1}{2}\sqrt{D(i,i-1)}} e^{-\frac{1}{2}y^2}\, dy + \sum_{i=1}^{n-1}\int_0^{\frac{1}{2}\sqrt{D(i+1,i)}} e^{-\frac{1}{2}y^2}\, dy\Bigg\}\Bigg]$$

So $$Q(R) = \frac{1}{n}\Bigg[(n-1) - \frac{2}{\sqrt{2\pi}}\sum_{i=1}^{n-1}\int_0^{\frac{1}{2}\sqrt{D(i+1,i)}} e^{-\frac{1}{2}y^2}\, dy\Bigg] \qquad (1)$$

Thus note that $Q(R)$, the total cost of misclassification, does not depend on $D(i,j)$ for $j \neq i-1, i+1$. But recall the definition (or perhaps criteria) of total interclass divergence, namely,

$$D = \sum_{i=1}^{n}\sum_{\substack{j=i\\ j\neq i}}^{n} D(i,j) \qquad (2)$$

I believe equations (1) and (2) express the main problem with the existing feature selection — classification scheme, namely that the feature selection criteria (2) is inconsistant with the classification criteria (1). This paper has shown that when $\Sigma_i = 1$ for all i, and $\mu_i < \mu_{i+1}$, a "better" feature selection criteria would be to desire $\tilde{D}$ large where in this case $\tilde{D} \leq \frac{1}{2}(n-1)$ and

$$\tilde{D} = \frac{1}{\sqrt{2\pi}}\sum_{i=1}^{n-1}\int_0^{\frac{1}{2}\sqrt{D(i+1,i)}} e^{-\frac{1}{2}y^2}\, dy$$

The "nice" property about $\tilde{D}$ is that $\frac{1}{\sqrt{2\pi}}\int_0^3 e^{-\frac{1}{2}y^2}\,dy = .4987$ and $\frac{1}{\sqrt{2\pi}}\int_0^\infty e^{-\frac{1}{2}y^2}\,dy = .5$ so that there is no need to worry about $D(i,j)$ becoming too large.

Finally, we consider a numerical example with all covariances equal to 1 and $n = 3$. Assume the means along the channel 1 axis are given by

$$\mu_1 = -6, \quad \mu_2 = 0, \quad \mu_3 = 6$$

and the means along the channel 2 axis are given by

$$\mu_1 = 0, \quad \mu_2 = 1, \quad \mu_3 = 12 .$$

Then $D\Big|_{\text{channel 1}} = D(1,2) + D(1,3) + D(2,3) = (-6)^2 + (-12)^2 + (6)^2 = 216$, and $D\Big|_{\text{channel 2}} = (1)^2 + (-12)^2 + (-11)^2 = 266$. Thus the divergence criteria would imply selection of channel 2.

But, the total probability of misclassification is given by

$$Q(R)\Big|_{\text{channel 1}} = \frac{1}{3}\left[2 - \frac{2}{\sqrt{2\pi}}\int_0^{\frac{1}{2}\sqrt{D(2,1)}} e^{-\frac{1}{2}y^2}\,dy - \frac{2}{\sqrt{2\pi}}\int_0^{\frac{1}{2}\sqrt{D(3,2)}} e^{-\frac{1}{2}y^2}\,dy\right]$$

$$= \frac{2}{3}\left[1 - \frac{1}{\sqrt{2\pi}}\int_0^3 e^{-\frac{1}{2}y^2}\,dy - \frac{1}{\sqrt{2\pi}}\int_0^3 e^{-\frac{1}{2}y^2}\,dy\right]$$

$$= \frac{2}{3}\,(1 - .4987 - .4987)$$

$$\approx .0017$$

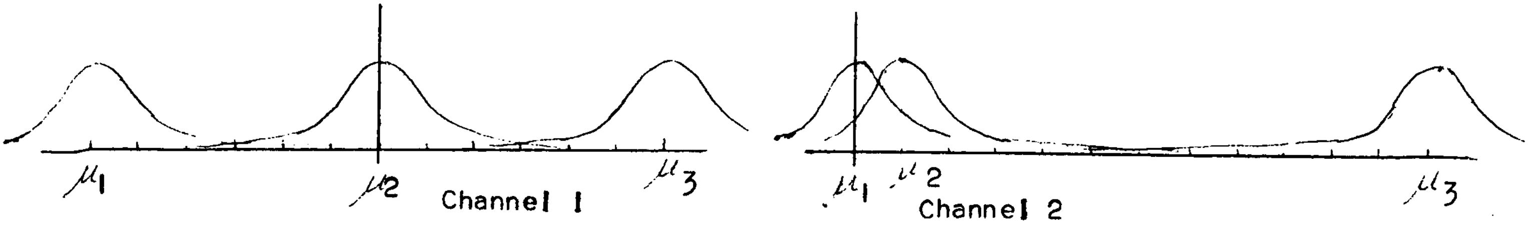
and

$$Q(R)\Big|_{\text{channel 2}} = \frac{1}{3}\left[2 - \frac{1}{\sqrt{2\pi}}\int_0^{\frac{1}{2}} e^{-\frac{1}{2}y^2}\,dy - \frac{1}{\sqrt{2\pi}}\int_0^{\frac{11}{2}} e^{-\frac{1}{2}y^2}\,dy\right]$$

$$= \frac{2}{3}\,(1 - .1915 - .5000)$$

$$\approx .2056$$

Since the probability of misclassification is much less by this criteria the choice would be channel 1. A pictorial representation is given below.



N73-29570 1

UNIVERSITY OF HOUSTON · Founded 1927 ·

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF HOUSTON HOUSTON, TEXAS

DIVERGENCE CONSIDERATIONS II

JCHN QUIREIN
SEPT. 1972

PREPARED FOR
EARTH OBSERVATION CIVISICN , JSC
UNDER
CONTRACT NAS-9-12777

D-1

DIVERGENCE CONSIDERATIONS II

by

J. A. Quirein
University of Houston
Department of Mathematics

Report # 5*

Contract NAS-9-12777

September 1972

DIVERGENCE CONSIDERATIONS II

by

J. A. Quirein

The interclass divergence D(i,j) in a sense, a measure of the "separability" of two classes Π_i and Π_j. The problem of determining a function F of the interclass divergence over all possible combinations of a fixed number of channels such that maximizing F will minimize the probability of misclassification (for that number of channels) has not yet been solved.

Consider the case of three distinct classes Π_1, Π_2, Π_3. One such function of the divergence typically constructed is of the form:

$$F = D(1,2) + D(1,3) + D(2,3).$$

It has been previously shown that maximizing F need not necessarily minimize the probability of misclassification. A second commonly constructed function of the divergence is the following

$$F = \min(D(1,2), D(1,3), D(2,3)).$$

To show how maximizing F does not necessarily minimize the probability of misclassification, let the means along the channel 1 axis be given by

$$\mu_1 = 0, \mu_2 = 2.2, \mu_3 = 5.2$$

and the means along the channel 2 axis be given by

$$\beta_1 = 0, \ \beta_2 = 2, \ \beta_3 = 8$$

then

$$F|_{\text{channel 1}} = \min\ (4.84,\ 27.04,\ 9) = 4.84$$

$$F|_{\text{channel 2}} = \min\ (4,\ 64,\ 36) = 4$$

and maximizing F implies selecting channel 1. The probability of misclassification is verified to be

$$Q(R)|_{\text{channel 1}} = .135$$

$$Q(R)|_{\text{channel 2}} = .107$$

which indicates in this case, the "best" choice would be channel 2.

N73-29571 1

DEPARTMENT OF MATHEMATICS

UNIVERSITY OF HOUSTON HOUSTON, TEXAS

PATTERN RECOGNITION AND THE PCTENTIAL FUNCTION

MARY ANN RCBERTS

SEPT. 1972

PREPARED FOR
EARTH OBSERVATICN CIVISICN , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

2-1-

PATTERN RECOGNITION AND THE POTENTIAL FUNCTION

Report # 6

by

M. R. Roberts

University of Houston
Mathematics Department

NASA Contract: NAS-9-12777

September 1972

E-1-a

PATTERN RECOGNITION AND THE POTENTIAL FUNCTION

Supposing that we have two sets A and B which do not intersect in a space(Hilbert) $\mathcal{X}$, then there exists at least one separation function $\psi(X)$ for which $\psi(X) > 0$ if $X \in A$ and $\psi(X) < 0$ if $X \in B$. The idea of the potential function is to build, by an iterative process with a finite number of known points from A and B, a sequence of functions $K_r(X)$ which tend to one of these separation functions as r increases.

Assume that in $\mathcal{X}$ there is a linearly independent system of functions, $\varphi_i(X)$, a subset of a complete system, such that for any two separable (always taken hereafter to mean in the geometric sense) sets in $\mathcal{X}$, $\psi(X) = \sum_{i=1}^{N} c_i \varphi_i(X)$ separates these two sets, N depending on the sets to be separated. In order to have convergence in probability let the $\varphi_i(X)$'s be an orthogonal or orthonormal system. Additionally if $K(X,Y)$, the potential function, is bounded on $A \cup B$ and the function $\psi(X)$ rigorously separates A and B(i.e. $\psi(X) \begin{cases} > \epsilon & \text{if } X \in A \\ < \epsilon & \text{if } X \in B \end{cases}$ where $\epsilon > 0$), it can be proved that there is an integer m, independent of the teaching sequence so that the number of errors corrected does not exceed m. If the appearances of the points in the teaching sequence are independent events and at any rth step there is a strictly positive probability of correcting an error if separation of the sets has not yet occurred, then the probability is unity that the separation of the sets will be realized in a finite number of steps. If we agree to terminate the teaching process as soon as no error has occurred in L examples in the sequence following an error correction (L, an arbitrary prescribed integer) then the entire teaching sequence will be terminated in Lm steps. Let P be the probability of error in the process after termination of teaching and $\epsilon > 0$, $\delta > 0$, then it can be proved that the probability that $P < \epsilon$ exceeds $1 - \delta$ if L satisfies $L > \dfrac{\ln(\delta/m)}{\ln(1-\epsilon)}$.

ALGORITHM

The construction of a separation function $\psi(X)$ shall be accomplished as follows:

Let the potential function be defined by:

$$K(X,Y) = \sum_{i=1}^{N} \lambda_i^2 \varphi_i(X) \varphi_i(Y)$$

and let A be the positive set and B, the negative one.

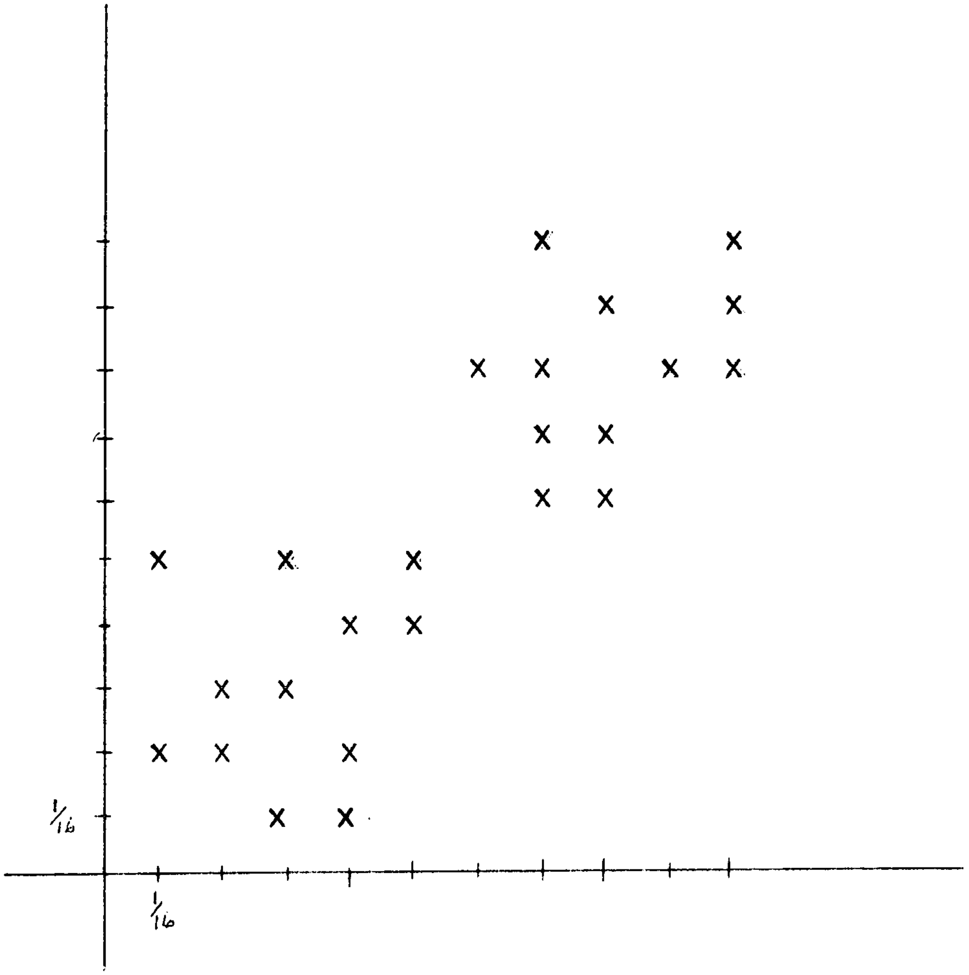
For $K_1(X)$ we will take:

$$K_1(X) = \begin{cases} K(X,X_1) & \text{if } X_1 \in A \\ -K(X,X_1) & \text{if } X_1 \in B \end{cases}$$

Inductively we proceed after the <u>r</u>th step, in which the function $K_r(X)$ was constructed. Compute $K_r(X_{r+1})$. If either $K_r(X_{r+1}) > 0$ and $X_{r+1} \in A$ or $K_r(X_{r+1}) < 0$ and $X_{r+1} \in B$ (i.e. the function $K_r(X)$ agrees at the point X_{r+1} with our original convention of A, positive and B, negative), we shall set $K_{r+1}(X) = K_r(X)$ and proceed to the next point X_{r+2}. If $K_r(X_{r+1}) > 0$ and $X_{r+1} \in B$, set $K_{r+1}(X) = K_r(X) - K(X,X_{r+1})$. If $K_r(x_{r+1}) < 0$ and $X_{r+1} \in A$, set $K_{r+1}(X) = K_r(X) + K(X,X_{r+1})$. In either of the latter two cases the potential function is altered by addition to it of the potential of the (r + 1)st point with sign necessary to "correct" the function at this step.

EXAMPLE

For our space we choose $[-1,1] \times [-1,1]$. Let $A = \left\{(x,y): \frac{1}{16} \leq x,y \leq \frac{5}{16}\right\}$ and $B = \left\{(x,y): \frac{3}{8} \leq x,y \leq \frac{5}{8}\right\}$ and, using the training points given in figure 1, build a separation function $\psi(X) = \sum_{i=1}^{N} c_i \varphi_i(X) \begin{cases} > 0 \text{ if } X \in A \\ < 0 \text{ if } X < B \end{cases}$. Since 1 and x + y are linearly independent and defined on using the Gram-Schmidt process we find for $\varphi_1(X) = 1$ and $\varphi_2(X) = x + y - 1$ we have an orthogonal set of functions where the inner product is defined by $(\varphi_i(X), \varphi_j(X)) = \int_{-1}^{1}\int_{-1}^{1} \varphi_i(X) \varphi_j(X)\, dx\, dy$. Letting $\lambda_i = 1$, $K(X,X_k) = 1 + (x_k + y_k - 1)(x + y - 1)$ where $X = (x,y)$ and $X_k = (x_k,y_k)$.



x_1=(10/16,10/16)	x_9 =(2/16, 2/16)	x_{17}=(7/16, 7/16)
x_2=(4/16, 4/15)	x_{10}=(9/16, 8/16)	x_{18}=(7/16,10/16)
x_3=(8/16, 7/16)	x_{11}=(4/16, 2/16)	x_{19}=(5/16, 4/16)
x_4=(1/16, 5/16)	x_{12}=(6/16, 8/16)	x_{20}=(7/16, 6/16)
x_5=(1/16, 2/16)	x_{13}=(4/16, 1/16)	x_{21}=(5/16, 5/16)
x_6=(8/16, 9/16)	x_{14}=(3/16, 5/16)	x_{22}=(7/16, 8/16)
x_7=(8/16, 6/16)	x_{15}=(10/16, 9/16)	x_{23}=(10/16, 8/16)
x_8=(2/16, 3/16)	x_{16}=(3/16, 3/16)	x_{24}=(3/16, 1/16)

Figure 1

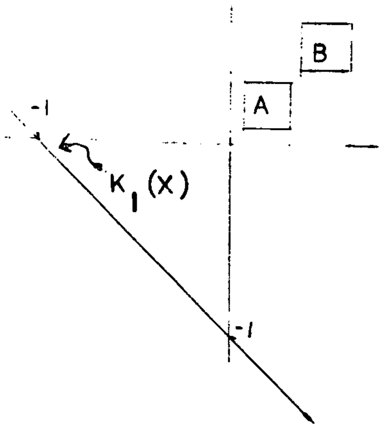


Figure 2

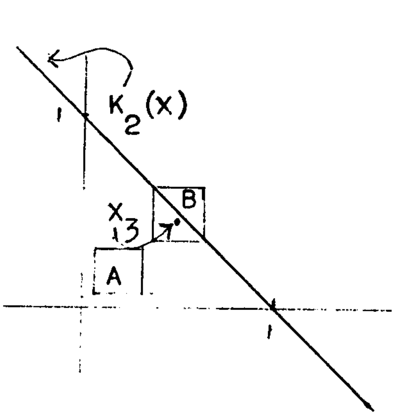


Figure 3

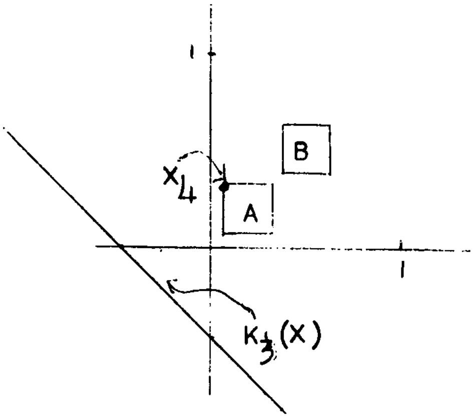


Figure 4

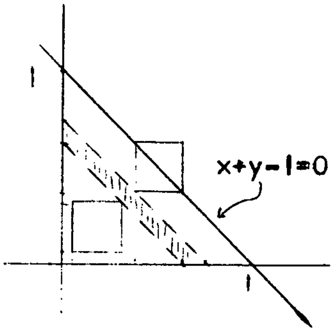


Figure 6

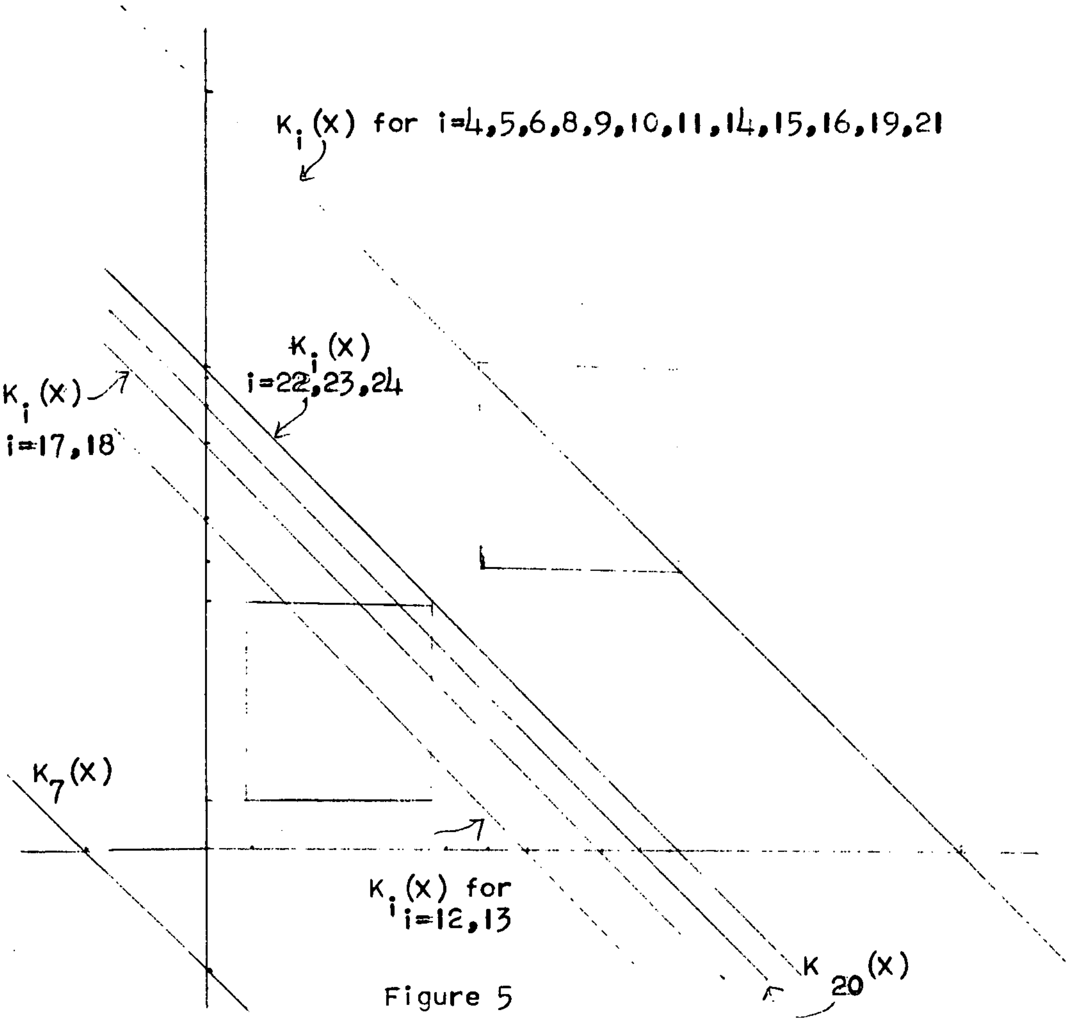


Figure 5

Construction of $K_r(X)$ will therefore always yield a line, moreover, a line whose slope is negative one. Since A and B can be separated by such a line choice of $N = 2$ will yield a separating function as desired.

By definition $K_1(X) = - K(X,X_1) = -1 - \frac{1}{4}(x + y - 1)$ since $X_1 \in B$. Figure 2 shows $K_1(X)$ in relation to A and B. Testing X_2 in $K_1(X)$ we find $K_1(X_2) = -\frac{7}{8} < 0$. Since $X_2 \in A$ put $K_2(X) = K_1(X) + K(X,X_2) = -\frac{3}{4}(x + y - 1)$. In figure 3 we see that X_3 lies below the line $K_2(X) = 0$ and testing we find that $K_2(X_3) = \frac{3}{64} > 0$ and since $X_3 \in B$, $K_3(X) = K_2(X) - K(X,X_3) = -1 - \frac{11}{16}(x + y - 1)$. Since $K_3(X_4) < 0$ and $X_4 \in A$ (see figure 4), $K_4(X) = K_3(X) + K(X,X_4) = -\frac{21}{16}(x + y - 1)$. Since $K_4(X_5) > 0$ and $X_5 \in A$, $K_5(X) = K_4(X) = -\frac{21}{16}(x + y - 1)$.

Continuing the process we find:

$$K_6(x) = K_5(x) = K_4(x) = -\frac{21}{16}(x + y - 1)$$
$$K_7(X) = -1 - \frac{19}{16}(x + y - 1)$$
$$K_8(X) = K_9(X) = K_{10}(X) = K_{11}(X) = -\frac{15}{8}(x + y - 1)$$
$$K_{12}(X) = K_{13}(X) = -1 - \frac{7}{4}(x + y - 1)$$
$$K_{14}(X) = K_{15}(X) = K_{16}(X) = -\frac{9}{4}(x + y - 1)$$
$$K_{17}(X) = K_{18}(X) = -1 - \frac{17}{8}(x + y - 1)$$
$$K_{19}(X) = -\frac{41}{16}(x + y - 1)$$
$$K_{20}(X) = -1 - \frac{19}{8}(x + y - 1)$$
$$K_{21}(X) = -\frac{11}{4}(x + y - 1)$$
$$K_{22}(X) = K_{23}(X) = K_{24}(X) = -1 - \frac{43}{16}(x + y - 1)$$

Figure 5 shows the relationship of $K_i(X)$, $i = 4, \ldots, 24$, to the sets A and B.

Taking $\psi(X) = K_{24}(X) = -1 - \frac{43}{16}(x + y - 1)$, $\psi(X) = \sum_{i=1}^{2} c_i \; \varphi_i(X)$ for $c_1 = -1$ and $c_2 = -\frac{43}{16}$. Testing the function, it does, indeed, separate the training sample, for $\psi(X) > 0$ for all $X \in A$ and $\psi(X) < 0$ for all $X \in B$. A geometric analysis of the sets shows any function of the form $-1 + q(x + y - 1)$ will separate if $\frac{5}{8} < \frac{q + 1}{q} < \frac{3}{4}$ and our $q = -\frac{43}{16}$ satisfies this inequality.

Although the training points of this example were purposefully "rigged" to insure that each part of the definition of $K_r(X)$ would be used and that convergence would be accomplished in the limited number of training points, in the latter case, if the training sample had run out without clear separation it could have been reused in the continued construction of the function. Certain points appear more critical to the process; in our example, those points in A nearest the shaded region in figure 6 are more sensitive to change and those points in B nearest to the line $x + y - 1 = 0$ and to its left produce more change as the algorithm progresses. However, these remarks are pertinent to this example alone as alteration by so simple a change as choice of $X_1 \in A$ would require completely different, though analogous, comments.

It was necessary to avoid any point X_{r+1} for which $K_r(X_{r+1}) = 0$ since the algorithm does not deal with this possibility (i.e. take X on the line $x + y - 1 = 0$ at alternating steps of the function construction beginning at $r = 2$). It would seem advisable to add to the algorithm "if $K_r(X_{r+1}) = 0$, let X_{r+2} become the $(r + 1)$st point, discarding the original X_{r+1} as a training point and renumbering the points."

EVALUATION

In [10], the purely geometric method of the potential function is compared with a structural approach, basically one of recognition of broad interclass similarities, and it is the opinion stated in this paper that neither method is suitable to solve complex problems. In the case of the recognition of the letters of the alphabet photoelectric cells 1000x1000 may be needed for a clear picture, making the vector representation 1,000,000-tuples, which might produce a memory storage problem. In the development of the idea of a potential function for construction of a separating function any orthonormal system of functions $\varphi_i(X)$'s will produce convergence of the algorithm. It seems obvious that for some choices of the system convergence might be more rapid than for others. However, nowhere was there mention of how this choice might be made to minimize N. In addition $\psi(X)$ can be realized as a finite linear combination of the $\varphi_i(X)$'s where the number N of the $\varphi_i(X)$'s necessary depended on the sets involved. There was no discussion of the problem of how determination of an appropriate N, let alone a minimal one, could be made.

This method does, however, have the advantage that convergence in probability is assured in a finite number of steps to any desired degree of reliability. The experiments made and reported bear out this result by the high percentage of accuracy attained.

BIBLIOGRAPHY

1. Braverman, E.M., "Experiments on Machine Learning to Recognize Visual Patterns", Automation and Remote Control, Vol. 23, no. 3, pp. 349-364, March, 1962.

2. Bashkirov, O.A., Braverman, E.M., and Muchnik, I.B., "Potential Function Algorithms for Pattern Recognition Learning Machines", Automation and Remote Control, Vol. 25, no. 5, pp. 692-695, May, 1964.

3. Aizerman, M.A., Braverman, E.M., and Rozonoer, L.I., "The Probability Problem of Pattern Recognition Learning and the Method of Potential Functions", Automation and Remote Control, Vol 25 no. 9, September, 1964.

4. Aizerman, M.A., Braverman, E.M., and Rozonoer, L.I., "Theoretical Functions of the Potential Function Method in Pattern Recognition Learning", Automation and Remote Control, Vol 25, no. 6, June 1964.

5. Aizerman, M.A., Braverman, E.M, and Rozonoer, L.I., "The Method of Potential Functions for the Problem of Restoring the Characteristic of a Function Converter from Randomly Observed Points", Automation and Remote Control, Vol. 25, no. 12, pp 1705-1714, December, 1964.

6. Braverman, E.M., and Pyatnitskii, E.S.,"Estimation of the Rate of Convergvence of Algorithms Based on the Potential Function Method", Automation and Remote Control, Vol. 27, no. 1, pp. 95-112, January, 1966.

7. Braverman, E.M., "On the Method of Potential Functions", Automation and Remote Control, Vol. 26, no. 12, pp. 2205-2213, December, 1965.

8. Aizerman, M.A., Braverman, E.M., and Rozonoer, L.I., "The Robbins-Monro Process and the Method of Potential Functions", Automation and Remote Control, Vol. 26, no. 11, pp. 1951-1954, September, 1965.

9. Aizerman, M.A., Braverman, E.M., and Rozonoer, L.I., "The Choice of Potential Function in Symmetric Spaces", Automation and Remote Control, Volume 10, pp. 124-152, October, 1967.

10. Aizerman, M.A., "A Note on Two Problems Connected with Pattern Recognition", Automation and Remote Control, Volume 4, pp. 137-144, April, 1969.

N73-29572

1

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF HOUSTON HOUSTON, TEXAS

THE FUZZY SETS APPROACH TO
PATTERN RECOGNITION
TERRY WILSON
SEPT. 1972

PREPARED FOR
EARTH OBSERVATION DIVISICN , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

F 1-

THE CONCEPT OF FUZZY SETS IN PATTERN RECOGNITION

by

Terry Wilson

Department of Mathematics

University of Houston

Contract NAS-9-12777

Report # 7

Septmeber 1972

Introduction

The purpose of this paper is twofold:

(1) Introduce the concept of fuzzy set (Zadeh [1])

(2) Apply the concept of fuzzy set to pattern recognition (Wee [2])

We will consider only the ideas from fuzzy set theory that are directly related to pattern recognition. Our approach to pattern recognition will follow the PhD thesis of W. G. Wee. In this thesis an iterative procedure for learning the equi-membership surfaces and for generating a set of discriminate functions for two pattern classes is given.

Fuzzy Sets

The concept of fuzzy sets was first introduced by Zadeh [1]. Since we will be interested in fuzzy sets only with respect to pattern recognition, we will define our concepts in $\Omega = E^n$.

Definition 1: A fuzzy set A in Ω is characterized by a membership function $f_A:\Omega \to [0,1]$ with the value of f_A at x representing the "grade of membership" of x in A.

As an example of a fuzzy set in E^1, let A be the set of all numbers "much larger" than 14. One can give a precise characterization of A by specifying $f_A(x)$ on E^1 (eg. $f_A(-1) = 0$, $f_A(1000) = .2$, $f_A(10^6) = .5$, etc.). It should be noted that this characterization is subjective.

Definition 2: The union of two fuzzy sets A and B is a fuzzy set C, written $C = A \cup B$, whose membership function is given by

$$f_C(x) = \mathrm{Max}[f_A(x), f_B(x)]$$

for $x \varepsilon \Omega$.

Definition 3: The intersection of two fuzzy sets A and B is a fuzzy set C, written $C = A \cap B$, whose membership function is given by

$$f_C(x) = \mathrm{Min}[f_A(x), f_B(x)]$$

for $x \varepsilon \Omega$.

Definition 4: A fuzzy set A is *convex* if and only if the sets T_α defined by

$$T_\alpha = \{x | f_A(x) \geq \alpha\}$$

are convex for all $\alpha \varepsilon (0,1]$.

Definition 5: A fuzzy set A is *bounded* if and only if the sets

$$T_\alpha = \{x | f_A(x) \geq \alpha\}$$

are bounded for all $\alpha > 0$.

Definition 6: The *maximal grade* of a fuzzy set A, written M_A is defined

$$M_A = \sup_{x \varepsilon \Omega} f_A(x)$$

Theorem 1: Let A be a bounded fuzzy set. Then there is at least one point $x_0 \varepsilon \Omega$ at which M_A is essentially attained in the sense that, for each $\varepsilon > 0$, every spherical neighborhood of x_0 contains points in $Q(\varepsilon) = \{x | f_A(x) \geq M_A - \varepsilon\}$.

Definition 7: The *core* of a bounded fuzzy set A, written $C(A)$, is the set of all points in Ω at which M_A is essentially attained.

Definition 8: Let A and B be two bounded fuzzy sets and H a hyperplane. Let $K_H \varepsilon \mathbb{R}$ such that $f_A(x) \leq K_H$ on one side of H

and $f_B(x) \leq K_H$ on the other side of H. Set $\bar{M}_H = \inf K_H$ and $D_H = 1 - \bar{M}_H$. D_H is called the degree of separation of A and B by H. The *degree of separation* of A and B, denoted D, is defined as $D = 1 - \bar{\bar{M}}$ where $\bar{\bar{M}} = \inf_H \bar{M}_H$.

Theorem 2 (Separation Theorem): Let A and B be bounded convex fuzzy sets. Set $C = A \cap B$. Then $D = 1 - M_C$ (where M_C is the maximal grade of C).

Note that Theorem 2 says that the highest degree of separation of two bounded convex fuzzy sets A and B that can be obtained with a hyperplane is $1 - M_C$.

The above definitions and theorems are contained in Zadeh's paper; they do not exhaust all of the material contained there. Wee introduces the following definitions.

Definition 9: A *fuzzy pattern class* is a pattern class which is a fuzzy set.

Definition 10: A *semi-fuzzy set* is a fuzzy set A such that

$$M_A = \sup_x f_A(x) = \operatorname{Max}_x f_A(x) = 1 .$$

Definition 11: Let A be a fuzzy set. The non-fuzzy section of A is defined by $NFS = \{x|f_A(x) = 1\}$ and the complete-fuzzy section of A is defined by $COM = \{x|f_A(x) < 1\}$.

Definition 12: A equi-membership surface of a fuzzy set is a separating surface such that points on the surface have equal grade of membership.

Recognition of Two Fuzzy Sets

The discussion that follows deals with the situation in which there are two bounded and convex fuzzy pattern classes, A and B, to be recognized.

Suppose we have a set X of training samples. Let $\alpha \varepsilon [0,1]$ and define

$$L_A = \{x|f_A(x) \geq \alpha \text{ and } f_B(x) < \alpha\}$$

and

$$L_B = \{x|f_B(x) \geq \alpha \text{ and } f_A(x) < \alpha\}$$

We further assume that α can be selected so that $X \subseteq L_A \cup L_B \subseteq \Omega_X = E^n$. Note that the separation theorem tells us that the lowest value of α that can be selected is $M_{A\cap B}$. In practice we seldom know $M_{A\cap B}$.

Wee's algorithm is an iterative procedure for searching for equi-membership surfaces until the complete set of training samples is contained within these surfaces.

The first step separates the non-fuzzy section and the complete-fuzzy section of the training samples for A(B). [Note that this step may not be necessary] Separating boundaries are then generated to retain the complete-fuzzy section of A(B). The retained training samples are then mapped into $\Omega_y = E^n$. Separation of the non-fuzzy and complete-fuzzy sections of "A"(B) in Ω_y (as in Ω_x) is then determined. The complete-fuzzy sections of A and B are retained and are mapped into $\Omega_z = E^n$. This procedure continues until Ω_x is partitioned into two regions. The algorithm converges in a finite number of steps. The algorithm generates a set of discriminate functions which partitions Ω_x into two regions; generalization to any other point in Ω_x is based on these discriminate functions. The evaluation of this generalization must be based upon experience.

Figure 1 gives a block diagram of the algorithm.

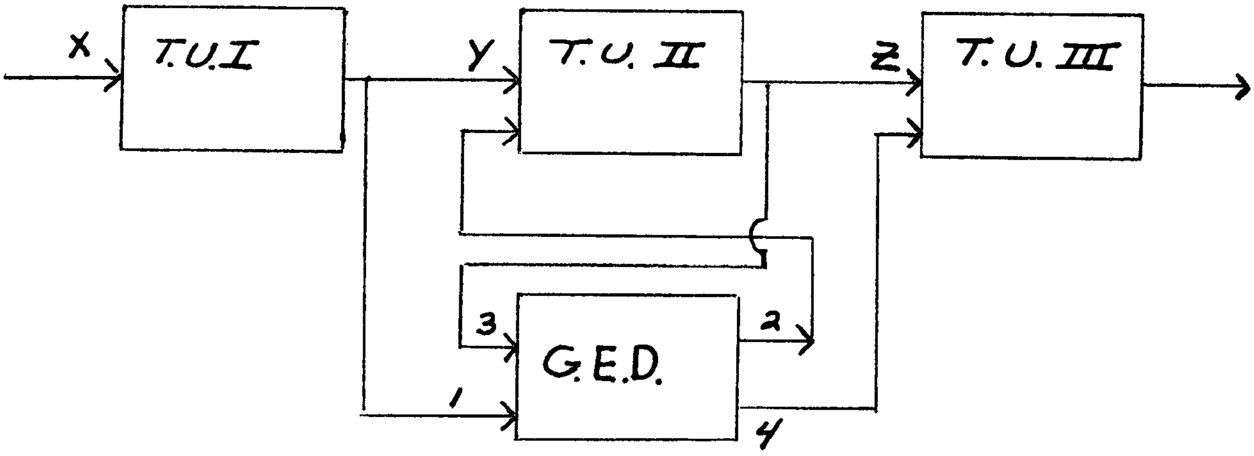


Figure 1: Block Diagram of Algorithm

The training samples X are the input for Transformation Unit (TU) I which is a polynomial transformation in many cases. The output of TU I is a set $Y \subseteq \Omega_y$ which is sent (usually) to the general adaptive element (GED). First the GED uses the generalized inverse algorithm (Ho and Kashyap's algorithm [3]) to test the linear separability of the samples and to find the separating hyperplane. If the samples are not linearly separable Widrow and Hoff's algorithm [4] is used to generate a minimum mean sequence error hyperplane $H: X^T W + W_0 = 0$. Note that the distance from a point X_i to H is $d_i = X_i^T W + W_0$. From the samples "close" to H and those erroneously classified, the minimum and maximum distances from H are searched in order to obtain two parallel separating hyperplanes H_1 and H_2. They are as follows:

$$H_1 : X^T W + W_0 - |W| d(\max) = 0$$

$$H_2 : X^T W + W_0 - |W| d(\min) = 0$$

The following decision rules are now implimented:

(1) $P \varepsilon A$ if $P^T W + W_0 \geq |W| d(\max)$

(2) $P \varepsilon B$ if $P^T W + W_0 \leq |W| d(\min)$

(3) If P is such that $|W| d(\min) < P^T W + W_0 < |W| d(\max)$, send P to TU II. Let Y' represent the set of P's that were not classified. Let $Y_i \varepsilon Y'$. Then TU II transforms $Y_i \varepsilon Y'$ into $Z_i \varepsilon \Omega_z = E^n$. <u>Two</u> of the types of transformations used are as follows:

$$(1) \quad Y_{ij} \rightarrow Z_{ij} = \alpha \frac{|Y_i^T W + W_0|}{|W|}$$

$$(2) \quad Y_{ij} \rightarrow Z_{ij} = \exp\{-\alpha \frac{|Y_i^T W + W_0|}{|W|}\}$$

The set Z of Z_i's is then sent to the GED and the process continues. (We remark again that the process terminates after a finite number of transformations.)

Bibliography

1. Zadeh, L. A., "Fuzzy Sets", Information and Control, 8, 338-353 (1965).

2. Wee, W. G., "On generalization of Adaptive Algorithms and Application of the Fuzzy Sets Concept to Pattern Classification, PhD. dissertation, Purdue University, 1967.

3. Ho, Y. C. and Kashyap, R. L., "An Algorithm for Linear Inequalities and its Applications", IEEE Trans. on Electric Computers, Vol. EC-14, No. 5, Oct., 1965, p. 683-688.

4. Widrow, B. and Hoff, M. E., "Adaptive Switching Circuits", Standord Electronics Laboratories Report 1553-1; June 30, 1960.

N73-29573

DEPARTMENT OF MATHEMATICS

UNIVERSITY OF HOUSTON HOUSTON, TEXAS

PATTERN RECOGNITION AND THE LINEAR DISCRIMINANT FUNCTION

L. H. FINCH

SEPT. 1972

PREPARED FOR
EARTH OBSERVATION DIVISION , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

B 1-

PATTERN RECOGNITION AND LINEAR DISCRIMINANT FUNCTION

Report # 8

by

L. H. FINCH

UNIVERSITY OF HOUSTON
DEPARTMENT OF MATHEMATICS

September 1972

NASA Contract: Nas-9-12777

G 1-a

ABSTRACT

The purpose of this paper is to discuss the properties of a linear discriminant function for the case of arbitrary distributions with equal covariance matrices. Using two examples, a comparison is made showing how the difference of the means relates to the covariance matrices.

In the solution of recognition problems the linear discriminant function LDF of the form

$$d(x) = x'\Sigma^{-1}(\mu_1 - \mu_2) - \tfrac{1}{2}(\mu_1 + \mu_2)'\Sigma^{-1}(\mu_1 - \mu_2)$$

finds wide application, where the vectors in the n-dimensional space Ω of the recognized object, the mean values, and the general covariance matrix of the distributions in question are denoted by x, u_1, u_2 and Σ respectively. The method of application of the LDF consists in determining the membership of the object x in the first class if $x \in R_1 = \left\{ x \mid d(x) \geq 0 \right\}$ and in the second class if $x \in \Omega - R_1$.

The problem is to carry out the discrimination process efficiently in the case of imcompletely known distributions, for identical covariance matrix $\Sigma_1 = \Sigma_2 = \Sigma$, since in practice the test of normality of multidimensional distribution is rarely made. If $\alpha = (\mu_1 - \mu_2)'\Sigma^{-1}(\mu_1 - \mu_2)$, the interclass divergence, then the bound on $p(\alpha)$, the probability of misclassification, is given by

(1) $$p(\alpha) \leq \left[\tfrac{1}{4}(u_1 - u_2)'\Sigma^{-1}(u_1 - u_2) + 1\right]^{-1}$$

for the upper limit and 0 for the lower limit. (For the proof we refer the reader to [6]).

If the recognized object x comes from the one dimensional space, then the relation between the distance between u_1 and u_2 and $p(\alpha)$ can be easily computed. In order to obtain $p(\alpha) \leq \epsilon$ for some $\epsilon > 0$

(2) $$|u_1 - u_2| \geq 2|\Sigma|^{\frac{1}{2}}(1 - \epsilon/\epsilon)^{\frac{1}{2}}$$

Thus in order to compare two different problems with given covariance matrices, consider the following numerical example.

Example 1.

Let $\Sigma_1 = (4)$, $\Sigma_2 = (\frac{1}{4})$ and $\epsilon = 1/10$. From the equation (2) we obtain $|u_1 - u_2| \geq 12$ for Σ_1 but $|\nu_1 - \nu_2| \geq 3$ for Σ_2 in order to have the maximum probability of misclassification less than or equal to $\epsilon = 1/10$. Note that in each case the inter-class divergence is 36. The Figure-1 describes this example graphically.

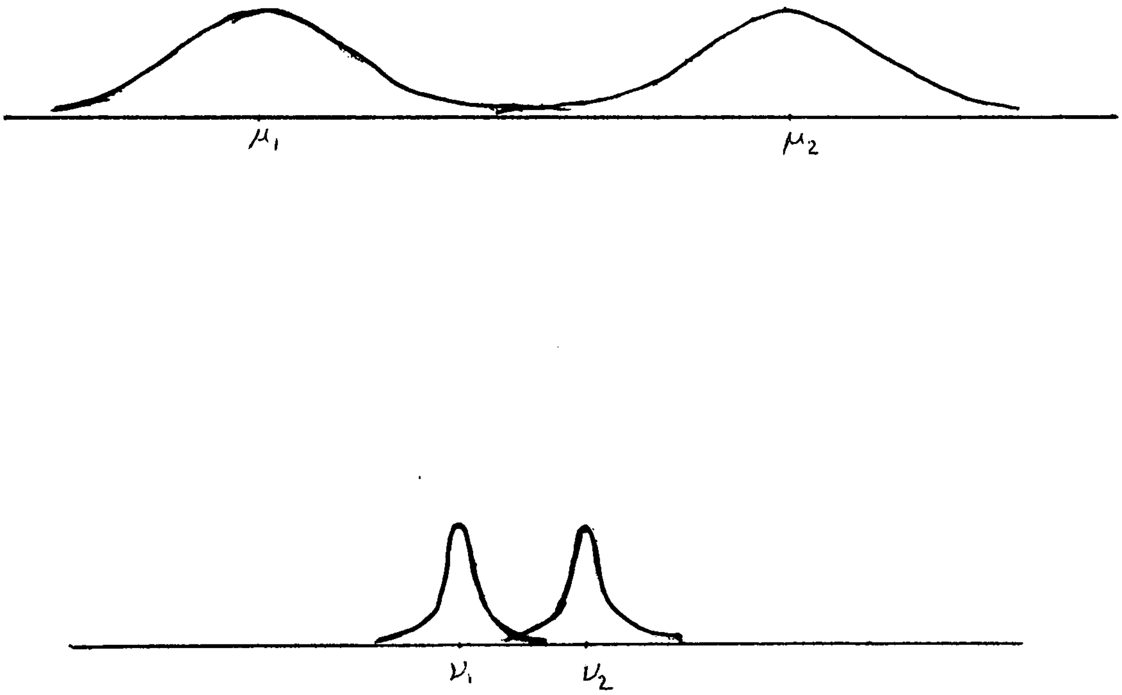


Figure-1

In the case of multidimensional space, from the equation (1) our scheme in comparison of two problems with given covariance matrices is quite obvious. Let $\alpha = (u_1 - u_2)'\Sigma^{-1}(u_1 - u_2)$, then this equation gives the ellipsoid in the principle axes plane with the length of the ith principal axis $2\sqrt{\lambda_i\alpha}$, where $\lambda_1, \lambda_2, \cdots, \lambda_n$ are the eigenvalues of Σ. Hence as long as the difference of the two means μ_1 and μ_2 lies on this ellipsoid, the interclass divergence will be constant and so the upper limit on the maximum probability of misclassification remains constant also. It is clear that the shape of the ellipsoid depends of the covariance matrix. The dependence of the function p on the magnitude of the degree of divergence of the classes α is shown in Figure-2. The curve denoted by p_n shows the relation in the case of normal distributions.

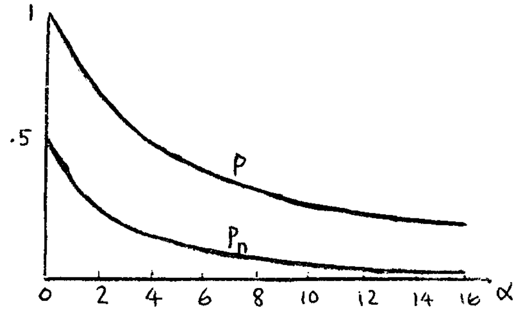


Figure-2

Evaluation.

For arbitrary interclass divergence α the maximum probability of misclassification of any classes using LDF with unknown u_1, u_2 and Σ is greater than the corresponding probability calculated for multidimensional normal distributions with the same u_1, u_2 and Σ. However, the maximum value of the probability of misclassfication is a decreasing function of α and tends to 0 as $\alpha \to \infty$. The lower limit of the probability of misclafication for arbitrary α is equal to 0, which signifies that cases may be encountered even for small where the LDF constructed will classify without error. For $\alpha > 4$ the probabulity of misclassification is always less than $\frac{1}{2}$, i.e., in these cases classification bu means of LDF will always be better than random classification with equal probabilities of assigning the objects to the two classes. For $0 \leqslant \alpha \leq 4$ the maximum probability of misclassification for the two classes is greater than $\frac{1}{2}$, which means the operation of the LDF may be poorer than random classification.

BIBLIOGRAPHY

1. Agmon, Shmuel, "The relaxation Method for Linear Enequalities", Canadian Journal of Mathematics, Vol. 6, No. 3, pp. 382-392, 1954.

2. Ho, Y. C. and Kashap, R. L., "An Algorithm for Linear Inequalities and its Applications", IEEE Trans. on Elec. Computers, Vol. EC-14, No. 5, pp. 683-688, October 1965.

3. Koford, J. S. and Groner, G. F., "The Use of an Adaptive Threshold Element to Design a Linear Optimal Pattern Classifier", IEEE Trans. on Inf-Theory, Col. IT-12, pp. 42-50, January 1966.

4. Sammon, John W. Jr., "Short Notes" (on an Optimal Discriminant Plane), IEEE Trans. on Computers, pp. 826-829, September 1970.

5. Wee, W. G., "Generalized Inverse Approach to Adaptive Multiclass Pattern Classification", IEEE Trans. on Computers, Vol. C-17, No. 12, pp. 1157-1164, December 1968.

6. Zhezhel, Yu. N., "The Efficiency of a Linear Discriminant Function for Arbitrary Distributions", Engineering Cybernetics, No. 6, pp.107-111, 1968.

N73-29514 1

DEPARTMENT OF MATHEMATICS

UNIVERSITY OF HOUSTON HOUSTON, TEXAS

LINEAR PROGRAMMING AND ITS APPLICATION TO PATTERN CLASSIFICATION

M. J. O'MALLEY

SEPT. 1972

PREPARED FOR
EARTH OBSERVATION DIVISION , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

H-1

LINEAR PROGRAMMING AND ITS APPLICATION TO PATTERN RECOGNITION PROBLEMS

by

M. J. O'Malley

Department of Mathematics

University of Houston, Houston, Texas

Contract NAS - 9 - 12777

Report # 9

September 1972

In this paper we discuss linear programming and linear programming like techniques as applied to pattern recognition problems. Our method will be to summarize three relatively recent research articles on such applications. In particular, we summarize the main results of each paper, indicating the theoretical tools needed to obtain them, and we include a synopsis of the author's comments with regard to the applicability or non-applicability of his methods to particular problems, including computational results wherever given. For more detailed information on the methods mentioned here or other such techniques, the reader is referred to the particular research article of interest.

The basic problem considered in all three papers is the following: Given two sets of patterns A and B (we consider each pattern as a point in E^n - Euclidean n-space), does there exist a surface in E^n which separates A and B? That is, does there exist a surface in E^n such that all the points of A lie on one side of the surface and all the points of B lie on the other side? A special, but much studied, case of the above question is: Does there exist a plane (hyperplane) in E^n which separates A and B?

The paper is appropriately divided into three sections, one for each article.

1. Linear and nonlinear separation of patterns by linear programming.[1]

Let A and B be two sets of patterns, the set A consisting of m patterns, the set B consisting of k patterns, where each pattern consists of n scalar observations. Assuming that each pattern represents a point in E^n, we wish to determine a surface in E^n that separates A and B.

The author of this article, O. L. Mangasarian, considers two methods of attempting to separate A and B and states that a generalization of his second method can be made. In particular, Mangasarian attempts to separate A and B by:

(i)linear separation (by a plane); and

(2)a quadratic surface.

We now give a summary of the theoretical details and development of the algorithm.

A pattern will be a row vector $(x_1,\ldots,x_n)$ in E^n, each entry x_i called an observation. We represent a set A containing m patterns as an $m \times n$ matrix, each row of which represents a pattern in A.. Using this notation, our problem is to determine a surface in E^n such that if the m rows of the matrix A and the k rows of the matrix B are considered as points in E^n, then they fall on opposite sides of the surface. Mangasarian states and derives his results for the linear separability case and states two of the corresponding results for the quadratic case. We follow his lead and restrict ourselves to the linear case.

Thus, we wish to determine a single plane

$$xd - \gamma = 0 \qquad (1)$$

where d is an n-dimensional column vector of real numbers, and γ is a scalar (real number) such that

$$Ad - e\gamma > 0 \qquad (2)$$

$$Bd - \ell\gamma < 0 \qquad (3)$$

where e and ℓ are respectively m- and k-dimensional column vectors of ones.

We now make the following definition.

Definition. Two sets of patterns A and B are linearly separable if and only if there exists some d,γ such that (2) and (3) are true. If no such d,γ exist, then A and B are said to be linearly inseparable.

Lemma 1. A and B are linearly separable if and only if there exists an n-dimensional vector c of constants and real numbers α and β such that

$$Ac - e\alpha \geq 0 \qquad (4)$$

$$-Bc + \ell\beta \geq 0 \qquad (5)$$

$$\alpha - \beta > 0 \qquad (6)$$

$$f \geq c \geq -f \qquad (7)$$

where f is an n-dimensional column vector of ones.

Now, if $\alpha - \beta$ is considered as the objective function of the linear programming problem with constraints (4), (5), and (7), we have the following theorem.

Theorem 1. Necessary and sufficient conditions for linear separability of A and B is that $\Theta(A,B) > 0$ where $\Theta(A,B)$ is the solution of the linear programming problem

$$\Theta(A,B) = \max_{c,\alpha,\beta}\{\alpha - \beta | \text{ subject to the constraints (4), (5), and (7)}\}.$$

Corollary 1. Necessary and sufficient conditions for linear inseparability of A and B is that $\Theta(A,B) = 0$.

(It should be remarked that the author suggests two possible approaches in case A and B are linearly inseparable.

(i)A technique of eliminating points of A or points of B so that those points remaining are linearly separable.

(ii)A technique which uses a finite number of planes to separate A and B.)

Mangasarian then invokes the duality principle of linear programming [8; p. 71-74] to obtain the analogues of theorem 1 and corollary 1. He uses the latter analogue to obtain the following condition, which is similar to a condition of Highleyman [12] and Nilsson [22]. It is an immediate way of determining linear inseparability, according to Mangasarian.

Theorem 3. (Dual Inseparability Criterion). Necessary and sufficient conditions in order that the sets of patterns A and B be linearly inseparable is that the system

$$
\begin{aligned}
A'u - B'v &= 0 \\
e'u &= 1 \\
\ell'v &= 1 \\
u &\geq 0 \\
v &\geq 0
\end{aligned}
$$

has a solution, where u and v are m- and k-dimensional column vectors and the prime denotes transpose. (e and ℓ are as defined previously.)

Although the author does not present any computational results for his method, he does make comments regarding its usefulness. He says that the most widely used method for nonparametric pattern separation is Rosenblatt's error correction procedure [26], [27] for linear separation or a modification of it. [10],[21]. This method is based on a very simple iterative procedure. One advantage of this method over his is its simplicity. Its main disadvantage seems to be its inability to determine inseparability of pattern sets when it occurs. This is a consequence of the fact that the error correction procedure converges only when the pattern sets are separable, a fact which is not known a priori. Since it is possible to construct some simple examples for which the error correction procedure converges very slowly, the problem of distinquishing between slow convergence and nonconvergence may be a difficult one. Another advantage of his technique, Mangasarian says, is that it can readily be extended to separate two sets by more than one plane or surface.

2. Pattern separation by convex programming,[2]

The basic problem considered in this paper by J. B. Rosen is the same as that of section 1. However, the approach to the problem is different and perhaps more complicated. Computational results are included; something lacking in Mangasarian's paper.

We summarize the techniques presented in the paper. Suppose that $A_1,\ldots,A_k$ are sets of patterns (point sets) in E^n. We wish to partition E^n into regions such that each region contains at most one of the A_i. The author considers two techniques.

(i)Given two pattern sets A_1 and A_2, the author shows that in order that A_1 and A_2 be linearly separable it is necessary and sufficient that a certain convex quadratic programming problem be solvable. Moreover, if A_1 and A_2 are linearly separable, then the author determines the distance between A_1 and A_2 and constructs the unique hyperplane which determines this distance. Extensions to k pattern sets are given.

(ii)The second technique or problem which the author considers is that of enclosing one pattern set in a "minimum" ellipsoid. Rosen defines what he means by "minimum" and shows that such an ellipsoid is unique.

In the last section of his paper, Rosen gives computational results achieved on certain problems.

The theoretical details of Rosen's paper are somewhat more complicated than that of section 1. We summarize these details here, again omitting proofs as in section 1.

For linear separability, the ideas are similar to those of Mangasarian, except that Rosen uses convex programming rather than linear programming to determine linear separability.

By a convex programming problem, Rosen means the minimization of a convex function subject to linear constraints. Given two point sets P_1 and P_2, we say P_1 and P_2 are linearly separable if and only if there exists a hyperplane (plane in the terminology of section 1) $H = H(z,\alpha) = \{p \in E^n \mid p'z = \alpha\}$ such that P_1 and P_2 lie on opposite sides of H, where z is an n-dimensional column vector in E^n, α is a real number, and $'$ denotes transpose. Through a series of substitutions and generation of equivalent problems, the author proves the following theorem.

Theorem. P_1 and P_2 are linearly separable if and only if the convex quadratic programming problem

$$\sigma = \min_y \{1/4 \cdot \Sigma_{i=1}^n y_i^2 \mid Q_1'y \geq e_1;\ -Q_2'y \geq e_2\}$$

has a solution. If P_1 and P_2 are linearly separable, then the distance δ between them is $\delta = 1/\sqrt{\sigma}$, and a unique vector $y_0 = \binom{X_0}{\beta_0}$ achieves the minimum σ. The separating hyperplane is given by $H(x_0,\beta_0) = \{p \in E^n \mid p'x_0 = \beta_0\}$

Although it is somewhat detailed, an explanation of the notation is in order.

Let P_1 be a point set; that is, a set of patterns. We think of each pattern as being a point $\tilde{p}_{1j}$ in E^n, where

$$p_{1j} = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \cdot \\ \cdot \\ \cdot \\ a_{nj} \end{bmatrix}$$

Suppose that P_1 has m_1 elements and write P_1 as the matrix whose j^{th} column is p_{1j}. Thus P_1 is an $n \times m_1$ matrix. Similarly for P_2, another point set. The distance δ between P_1 and P_2 is Euclidean distance; Rosen claims that this distance will be the maximum value of γ (real number) for which a hyperplane $H(z,\alpha)$ exists such that

$$P_1'z \geq (\alpha + 1/2\,\gamma)\, e_1$$

$$P_2'z \leq (\alpha - 1/2\gamma)\, e_2$$

$$\|z\| = 1 \quad \text{(Euclidean norm)}$$

where ' denotes transpose and e_1 and e_2 are m_1- and m_2-dimensional column vectors of ones.

Letting $z = x/\|x\|$;$\alpha = \beta/\|x\|$; $\gamma = 2/\|x\|$, and arriving at an equivalent problem to his original one, the author makes the following definitions:

$$y = \binom{x}{\beta}$$

$$q_{1j} = \binom{p_{1j}}{-1} \text{ for each } j = 1,\ldots,m_1$$

$$q_{2j} = \binom{p_{2j}}{-1} \text{ for each } j = 1,\ldots,m_2,$$

where m_2 is the number of elements p_{2j} in P_2. (Note that y, q_{1j}, and q_{2j} are (n+1)-dimensional vectors.)

Finally, define Q_1 and Q_2 to be the $(n+1)\times m_1$ and $(n+1)\times m_2$ matrices (respectively), whose j^{th} columns are q_{1j} and q_{2j} (respectively). Thus, we have the notation of the theorem.

Rosen then shows that if P_1 and P_2 are linearly separable, then <u>basic subsets</u> $\overline{P}_1 \subseteq P_1$, $\overline{P}_2 \subseteq P_2$ can be chosen such that: (i)$\overline{P}_1$ and $\overline{P}_2$ determine the the same separating hyperplane as P_1 and P_2; (ii) the distance between $\overline{P}_1$ and $\overline{P}_2$ is the same as the distance between P_1 and P_2; and (iii) $\overline{P}_1$ and $\overline{P}_2$ have the property that removing one or more points from either $\overline{P}_1$ or $\overline{P}_2$ results in an increase in the distance between them. The author then generalizes his results to the case of k pattern sets, k a positive integer.

For the ellipsoidal separation (nonlinear separation), Rosen wishes to enclose a pattern set in a unique ellipsoid of "minimum" size. He achieves this by minimizing the sum of the squares of the ellipsoid's semi-axes. This is shown to be equivalent to the problem of minimizing the trace of a certain set of matrices. The author proves that such an ellipsoid is unique. Rosen then describes an iterative technique of determining this "minimal" ellipsoid. The procedure is to alternatively solve two convex programming problems, each of which involves the minimization of quadratic forms. Finally, Rosen shows that this procedure converges to the unique solution.

The author is quite detailed with regard to computational results of his techniques and in suggestions for overcoming computational problems.

We will not detail these here. Computational techniques and the corresponding computer programs have been developed for each of the two methods presented by Rosen ([9], [25], [6], [23]), and computational results for particular problems are given. (see [6], [23]). Computer times seem quite good, although the size of the problems Rosen considers in his computational work may account for this. Finally, Rosen makes no comparison of his techniques with others.

3. Pattern classifier design by linear programming.[3]

This paper by F. W. Smith is probably the most detailed of the three papers reviewed, as far as examples and computational techniques and results are concerned. Smith considers the same problem as that of the previous two sections. However, his work is almost exclusively for the linearly separable case; only brief mention is made that his techniques extend to the linearly inseparable case.

Smith's approach to the problem differs from that of the previous two in that he attempts to determine the separating hyperplane subject to the minimization of the mean error function. [15], [16]. Two types of the *fixed-increment adaptive method*; namely, the *steepest descent design method* [15], and the *one-at-a-time design method* [15], [17], [22] are considered. Both of these methods are iterative type techniques. The author formulates this approach (that is, minimizing the mean error function) as a linear programming problem and then compares this formulation with the two previously mentioned fixed-increment adaptive methods. Computational results, suggestions for handling special types of problems; suggestions for overcoming computational difficulties, etc. abound in the paper.

We briefly summarize the author's approach to the problem. Smith's formulation of the problem as a linear programming problem and his many comments and suggestions for special cases made in doing this are too detailed for the purposes of this report.

Let $A = \{Y_1,\ldots,Y_K\}$; $B = \{Z_1,\ldots,Z_M\}$ be two sets of patterns. As in sections 1 and 2, each Y_i and Z_j is considered as a point of E^n. We wish to find a $\widetilde{W} \in E^n$ and a real number d such that

$$Y_k^T \widetilde{W} \geq d \quad \text{and} \quad -Z_k^T \widetilde{W} \geq d \quad \forall k \tag{1}$$

(Smith calls d a scale factor [17], which for the purposes of this paper was taken to be 1.)

The <u>mean error function</u>, $\overline{h}$, is defined by

$$\overline{h} = \Sigma_{k=1}^{K} \pi_k h_k + \Sigma_{k=K+1}^{K+M} \pi_k h_k$$

where h_k is the <u>pattern error function</u> associated with

$$Y_k, \text{ if } k = 1,\ldots,K$$

and associated with

$$Z_{k-K}, \text{ if } k = K+1,\ldots,K+M,$$

and π_k is a weighting coefficient for each k.

For the fixed-increment adapter method h_k is defined by:

$$h_k = -(X_k^T W - d) \quad \text{if} \quad X_k^T W < d$$

$$= 0 \qquad\qquad \text{if} \quad X_k^T W \geq d$$

where W is an n-dimensional column vector of E^n and

$$X_i = Y_i \quad \text{for } i = 1,\ldots,K$$

$$X_{K+i} = -Z_1 \quad \text{for } i = 1,\ldots,M.$$

Note that if $\tilde{W} \in E^n$ and if $\tilde{W}$ is such that $X_k^T \tilde{W} \geq d$ for each k, then $h_k = 0$ for each k. Thus, $\bar{h} = 0$, and $\tilde{W}$ satisfies (1).

Each of the two techniques with which Smith compares his method are initiated by choosing an arbitrary (but Smith suggests it can be well chosen) W. One then proceeds by incrementing the initial W, subject to the criteria of minimizing $\bar{h}$. The main content of Smith's paper is the detailing of the formulation as a linear programming problem the problem of determining $\tilde{W}$ subject to the criteria of minimizing $\bar{h}$.

The author's primary comments on computational results are comparisons of his linear programming technique with that of the steepest descent and one-at-a-time design methods. He is quite detailed on this, giving: conjectures for when one method is better than another; calculations for the computer time required for a given, but arbitrary problem; suggestions for methods of handling certain types of problems, as well as computational results with time and accuracy comparisons for the three techniques.

The author also gives suggestions on how to eliminate some of the elements in the pattern sets in order to reduce computer time, but still arrive at the same, or nearly the same, $\tilde{W}$ as one gets using all the patterns.

Finally, the author comments that he thinks his techniques should extend to the nonseparable case; however, all detailed computational results are for the linearly separable case.

While it is not our purpose to judge the merits of these linear programming type approaches with regard to the pattern recognition problems of MSC and NASA, some comments can be made.

While a nonstatistical approach to the pattern recognition problems of MSC and NASA is somewhat questionable, there may still be some partial utilization of such an approach.

An application of theorem 3 of section 1 might be useful for considering pattern sets that one suspects to be linearly separable. Mangasarian claims this to be an immediate way of determining linear separability. The techniques suggested in section 2 have the advantage over those of section 1 in that commuter programs have already been developed for them. The idea of enclosing a pattern set in a minimal ellipsoid is applicable in the linear inseparable case and perhaps would have application in, at least, special problems. The approach suggested in section 3 is different than those of sections 1 and 2, and appears to perhaps have more potential than the first two. Computer programs have also been developed for this technique.

FOOTNOTES

[1] Mangasarian, O. L. "Linear and nonlinear separation of patterns by linear programming", Operations Research Soc. of America Journal, 13, No. 3, 444-452 (1965).

[2] Rosen, J. B., "Pattern separation by convex programming", Journal of Math. Analysis and Applications, 10, 123-134 (1965).

[3] Smith, F. W., "Pattern classifier design by linear programming", IEEE Trans. On Computers, vol.C-17, No. 4, 367-372 (1968).

BIBLIOGRAPHY

1. Albert, A., "A mathematical theory of pattern recognition," Ann. Math. Stat. 34, 284 - 299 (1963).

2. Bellman, R., Introduction to Matrix Analysis, McGraw - Hill, New York (1960).

3. Braverman, E., "Experiments in training a machine to recognize visual images, Parts 1 and 2," translated from Avtomat. i Telemekh. 13, 349 - 364 (1962). Automat. Expr. 4, No. 8, 31 - 33(1962); ibid. 4, No. 9, 34 - 40 (1962).

4. Charnes, A., "Some fundamental theorems of perception theory and their geometry," in Computer and Information Sciences, J. T. Tou and R. H. Wilcox (eds.), Spartan Books, Washington, D. C. (1964).

5. Fan, K., "Some inequalities concerning positive definite matrices," Proc. Cambridge Phil. Soc. 51, 414 - 421 (1955).

6. Fisher, D., "Minimum Ellipsoids," Tech. Rept. 31, Computer Science Division, Stanford University (1963).

7. Gale, D., "The basic theorems of real linear equations, inequalities, linear programming, and game theory," Nav. Res. Log. Quart. 3, 193 - 200 (1956).

8. Gass, S., Linear Programming, McGraw - Hill, New York (1958).

9. Gradient projection 7090 program. SHARE distribution #1399.

10. Greenberg, H. and Konheim, A., "Linear and nonlinear methods in pattern classification," IBM J. Res. and Devel. 8, 299 - 307 (1964).

11. Hadley, G., Linear Programming, Addison - Wesley, Reading, Mass. (1962).

12. Highleyman, W.,"A note on linear separability,"IRE Trans. on Electronic Computers, 777 - 778 (1961).

13. ________, "Linear decision functions with application to pattern recognition," Proc. IRE 50, 1501 - 1515 (1962).

14, Kiefer, J., "Optimum experimental designs," J. Roy. Statist. Soc. B, 11, 272 - 314 (1959).

15. Koford, J., "Adaptive pattern dichotomization," Stanford Electronics Labs., Stanford, Calif., Rept. SEL - 64 - 048, TR 6201 - 1 (1964).

16. Koford, J. and Groner, G., "The use of an adaptive threshold element to design a linear optimal pattern classifier," IEEE Trans. Information Theory, vol. IT - 12, 42 - 50 (1966).

17. Mays, C., "Effects of adaptation parameters on convergence time and tolerance for adaptive threshold elements," IEEE Trans. Electronic Computers, vol. EC - 13, 465 - 468 (1964).

18. ____________, "The boundary matrix of threshold functions," IEEE Trans. Electronic Computers, vol. EC - 13, 65 - 66 (1965).

19. Minnick, R., "Linear - input logic," IRE Trans. Electronic Computers, vol. EC - 10, 6 - 16 (1961).

20. Muroga, S. "Lower bounds of the number of threshold functions and a maximum weight," IEEE Trans. Electronic Computers, vol. EC - 14, 136 - 148 (1965).

21. Nilsson, N., Introduction to Theory of Trainable Pattern Classifying Machines, Stanford Research Institute, Menlo Park, California (1964).

22. ________, Learning Machines, McGraw - Hill, New York (1965).

23. Pipberger, H., "Use of computers in interpretation of electrocardiagrams", Circ. Res.11, 555 - 562 (1962).

24. Ralston, A. and Wilf, H., Mathematical Methods for Digital Computers, Wiley, New York (1965).

25. Rosen, J., "The gradient projection method for nonlinear programming. Parts I and II.," J. Soc. Ind. Appl. Math. 8, 181 - 217 (1960); ibid. 9, 514 - 532 (1961).

26. Rosenblatt, F., Principles of Neurodynamics, Spartan Books, Washington, D. C. (1961).

27. ___________, "On the convergence of reinforcement procedures in simple perceptions," Cornell Aeronautical Laboratory Report, #VG - 1196 - G - 4, Buffalo, New York (1960).

28. Singleton, R., "A test for linear separability as applied to self-organizing machines," in Conference on Self-Organizing Systems - 1962, M. C. Yovits, G.T. Jacobi, and G. D. Goldstein(eds.), Spartan Books, Washington, D. C. (1962).

29. Smith, F., "Automatic HF signal classification by the method of moments", Sylvania Electronic Systems - West, Mountain View, California, Rept. EDL - M1037 (1967).

30. Widrow, B. and Angell, J., "Reliable, trainable networks for computing and control," Aerospace Eng. 11, 78 (1962).

N73-29575

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF HOUSTON HOUSTON, TEXAS

CLUSTER SEEKING TECHNIQUES IN PATTERN CLASSIFICATION

B. J. BARR
JUNE 1972

PREPARED FOR
EARTH OBSERVATION DIVISION , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

I-1

CLUSTER SEEKING TECHNIQUES
IN
PATTERN RECOGNITION

Report # 10

Dr. Betty J. Barr
Department of Industrial
And Systems Engineering
University of Houston

June 1972

Contract NAS-9-12777

INTRODUCTION

A cluster seeking technique is a method of dividing data into subsets, called clusters. These clusters contain data points that are "similar" to each other and "different" from the elements of other clusters. The methods for determining the clusters differ in a variety of ways.

Basically these methods all stem from the inadequacy of the most commonly used statistics (the overall mean, covariance, and correlation) when the distribution is non-Gaussian. It is relatively easy to construct data sets which, when plotted, appear quite different but whose covariance matrices, for example, are identical [3].* Moreover, the classes into which it is desired to sort data are usually those established by human perception, and it has been argued that the usual statistical descriptors have little perceptual significance [30].

Notation:

In the sequel, X^j will denote the j-th data vector or pattern. N will be the total number of patterns. If the patterns are members of a finite dimensional vector space,

* Bracketed references refer to entries in the bibliography.

D will denote the dimension and $X^j(i)$ will denote the i-th component of X^j as a member of E^D. S_{ij} will denote the similarity coefficient between the i-th and j-th patterns, and d_{ij} will denote the "distance" (not necessarily a metric) between them.

Since the measure of similarity is crucial to all the cluster seeking techniques, some of the various measures that have been used are summarized in Table 1 [3,53]. Some of the algorithms may be applied with any of the measures, while others are more specific.

The various cluster seeking techniques have been broken down into seven categories: [3]

1.Probabilistic

2.Signal Detection

3.Clustering

4.Clumping

5.Eigenvalue

6.Minimal mode seeking

7.Miscellaneous

In the following sections of this report, each category will be described and one or more algorithms of that type will be presented.

TABLE 1

MEASURES OF SIMILARITY

Dot Product:	$S_{ij} = X^{i} \cdot X^{j}$
Similarity Ratio:	$R_{ij} = X^{i} \cdot X^{j}$ $S_{ij} = R_{ij}/(R_{ii}+R_{jj}-R_{ij})$ $d_{ij} = -\log S_{ij}$
Weighted Euclidean Distance:	$d_{ij} = \sum_{k=1}^{D} w_k (X^{i}(k)-X^{j}(k))^2$
Unweighted Euclidean Distance:	$d_{ij} = \sum_{k=1}^{D} (X^{i}(k)-X^{j}(k))^2$
ℓ_1 Distance:	$d_{ij} = \sum_{k=1}^{D} \lvert X^{i}(k)-X^{j}(k) \rvert$
Component Correlation:	$S_{ij} = \sum_{k=1}^{D} \sum_{l=1}^{D} r_{kl} [1- \lvert X^{i}(k)-X^{j}(k) \rvert] \cdot$ $[1- \lvert X^{i}(l)-X^{j}(l) \rvert] \cdot$ $[1-2 \lvert X^{i}(l)-X^{j}(l) \rvert]$ where r_{kl} is correlation coefficient between components k & l.
Normalized Correlation:	$S_{ij} = X^{i} \cdot X^{j}/ (X^{i} \cdot X^{i})(X^{j} \cdot X^{j})$
Coefficient of Correlation:	$S_{ij} = \dfrac{\sum_{k=1}^{D} (X^{i}(k)-u_k)(X^{j}(k)-u_k)}{\sqrt{\sum_{k=1}^{D} (X^{i}(k)-u_k)^2 \sum_{k=1}^{D} (X^{j}(k)-u_k)^2}}$ where u_k is the overall mean of the k-th component.

TABLE 1 (Continued)

Coefficients of Association: For binary data, n will denote number of, a capital subscript denotes '1' and a small subscript denotes '0'.

1. $n_{JK}/(n_{JK}+n_{Jk}+n_{jK})$
2. $(n_{JK}+n_{jk})/D$
3. n_{JK}/D
4. $2n_{JK}/(2n_{JK}+n_{Jk}+n_{jK})$
5. $2(n_{JK}+n_{jk})/(2(n_{JK}+n_{jk})+n_{Jk}+n_{jK})$
6. $n_{JK}/(n_{JK}+2(n_{Jk}+n_{jK}))$
7. $(n_{JK}+n_{jk})/(n_{JK}+n_{jk}+2(n_{Jk}+n_{jK}))$
8. $n_{JK}/(n_J+n_K-2n_{JK})$
9. $(n_{JK}+n_{jk})/(n_{Jk}+n_{jK})$
10. $\frac{1}{4}((n_{JK}/n_J)+(n_{JK}/n_K)+(n_{jk}/n_j)+(n_{jk}/n_k))$
11. $\frac{1}{2}((n_{JK}/n_J)+(n_{JK}/n_K))$
12. $n_{JK}/\sqrt{n_J n_K}$
13. $n_{JK}n_{jk}/\sqrt{n_J n_K n_j n_k}$
14. $(n_{JK}+n_{jk}-n_{Jk}-n_{jK})/D$
15. $(n_{JK}n_{jk}-n_{Jk}n_{jK})/(n_{JK}n_{jk}+n_{Jk}n_{jK})$
16. $(n_{JK}n_{jk}-n_{jK}n_{Jk})/(n_J n_K n_j n_k)^{\frac{1}{2}}$

PROBABILISTIC

Probabilistic cluster seeking techniques are primarily analytical studies. The probability of occurance of a pattern is estimated and then a weighted combination of patterns is used to estimate probability distributions.

The following algorithm developed by Fralick is typical [22]:

Suppose there are M possible classes $w_1, \ldots, w_M$, and associated with each is a conditional probability density $p(X/w_i)$ which is known except for a single parameter θ^i, that is, assume $p(X/w_i, \theta^i)$ is known. Assume also that the a priori probabilities of occurence $p(w_i)$ are known, that the a priori distribution of θ^i, $p_o(\theta^i)$ is known, and that θ^i can assume only a finite number of values. Then the desired density can be determined as follows:

$$p_k(X_{k+1}/w_i) = \int p(X_{k+1}/w_i, \theta^i) p_k(\theta^i)\, d\theta^i$$

where

$$p_k(\theta^i) = p_{k-1}(\theta^i) \cdot \left[\frac{p(X_k/w_i, \theta^i) p(w_i) + \sum_{j \neq i} p_{k-1}(X_k/w_j) p(w_j)}{\sum p_{k-1}(X_k/w_j) p(w_j)} \right]$$

For the case of an unknown signal in noise, he proves that $p_k(X_{k+1}/w_i) \rightarrow p(X/w_i)$. However, the amount of computation and storage required is considerable, particularly for multivariate problems. Moreover, in the case where the class a priori probabilities are all the same. the initial selection of the probability distributions for the various classes must be different for "learning" to occur [21].

Other probabilistic techniques are discussed in [17,45,16]

SIGNAL DETECTION TECHNIQUES

Signal detection techniques grew out of a desire to detect unknown signals in noise. The final decision is based on correlation detection to estimate parameters of a matched filter.

The following algorithm of Jakowatz is typical [29]:

A sample waveform M is stored in the memory of a correlation detection device. When the dot product $b(t) = M(t) \cdot X(t)$ of the incoming wave X exceeds a threshold $b_T(t)$, the waveform in memory is modified as follows:

Let t_i be the time at which memory is changed. Then $M(t) = (gM(t_{i-1})e^{-ds}+X(t_i))/(g+1)$ where g depends on a capacitor ratio, d is a time constant associated with the memory device, and $s = t-t_{i-1}$ for $t_i \geq t > t_{i-1}$.
The threshold grows with successful detection and decays with failure to detect.

Other signal detection techniques may be found in [25,54,51]. All of these are primarily used for signal detection and as presently conceived their utility outside this area seems limited. One severe problem is the use of energy detection to start the process going. There is a definite thresholding effect for weak signals, and apparently a minimal adaptable signal, which may be a function of signal waveform.

CLUSTERING TECHNIQUES

Clustering techniques can be characterized by sorting of patterns using multiple cluster points. Tentative assignments are made to clusters and these assignments improved until the centroid of the cluster adequately describes the data. Since these techniques vary in a number of ways, several algorithms will be presented here.

Okajima proposed the following algorithm for use with electrocardiagram data [43]:

Step Number — Step Description

1 The data vectors are arranged in random order and a bank of memory filters $\{M\}$ is initialized to zero.

2 The incoming data vector X is selected and weighted (if desired).

3 The correlation $X \cdot M/((X \cdot X)(M \cdot M))$ with each used memory filter is computed and a memory filter M is selected which gives the maximum correlation.

4 If this maximum correlation exceeds a predetermined threshold, the filter is modified by the rule: If X_i is the i-th pattern entering the same filter M, then $M = (1/i)(X_1+X_2+\ldots+X_i)$.

5 If not, the data vector goes into a new filter.

6 Repeat 1-5 until all data has been examined.

The algorithm depends on the threshold, the weighting, and the order in which the pattern vectors are selected. Algorithms very similar to this have recently been proposed using different measures of similarity [40,48,59]. These "one-pass" techniques are definitely time-savers [34].

Sebestyen is concerned with computing a probability distribution based on the sample data [49,50]. A pattern is selected and compared with existing cluster centers. The measure of similarity is a weighted Euclidean distance with the weight depending on both the component and the cluster. The minimum distance of the pattern from a cluster point is compared with two thresholds. If the smaller threshold is not exceeded the pattern is added to that cluster and a new mean for the cluster is computed. If the larger threshold is exceeded, a new cluster is formed using that pattern as its centroid. If the pattern distance is between the two thresholds, the pattern is temporarily rejected and will be considered later on in the process. This algorithm is computationally complex, and very sensitive to the weight factors.

The ISODATA program of Ball and Hall [5,6,7] has recently undergone comprehensive study [27,31,32,33].

The version presented here is the "final" version recommended in [34].

The ISODATA Algorithm

```
     0. Initialize
Loop 1. CLASSify and calculate STATistics
     2. Change cluster structure:
        2.1 DELETE
        2.2 If iteration is a split (S) iteration, SPLIT, and
            go to Step 3; otherwise continue.
        2.3 COMBINE
    3.  If iteration is the final one in the SC sequence, STOP;
        otherwise, go to Loop for the next iteration.
```

Before the subroutines mentioned above can be explained, some notation must be developed:

SGMAX — Maximum standard deviation allowed in a cluster, larger than which the cluster is split.

DLIM — Minimum distance between two clusters, less than which they are combined.

NCLUSTR — Number of clusters at any particular iteration.

NDATA(I) — Number of data points in the i-th cluster at any particular iteration.

NMIN — Minimum number of points in a legitimate cluster.

NTOTAL — Total number of data points in the input.

SC sequence Split(S) and combine(C) sequence.

u_j^i, s_j^i Mean and standard deviation of the i-th cluster along the j-th coordinate.

Initialize: Input values for SGMAX, DLIM, NMIN, SC sequence, and a starting procedure. The default option sets SGMAX = 4.5, DLIM = 3.2, and NMIN = 20. If no starting procedure is specified, the SC sequence = SSSCSCSCSCSCCC, NCLUSTR = 1, and $u_j^1 = 0$, $j = 1,\ldots,D$.

CLASS and STAT: From the previous iteration there are left NCLUSTR cluster centers. The subroutine reclassifies the data points to their respective closest reference points, using ℓ_1 distance. The means and standard deviations of these new clusters are iteratively accumulated at the same time the points are assigned.

DELETE: This subroutine deletes the existence of a cluster when it contains less than a prespecified minimum number of points (NMIN).

SPLIT: This subroutine splits a cluster along the j-th coordinate by creating two clusters with centers at $(u_1^i,u_2^i,\ldots,u_j^i \pm s_j^i,\ldots,u_D^i)^T$ if (i)Its standard deviation along the j-th coordinate is larger than SGMAX; and if (ii)It has more than

2(NMIN+1) data points.

COMBINE: This subroutine combines two clusters if the distance between them:

$$d(u^p,u^q) = \sum_{j=1}^{D} (1/s_j^p s_j^q)(u_j^p - u_j^q)^2$$

is less than DLIM.

Although reasons are given in [34] for the use of three different distance measures in the same program (computational simplicity), the logic behind mixing ℓ_1 for distance from data to cluster, ℓ_2 for standard deviation of cluster, and a weighted ℓ_2 for distance between clusters, is difficult to follow. The user specified thresholds have a great influence on the clusters formed, although the iterative nature of the algorithm somewhat ameliorates this.

CLUMPING

In these techniques a single pair of patterns is selected as a nucleus for a clump of patterns. Other patterns are assigned to this clump on the basis of the similarity measure. Genrally speaking, these techniques require the calculation of all pairwise similarity coefficients, forming a similarity matrix, and some of these must be recalculated after each new combination.

Several "clustering by linkage" techniques have been suggested [39,52,53]. All involve first calculating a similarity matrix. The nucleus of a cluster is established using those two patterns with the highest similarity coefficient. Then patterns are added to this nucleus one at a time. Single linkage calls for admitting a pattern if its similarity coefficient with any one member of the cluster exceeds a threshold. Iterative improvement is provided by recalculating the mean similarity both within groups and between groups. Complete linkage requires that a pattern joining a cluster must have a value above the threshold with all members of the cluster. If there is a choice, it should be made first to give the larger group, second to have fewest residual patterns, and third to give the highest average similarity coefficient. After each iteration a new similarity matrix is calculated using the means of the clusters. Clustering by average linkage bases admission on the average similarity of that pattern to all members of the cluster. If an admission would lower this average similarity by more than .03 (an empirically determined value) the pattern should not be admitted.

Rogers and Tanimoto use a function related to information theoretic entropy as a criterion for clustering binary patterns [46]. Their algorithm is as follows:

Step Number   Step Description

1 Compute $R_{ij} = X^{i} \cdot X^{j}$

$$S_{ij} = R_{ij}/(R_{ii}+R_{jj}-R_{ij})$$

$$H_i = \sum_{j=1}^{N} (-\log_2 S_{ij})$$

R_i = # of pattern vectors j such that $R_{ij} > 0$.

2 Now rank all patterns, first in order of R_i, and then, for those with equal R_i, in order of H_i.

3 Let $d_{ij} = -\log_2 S_{ij}$ and form the distance matrix M.

4 Let $E_n(M) = -\frac{1}{2} \sum_{ij}{}' (d_{ij}/T_n(M))(\log_2(d_{ij}/T_n(M)))$ where $T_n(M) = \frac{1}{2}(\sum_{ij}{}' d_{ij})$ and $\sum{}'$ denotes summation of the finite terms of M, after repeated rows and columns have been deleted.

5 Let g be the number of zeros above the diagonal of M and h the humber of infinite terms above the diagonal which are not in the same

row or column as one of the g zeros. Set
$F_n(g,h) = \log_2((n-g)(\frac{1}{2})(n-g-1) - h)$ and
$U_n(M) = 1 - (E_n(M)/F_n(g,h))$.
U_n is a measure of heterogeneity.

6 If $U_n(M)$ is near one, clusters do exist and the process proceeds by selecting X^{io}, the highest ranked pattern, and X^{jo}, the second highest.

7 Consider all patterns X^j with $d_{io,j} < d_{io,jo}$, and determine U for this subset. If U is small, add X^{jo} to the clump and recompute U for the larger clump. Continue until U takes a large jump, indicating the end of a clump. Remove those cases nearest the edge and start a new clump.

Bonner proposes two methods [11]. They are both of sufficient interest to be presented here.

The first method involves computation of a similarity matrix. This matrix is then "thresholded" by comparing each entry with a predetermined constant (eg. .45). If the threshold is exceeded, a one is entered in the corresponding position in the new matrix. Otherwise a zero is entered. This new similarity matrix is then manipulated according to the following algorithm:

<u>CLUSTER I</u>: The similarity matrix is now regarded as a set of binary patterns and its similarity matrix

is formed using the following measure:

If C_{ij} is the number of ones the i-th and j-th pattern have in common, then

$S_{ij} = C_{ij}/(C_{ii}+C_{jj}-C_{ij})$.

This new matrix is then thresholded. This process of taking the similarity matrix of the similarity matrix may be repeated as often as desired, hopefully until stabilization is reached.

CLUSTER II: The input here may be the original matrix, or the result from CLUSTER I. First "tight" clusters are formed in which all members are similar and no nonmember is similar to all. Then using the tight clusters a set of "core" clusters is located in which no object is in more than one cluster and all objects in a cluster are similar. Finally, a cluster adjustment program attempts to build around the cores.

Algorithm for tight clusters: This algorithm keeps track of three things at each level of buildup:

1. The set of objects (A_i) in the cluster to this point.
2. The set of objects (C_i) which could possibly be added to A_i to further increase the cluster.
3. The number (L_i) of the last object C_i to be considered for addition to the cluster.

These three things are stored for each i which is smaller than or equal to the present i. Also needed is the similarity matrix where $S_{L_i} = \{X^j \mid S_{L_i,j} = 1\}$.

Step Number — Step Description

1 $i = 1$, C_1 = all objects, $A_1 = \emptyset$, $L_1 = 1$.

2 If $X^{L_i} \notin C_i$, $L_i = L_i + 1$ and go to Step 5. Otherwise continue to step 3.

3 $C_{i+1} = C_i \cap S_{L_i} - \{X^{L_i}\}$, $A_{i+1} = A_i \cup \{X^{L_i}\}$

4 $L_{i+1} = L_i + 1$, $i = i+1$

5 Is L_i greater than the number of the last possible object? If so, go to Step 6, if not go to Step 2.

6 If $C_i = \emptyset$, store A_i as cluster. If not, A_i is a subset of a cluster already found and so need not be stored. In any event, $T = A_i$.

7 $i = i-1$. If $i = 0$, STOP. Otherwise, go to Step 8

8 $C_i = \{X^j \mid X^j \in C_i$ and $j > L_i\}$. Is $C_i \subseteq T$? If yes, go to 7. If not, go to 2.

Algorithm for core clusters: Let i be the alternative index and j the buildup level.

Step Number — Step Description

1 Find the tight cluster having the largest number of members and store it as the first core. Set $j = 1$. If there is a tie for the largest cluster, go to Step 9.

2 i=1

3 Find the tight cluster having the most members different from the total set of members in all stored "core" clusters of alternative i of buildup level j. Call this its difference set. Call the cluster itself a maximum distance cluster.

4 If this difference set is larger than that of any of the other alternatives of buildup level j yet considered, drop these alternatives, consider only the present alternative and go to Step 5. If it is smaller, drop the present alternative and go to Step 6. If it is the same as that of other alternatives of buildup level j, consider all still as possible alternatives and go to 5.

5 If there is only one maximum distance cluster, store its difference set as the next core cluster for alternative i and go to 6, if there is a tie, go to 8.

6 Have all alternatives of buildup level j been considered? If so, go to 7. If not i = i+1 and go to 3.

7 For any given alternative, are all possible objects in one of the core clusters? If so, print out the core clusters for all

alternatives and STOP. Otherwise, j = j+1 and go to 2.

8 Of the set of clusters in the tie, pick the smallest and store its difference set as a core for alternative i and go to 6. If there is still a tie, go to 9.

9 Form a dissimilarity matrix for the clusters in the tie, where two clusters are considered dissimilar if their difference sets contain no common member. Find all the tight clusters for this matrix. Each tight cluster here will represent a set of the original tight clusters whose difference sets are disjoint, Store the largest such set of difference sets as a set of core clusters. If there is a tie, all alternatives will be followed in the hope that subsequent choices of cores will favor some alternatives over others. They are therefore added to the alternative list of the next level of buildup. Note that it is possible that more than one core will be added to each alternative by Step 9. By convention, this addition is still treated as one level of buildup. Go to step 6.

Cluster Adjustment Program: Specify a criteria for judging a cluster as "large".

Step Number	Step Description
1	i = 1
2	j = 1
3	Consider the j-th member of cluster i: Compute from the similarity matrix the number of objects in the first large cluster to which this j-th object is similar. Divide this by the number of objects in the first large cluster to produce a percentage match of the j-th object to the first large cluster. Compute such a percentage match of the j-th object with each of the large clusters and with each of the small clusters already considered.
4	Are any of these matches above some threshold (eg. .8)? If yes, go to 5, if not go to 6.
5	Delete the j-th object from the small cluster and put it into the cluster offering the best match.
6	j = j+1. Have all members of cluster i been considered? If no, go to 3, if yes go to 7.

7 $i=i+1$. Have all clusters been considered? If no, go to 2. If yes, go to 8.

8 Iterate this entire procedure as many times as desired with the hope that stability will be obtained.

9 Compute for all remaining pairs of clusters C_i and C_j, a measure of their interaction:

$$I_{ij} = (1/N_i N_j) \sum_{a=1}^{N_i} \sum_{b=1}^{N_j} S_{ab}$$

where N_k is the number of objects in the k-th cluster, $S_{ab} = 1$ if object a in C_i is similar to object b in C_j, and $S_{ab} = 0$ otherwise. A measure of value for the i-th cluster is then

$$V_i = I_{ii} - (1/N_R) \sum_{j=1}^{N_R} I_{ij}$$

where N_R is the number of clusters other than the i-th. For the whole set

$$V = (1/N_R+1) \sum_{i=1}^{N_R+1} V_i \ .$$

Bonner admits that this procedure becomes difficult as the number of clusters becomes large and when "ties" occur frequently in the core building subprogram. He presents the following rather ingenious alternative. He states that he has a program for this algorithm which can handle 2000 objects of 360 binary variables each and which averages 3 minutes of computer time.

Consider a cluster of N_k patterns. Define

$G_k = \sum_{i=1}^{D} ((u_i)_k-u_i)^2/(s_i^2/N_k)$ where $u_i = (1/N) \sum_{j=1}^{N} X^j(i)$,

$(u_i)_k = (1/N_k) \sum_{j \text{ over } j_o} X^j(i)$ where j_o is the index set for the objects in the cluster,

$s_i^2 = \sum_{j=1}^{N} (X^j(i)-u_i)^2/(N-1)$.

Using a χ^2-distribution, calculate the probability P that $G \geq G_k$.

Step Number Step Description

1 Pick an object to act as a cluster center.

2 Find the similarity coefficient between this pattern and all others. All objects more similar than an arbitrary threshold T are considered to be in the crude cluster.

3 Compute the centroid of this cluster. Compute the expected number of clusters rarer than this to be found in an uncorrelated population, as given by $\binom{N}{N_k}$ P. If this number exceeds a preset number K, go to 7. Otherwise, "hill-climbing" will be done in 4.

4 Find the similarity between the centroid and all other objects using the following: Add up the weights $((u_i)_k-u_i)^2/(s_i^2/N_k)$ of all attributes i where there is a bit match between an object and the centroid.

If this sum is greater than a certain percentage Y of G_k, then the object is judged as similar to the centroid. All objects similar to this centroid are now members of the new cluster.

5 Is this cluster the same as the last? If so go to 6, otherwise to 3.

6 Store the stable clusters as final clusters. Delete each member of the cluster from consideration as a future cluster center.

7 Have all allowable objects been used as cluster centers? If not, pick one and go to 2; if yes, STOP.

Bonner used both these algorithms on some disease symptom data and got similar results. The same results, with one notable exception, were also found through a standard factor analysis.

Ward describes an algorithm which repeatedly combines those patterns which maximally increase an "objective function" [60,61]. This function is supposed to measure the remaining information when two sets are united into one (assuming maximal information corresponds to singleton sets). For example, when the patterns are grouped into one, he suggests ESS $= \sum_{i}^{n} X^{i} \cdot X^{i} - (1/n)(\sum_{i}^{n} X^{i} \cdot \sum_{i}^{n} X^{i})$. It is necessary to know in advance the number of clusters

N_c to be formed. Let p_{i-1} be the smaller and q_{i-1} be the larger of the two numbers used to identify the subsets $S(p_{i-1},i)$ and $S(q_{i-1},i)$ at the i-th stage. Then $S(p_{i-1},i-1) = S(p_{i-1},i) \vee S(q_{i-1},i)$, and the associated objective function is $Z(p_{i-1},q_{i-1},i-1)$.

Step Number   Step Description

1    $k = N$

2    $Z(p_{k-1},q_{k-1},k-1)$ = initial value worse than all others, i = smallest active index.

3   j = first active index $> i$.

4   Compute $Z(i,j,k-1)$

5   Is $Z(i,j,k-1) > Z(p_{k-1},q_{k-1},k-1)$? If yes go to 6, if no, go to 7

6    $Z(p_{k-1},q_{k-1},k-1) = Z(i,j,k-1)$, $p_{k-1} = i$, $q_{k-1} = j$.

7   Is j = last active index? If not, set j = next higher active index and go to 4. If yes, go to 8.

8   Is i = next to last active index? If not, set i = next higher active index and go to 3. If yes, go to 9.

9   Identify the union by p_{k-1} and make q_{k-1} inactive.

10   Is $k = N_c+1$? If so, stop. If not, $k = k-1$ and go to 2.

Fisher examines all possible partitions on the real line and selects that partition which minimizes the weighted square distance from the cluster center [19]. For an ordered collection of patterns he proves two lemmas that allow him to reduce the number of partitions he must consider. He has a program for his algorithm for $N \leq 200$ and the number of clusters (assumed known) is less than 10. He remarks that even with these size restrictions there are still a number of sources of difficulty.

Sawrey proposes the following for psychological data [47].

Step Number	Step Description
1	Form the distance matrix
2	Select potential clusters:
2.1	Decide on a similarity threshold (eg. $\Sigma(s_j/2)^2$ where s_j is the standard deviation of the j-th component).
2.2	Construct a chart of the N patterns, listing with each all others that are similar.
2.3	Select as a nucleus any two or more, beginning with the largest number of similar patterns.
2.4	When a pattern or one similar to it is selected, it is deleted from the chart.
3	Select dissimilar clusters:
3.1	Decide on dissimilarity threshold

(eg. Σs_j^2).

3.2 Construct distance matrix of selected patterns for nucleus group.

3.3 Sum all columns, selection beginning with the largest. When selected, all patterns that are not dissimilar are removed. Continue until all are gone.

4 Compare and add remaining patterns as follows:

4.1 Find centroid of each group.

4.2 Make a chart of all possible additions (those that are not dissimilar)

4.3 Find distance between possible additions and nucleus.

4,4 Set several thresholds: $(1/4)\Sigma s_j^2$, $(1/3)\Sigma s_j^2$, $(1/2)\Sigma s_j^2$, $(3/4)\Sigma s_j^2$, Σs_j^2.

4.5 Add those patterns closer than the first threshold, except that if a pattern could be added to more than one group, it should not be added to any.

4.6 Recompute centroid, determine the new distances, and add those less than the second threshold.

4.7 Continue until all thresholds have been used.

McQuitty has a somewhat more stringent definition of a cluster or "type" [38]. A type is a set such that every one of its members is more like the other members of the type than like any other nonmember. In order to locate these types, first the similarity matrix M must be calculated. The entries of the matrix are then listed in order, omitting the diagonal, from the largest to the smallest.

Step Number Step Description

1 Let $T_1, T_2, \ldots$ be the types found so far.

2 Let $C_1, \ldots$ be the categories "expanded" in finding the types $T_1, T_2, \ldots$, which have not qualified as types.

3 Let $T^1, T^2, \ldots$ denote the first, second, etc. times a category requalifies as a type.

4 Let X^a and X^b be the two patterns corresponding to the highest similarity score.

5 Since $S_{ab} > S_{ay}, S_{by}$ for any other pattern X^y, X^a and X^b form a dyadic type T_1.

6 Let X^c and X^d be the pair corresponding to the second highest similarity coefficient.

7 If either X^c or X^d is X^a or X^b, assign all to C_1.

8 If not, then X^c, X^d constitute T_2.

9 Let X^e and X^f be the pair for the next highest coefficient.

10 If X^e is any of the preceding patterns, assign it to the corresponding C_i.

11 Repeat for X^f.

12 If X^e is in one category and X^f in another, then neither category can qualify as a type, so combine the two categories into one.

13 If X^e or X^f is in a category, but the other is in neither, then assign both to the category in which the one is found.

14 If both X^e and $X^f$are in the same category, leave them alone.

15 If either 13 or 14 occured, the categories must be continued.

16 If neither X^e nor X^f are in the previous categories, start a new category C_j with them in it.

17 Repeat for all ranked patterns in the order of their rank with all categories operative at the time the pattern is considered.

McQuitty claims to prove his method works, but he does not provide for ties in the ranking.

Other clumping algorithms may be found in [20,44,41,1,2,36,12,35,28].

EIGENVALUE

Eigenvalue techniques, unlike the other techniques, are noniterative. They depend on calculation of a matrix associated with the pattern and determination of one or more of its eigenvalues and corresponding eigenvectors. The early efforts in this direction involve estimation of the covariance matrix followed by its diagonalization and factor analytic techniques [8,56,57,58]. Since a large number of samples are required, especially as the number of dimensions increases, the computational aspects are formidable.

Nunnally is in some sense intermediate between the clumping techniques and the eigenvalue techniques [42]. He constructs a distance matrix rather than the classical covariance matrix, but he does use the eigenvectors of the matrix to define the clusters. All patterns are examined with respect to the eigenvector basis and those with which many patterns are highly correlated are selected.

Cooper [13,14,15] and Mattson [37] both find clusters by finding the maximum eigenvalue of the covariance matrix and splitting patterns on the basis of correlation with the corresponding eigenvector. Both papers are essentially limited to the two category case.

Cooper is more analytic in that he proves, for specific distributions, that the hyperplane determined by the sample mean and principal eigenvector of the covariance matrix does define the optimal partition. However, he does depend heavily on a number of assumptions regarding the nature of the data. The cases he treats are those in which the two cluster distributions are: (1)univariate normal with the same standard deviation; (2)spherically symmetric multivariate normal with equal covariance;(3)multivariate normal, either with diagonal covariance matrices or with one mean known. He mentions that the analysis for the K category case is very complicated. Here his only result is that for K spherically symmetric distributions differing only in location, the number K can be determined from the multiplicity of the smallest eigenvalue of the overall covariance matrix. This is interesting in that much earlier Young [62] proposed the dispersion of the eigenvalues of the covariance matrix as an "index of clustering", and gave a method for determining the number of clusters based on this dispersion.

In a sense, Mattson took Cooper's idea a step further. Making no assumptions regarding the underlying distributions, he suggests the following procedure: Find A = (a_{ij}) where $a_{ij} = \sum_{k=1}^{N} (X^k(i)-u_i)(X^k(j)-u_j)$, u_i being the corresponding component of the mean. Find the largest eignevalue of A and the corresponding normalized eigenvector, w. Then use $S = \sum_{k=1}^{D} X(k)w(k)$ and a threshold T. If $S \gtrless T$, X is in case

1, if $S \angle T$, X is in case 0. For more than two categories, he suggests constructing a "network" of these linear threshold elements and using them to produce a binary code word for each class.

For those not mathematically minded, the relaxation of assumptions "make the Mattson technique particularly useful" as an "excellent example of combination of analytical and intuitive approaches" [3]. However, for those concerned with rigor, it is unavoidable to wonder at the logic of applying the method when the covariance matrix is not an adequate reflection of the data (a point which Nunnally also raises).

MINIMAL MODE SEEKING

These techniques require categorization information to work. A new mode is created only when patterns in one class are nearer to a mode of a different class. Pattern density, as such, is not used in cluster seeking.

Firschein [18] partitions classes into subclasses so that each member of a particular class is closer, in the sense of high dot product, to the centroid of its own subclass than to the centroid of any other subclass. Unlike previous procedures, this method does not require the specification of an arbitrary fixed distance as a criterion for membership in a subclass, nor is it necessary to specify the required number of subclasses beforehand.

The algorithm begins by setting subclass equal to class. The centroid of each subclass is computed. Each vector in the first subclass is dotted with every centroid to form a dot product array: $(a_{ij}) = X^{i} \cdot u^{j}$ = #components agree-#components disagree. Considering each row in the array, determine if the corresponding pattern

Class I: Has highest dot product with centroid of its own subclass.

Class II: Has highest dot product with centroid of another subclass in the same class.

Class III: Has highest dot product with centroid of another subclass of a different class.

If Case I, go to next vector.

If Case II, put vector is subclass with highest dot product and recompute the centroids and dot products for the revised subclasses. All asterisks (see below) are deleted and the procedure returns to the first vector.

If case III, an asterisk is placed next to the vector and the next vector is examined.

When all vectors in a subclass have been examined, the *vector (if any) with lowest dot product with its own subclass is chosen as centroid for a new subclass and all asterisks are deleted. Centroids and dot products are recomputed. Go back to first vector and repreat until only Class I vectors remain, or until an arbitrary number of iterations has been performed.

This technique appears useful when the pattern subclasses are linearly separable. However, some modification is necessary if classes are badly overlapped and intermixed.

Steinbuch forms subclasses if the distance between a pattern and a mode of its particular class is greater than a fixed threshold [55]. The procedure is iterated until adequate separation is achieved or other constraints are satisfied. He seems primarily concerned with a description of the "learning matrix" itself, rather than how it works and its limitations.

MISCELLANEOUS

Certain techniques do not fit neatly into any of the above categories.

The technique of Block [10] utilizes a high probability of contiguous runs of patterns in a time sequence being from the same class to adjust the machine to a particular mode. This high probability of runs provides marginal "teacher information"

Bledsoe [9] seeks to find the set of hyperplanes passing through "corridors" in the data that have maximal average distance from the patterns. An arbitrary plane passing through the patterns is selected. Distances from this plane are computed for all patterns. The average distance of the pattern from this plane is maximized by a series of iterative adjustments of the plane. This procedure is tried for several different initial starting points. The plane having maximum average starting distance is selected as the best plane. All patterns are projected onto this plane and a second plane in D-1 dimensions that maximizes distance from all patterns is sought. This appears similar to the technique of Fu [23].

Gengerelli [24] analyzes the distribution of pairwise distances between patterns. He defines a cluster as an aggregate of points in the test space such that the distance between any two points in the set is less than the distance between any point in the set and any point not in it. First, the distance matrix is constructed. Then, by applying a predetermined threshold, it is decided if each pair of patterns is a neighbor (assign 1) or a stranger (assign 0). This N-S matrix is then analyzed:

1. Add each column and augment by 1. The augmented sum is the maximum size of a cluster to which that pattern could belong.
2. Identical columns are eliminated.
3. Choose the column with the largest and next largest sum.
4. Consider the intersection of the corresponding row and column (symmetry permits row = column). If it is a 1, the new column is retained. If not, it is rejected and the next largest sum is taken.
5. Continue, at each step considering the intersection of all columns that have been kept. When all columns have been considered, the kept columns form the first cluster. This is removed and the whole process repeated.

Hartigan [26] proposed an "adding algorithm" . The idea is to draw a tree of L_{sp} levels, where each node represents a cluster. The node at any level is the parent node of the nodes (descendent) at one level lower and which are connected to the node from below.

The algorithm is:

1. Initialize the means of the nodes. Set i = 1.
2. Remove X^i from tree, modify node means.
3. Reassign X^i to nearest node at level 1, then to the nearest at level 2, and so on down the tree till level L_{sp}. Update node means.
4. Go back to 2 and repeat until all patterns have been used.
5. If the process stabilizes, stop. Otherwise, set i = 1 again and go back to 2.

This program is very adaptive in the sense that at each assignment the statistics of all relevant nodes are accordingly modified and updated. It is very easy to trace the kinship between clusters by the existence of the tree structure. Unfortunately, the end result invariably has a prespecified number of clusters equal to $2^{L_{sp}}$. Big or small clusters are indiscriminantly broken up into smaller clusters whenever more levels are allowed. Dichotomization of patterns contained in any node at any specified level (except the lowest) is always carried out. This means that patterns which should constitute a single cluster may be split and end up in nodes which do not have the same parent, making it impossible to identify the true cluster [34].

CONCLUSION

Each of the algorithms described in the preceding chapters are illustrated by one or more examples in the papers in which they are referenced. In some cases these examples are small, rigged cases where the algorithm is easy to follow and its accuracy may be judged. In other cases, the examples are of "real-life" data (classification of bees, plants, diseases) which certainly give a better feel of the practicality of the algorithm, but there is no "absolute truth" against which to examine the results. It would be interesting to apply each of the algorithms to one or more test cases and compare the results.

In relating and judging the techniques, consideration must be given to the similarity measure used, to the criterion for a cluster and to the computational complexity and amount of memory required.

The understanding of "convergence" of the methods must be regarded as minimal, particularly with nonGaussian data. It appears, from the examples, that if the data is indeed clustered, then the final clustering will tend to be unique. If, however, the data is "smeared" and "amoebic" then a greater variety of clusterings can exist. Finally, if the data is uniform, then no real stable clusters are formed--which is as it should be since no clusters in fact exist [3].

BIBLIOGRAPHY

*1. C.T. Abraham, "Evaluation of Clusters on the Basis of Random Graph Theory," unpublished report, IBM, Yorktown Heights, N.Y.

*2. ____________, "A Note on a Measure of Similarity Used in the DICO Experiment", Appendix I, Quarterly Report 3, vol. 1, Contract AF 19(626)-10.

3. Geoffrey H. Ball, "Data Analysis in the Social Sciences: What About the Details?", 1965 Fall Joint Computer Conference, AFIPS Proceedings, vol. 27, pt. 1, pp. 533-559.

**4. ____________ and D.J. Hall, "ISODATA, A Novel Method of Data Analysis and Pattern Classification," Stanford Research Institute, Menlo Park, Calif. (Apr. 1965).

**5. ____________ and D.J. Hall, "ISODATA, an Iterative Method of Multivariate Analysis of Pattern Classification," Proceedings of the International Communication Conference, Philadelphia, Pa. June 1966.

**6. ____________ and D.J. Hall, "A Clustering Technique for Summarizing Multivariate Data," Behavioral Sciences, vol. 12, #2, pp. 153-155, Mar. 1967.

**7. ____________, "Listing of CDC 6400 ISODATA program, Stanford Research Institute, Menlo Park, Calif., Feb. 1972.

8. B.M. Bass, "Iterative Inverse Factor Analysis--A Rapid Method for Clustering Persons," Psychometrika, vol. 22, no. 1, pp. 105-107 (Mar. 1957).

*9. W.W. Bledsoe, "A Corridor-Projection Method for Determining Orthogonal Hyperplanes for Pattern Recognition," unpublished report, Panoramic Research Cor., Palo Alto, Calif. (1963).

10. H.D. Block, B.W. Knight and F. Rosenblatt, "The Perceptron: A Model for Brain Functioning, II, Reviews of Modern Physics, vol. 34, #1, pp. 135-142 (Jan. 1962).

*These references were unavailable to me. My information on them is based on the comments of [3].

**These references were also unavailable. My information is based on the comments of [34].

11. R.E. Bonner, "On Some Clustering Techniques," IBM Journal of Research and Dev., Jan. 1964, pp. 22-32.

12. R. Cattell, "A Note on Correlation Clusters and Cluster Search Methods," Psychometrika, vol. 9, #3 (Sept. 1944).

13. D.B. Cooper and P.W. Cooper, "Adaptive Pattern Recognition and Signal Detection Without Supervision," IEEE International Convention Record, pt. 1, 1964, pp. 246-256.

14. __________ and __________, "Nonsupervised Adaptive Signal Detection and Pattern Recognition," Information and Control, vol. 7, #3 (Sept. 1964).

15. Paul W. Cooper, "Nonsupervised Learning in Statistical Pattern Recognition," Methodologies of Pattern Recognition, ed. Satosi Watanabe, 1969, pp. 97-109.

*16. R.F. Daly, "Adaptive Binary Detection," Stanford Elec. Lab. Tech. Report No. 2003-2, Stanford, Calif. (June 26, 1961).

*17. __________, "The Adaptive Binary-Detection Problem on the Real Line," Stanford Elec. Lab. Report SEL-62-030, Stanford, Calif. (Feb. 1962).

18. O. Firschein and M. Fischler, "Automatic Subclass Determination for Pattern Recognition Applications," Trans. PGEC, EC-12, #2 (Apr. 1963).

19. W.D. Fisher, "On Grouping For Maximum Homogeneity," Journal of American Stat. Assn., vol. 53, pp. 789-798 (Dec. 1958).

*20. J.J. Fortier and H. Solomon, "Clustering Procedures," Tech. Report 7, Dept. of Statistics, Stanford University (Mar. 20, 1964).

*21. S.C. Fralick, "The Synthesis of Machines Which Learn Without a Teacher," Tech. Report No. 6103-8, Stanford University (April 1964).

22. __________, "Learning to Recognize Patterns Without a Teacher," IEEE Trans. on Information Theory, vol. IT-13 (1967). pp. 57-64.

23. K.S. Fu, "Statistical Pattern Recognition," Adaptive Learning and Pattern Recognition Systems, ed. J.M. Mendel and K.S. Fu, (1970) pp. 68-75.

24. J.A. Gengerelli, "A Method for Detecting Subgroups in a Population and Specifying their Membership," Journal of Psychology, vol. 55 pp. 457-468 (1963).

25. E.M. Glaser, "Signal Detection by Adaptive Filters," IRE Trans. On Info. Theory, vol. IT-7, #2 (Apr. 1961).

**26. J.A. Hartigan, "Clustering," Lecture Notes in a 2-Day seminar, held at St. Louis, MO., Aug. 30-31, 1971, sponsored by the Institute for Advanced Technology, Washington, D.C.

**27. W.A. Holley, "Description and User's Guide for the IBM 360/44 ISODATA Program," Lockheed Electronics Co., Inc., HASD, Houston, Texas, Tech. Rep. 640-TR-030, Sept. 1971.

28. L. Hyvarinen, "Classificaiton of Qualitative Data," British Info. Theory J., 1962 pp. 83-89.

29. C.V. Jakowatz, R.L. Shuey and G.M. White, "Adaptive Waveform Recognition," Information Theory, ed. C. Cherry, Butterworths, Washington, D.C. 1961.

30. Bela Julesz, "Cluster Formation at Various Perceptual Levels," Methodologies of Pattern Recognition, ed. Satosi Watanabe (1969) pp. 297-307.

31. E.P.F. Kan, "Data Clustering: An Overview," Lockheed Electronics Company, Inc., HASD, Houston, Texas, Tech. Rep. 640-TR-080, Mar. 1972.

**32. _______, "ISODATA: Thresholds for Splitting Clusters," Lockheed Electronics Co., Inc., HASD, Houston, Texas, Tech. Rep. 640-TR-058, Jan. 1972.

**33. _______ and W.A. Holley, "Experience with ISODATA," Lockheed Electronics Co., Inc., HASD, Houston, Texas, Tech. Memo. TM 642-354, Mar. 1972.

34. _______ and ___________, "More on Clustering Techniques with Final Recommendations on ISODATA," Lockheed Electronics Co., Inc., HASD, Houston, Texas, Tech. Rep. #LEC 640-TR-112, May, 1972.

*35. G. Kaskey et. al, "Cluster Formation and Diagnostic Significance in Symptom Evaluation," Proc. Fall. Jt. Computer Conf., 1962, P. 285.

*36. M. Kochen, "Techniques for Information Retrieval Research: State of the Art," presented at IBM World Trade Corporation Information Retrieval Symposium at Blaricum, Holland, Nov. 1962.

37. R.L. Mattson and J.E. Damman, "A Technique for Determining and Coding Subclasses in Pattern Recognition Problems," IBM J. of Res. and Dev., vol. 9, 1965, pp. 294-302.

38. L.I. McQuitty, "Typal Analysis," Educational Psychological Meas., vol. 21, pp. 677-696 (1961).

39. C.D. Michener and R.R. Sokal, "A Quantitative Approach to a Problem in Classification," Evolution, vol. 11, pp. 130-162 (June 1957).

**40. G. Nagy and J. Tolaba, "Nonsupervised Crop Classification Through Airborne MSS Observations," IBM Journal of Res. and Dev., Mar. 1972.

*41. R.M. Needham, "The Theory of Clumps, II," Report M.L. 139, Cambridge Language Research Unit, Cambridge, England (Mar. 1961).

42. J. Nunnally, "The Analysis of Profile Data," Psychological Bulletin, vol. 59, #4, pp. 311-319, 1962.

43. M. Okajima, L. Stark, G. Whipple and S. Yasui, "Computer Pattern Recognition Techniques: Some Results with Real Electrocardiographic Data," IEEE Trans. on Bio-Medical Electronics, vol. BME-10, #3 (July 1963).

*44. A.F. Parker-Rhodes, "Contributions to the Theory of Clumps," IML. 138, Cambridge Language Research Unit, Cambridge, Eng. (Mar. 1961).

*45. E.A. Patrick and J.C. Hancock, "The Nonsupervised Learning of Probability Spaces and Recognition of Patterns," Tech. Report, Purdue Univ., Lafayette, Ind. (1965).

46. D.J. Rogers and T.T. Tanimoto, "A Computer Program for Classifying Plants," Science, vol. 132, Oct. 21, 1960.

47. W.L. Sawrey, L. Keller and J.J. Conger, "An Objective Method of Grouping Profiles by Distance Functions and its Relation to Factor Analysis," Educational and Psychological Measurement, vol. 20 #4 (1960).

**48. J.A. Schell, "A Comparison of Two Approaches for Category Identification and Classification Analysis from an Agricultural Scene," presented at Conference on Earth Resources Observation and Information Analysis Systems, Tullahoma, Tenn., Mar. 1972.

49. G.S. Sebestyen, "Pattern Recognition by an Adaptive Process of Sample Set Construction," IRE Trans. on Info. Theory, vol IT-8, Sept. 1962.

*50. _______ and J. Edie, "Pattern Recognition Research," Air Force Cambridge Res. Lab. Report 64-821 (AD 608 692), Bedford, Mass (June 14, 1964).

51. J.W. Smith, "The Analysis of Multiple Signal Data," IEEE Trans. On Information Theory, vol. IT-10, #3 (July 1964).

*52. R.R. Sokal and C.D. Michener, "A Statistical Method For Evaluating Systematic Relationships," University of Kansas Science Bulletin, Mar. 20, 1958.

53. _______ and P.H.A. Sneath, Principles of Numerical Taxonomy, W.H. Freeman and Co. San Francisco, 1963.

*54. J.J. Spilker, Jr., D.D. Luby and R.D. Lawhorn, "Progress Report--Adaptive Binary Waveform Detection," Tech. Report 75, Communication Sciences Department, Philco Corp., Palo Alto, Calif. (Dec. 1963).

55. K. Steinbuch and U.A.W. Piske, "Learning Matrices and Their Applications," IEEE Trans. on Electronic Computers, vol. EC-12, #6 (Dec. 1963).

56. R.C. Tryon, "Cluster Analysis," Psychometrika, vol. 22, #3, pp. 241-260 (Sept. 1957).

57. ______, "Domain Sampling Formulation of Cluster and Factor Analysis," Psychometrika, vol. 24, no. 2, pp. 113-135 (June 1959)

58. ______ and Daniel E. Bailey, "Cluster Analysis," McGraw Hill, New York (1970).

**59. B.J. Turner, "Cluster Analysis of MSS Remote Sensors Data," presented by Conference on Earth Resources Observation and Information Analysis

Systems, Tullahoma, Tenn., Mar. 1972.

60. J.H. Ward, Jr., "Hierarchical Grouping to Optimize an Objective Function," Journal of American Statistical Association, vol. 58, 301 (Mar. 1963).

61. ________ and Marion E. Hook, "Application of an Heirarchical Grouping Procedure to a Problem of Grouping Profiles," Educational and Psychological Measurement, vol. 23, no. 1 (1963).

62. G. Young, "Factor Analysis and the Index of Clustering," Psychometrika, vol. 4, no. 3 (Sept. 1939).

J-1

AN EVALUATION OF AN ALGORITHM
FOR LINEAR INEQUALITIES AND ITS APPLICATIONS

Report # 11

Contract NAS-9-12777

by

John Jurgensen

Department of Mathematics

University of Houston

September 1972

J-a

AN EVALUATION OF AN ALGORITHM FOR LINEAR INEQUALITIES AND ITS APPLICATIONS

by

John Jurgensen
Mathematics Department
University of Houston

ABSTRACT

The following presents an algorithm for obtaining a solution α to a set of inequalities $A\alpha > 0$ where A is an $N \times m$ matrix and α is an m-vector. If the set of inequalities is consistant, then the algorithm is guaranteed to arrive at a solution in a finite number of steps. Also, if in the iteration, a negative vector is obtained, then the initial set of inequalities is inconsistant, and the iteration is terminated.

Several mathematical errors were encountered. These have been corrected, and distinct correct proofs have replaced the original proofs whenever possible. When the damage was irreparable, the material was deleted after appropriate comments.

J-b

AN EVALUATION OF AN ALGORITHM FOR LINEAR INEQUALITIES AND ITS APPLICATIONS

Let A be a given $N \times m$ matrix, with $N > m$. Find $\beta > 0$ and α such that $J = \|A\alpha - \beta\|^2$ is minimized. The gradient of J with respect to α is

$$\frac{\partial J}{\partial \alpha} = A^T(A\alpha - \beta) \ .$$

Thus

$$\begin{aligned}\frac{\partial J}{\partial \alpha} = 0 \Rightarrow \alpha &= (A^T A)^{\#} A^T \beta \\ &= A^{\#} A^{T\#} A^T \beta \\ &= A^{\#} \beta\end{aligned}$$

where $A^{\#}$ is the generalized inverse of A.

From $\beta > 0$ and the descent procedure

$$\beta(i+1) = \beta(i) + \delta\beta(i)$$

where

$$\delta\beta_j(i) \text{ is proportional to } \begin{cases} (A\alpha(i) - \beta(i))_j & \text{if } (A\alpha(i) - \beta(i))_j > 0 \\ 0 & \text{if } (A\alpha(i) - \beta(i))_j \le 0 \end{cases}$$

that is, $\delta\beta(i) = \rho[A\alpha(i) - \beta(i) + |A\alpha(i) - \beta(i)|]$ where $\rho > 0$ is a positive constant scalar, to be determined later, we obtain the following algorithm

$$\begin{aligned}&\alpha(0) = A^{\#} \beta(0) \ , \quad \beta(0) > 0 \ , \quad \text{arbitrary} \\ &\text{define} \quad y(i) = A\alpha(i) - \beta(i) \\ (5) \qquad &\beta(i+1) = \beta(i) + \rho[y(i) + |y(i)|] \\ &\alpha(i+1) = A^{\#}\beta(i+1) \\ &\qquad = A^{\#} \beta(i) + \rho A^{\#}[y(i) + |y(i)|] \\ &\qquad = \alpha(i) + \rho A^{\#}[y(i) + |y(i)|] \ .\end{aligned}$$

The algorithm (5) can be rewritten as:

$$\begin{aligned} y(i+1) &= A\,\alpha(i+1) - \beta(i+1) \\ &= A[\alpha(i) + \rho\, A^{\#}(y(i) + |y(i)|)] \\ &\qquad - \beta(i) - \rho(y(i) + |y(i)|) \\ &= [A\,\alpha(i) - \beta(i)] + \rho(AA^{\#} - I)[y(i) + |y(i)|] \\ &= y(i) + \rho(AA^{\#} - I)[y(i) + |y(i)|]. \end{aligned}$$

<u>Lemma:</u> Consider the inequalities (6), and the algorithm (5) to solve them.

Then

(1) $y(i) \not\geq 0$ for any i (<u>clearly false</u>)

(2) If (6) is consistent, then $y(i) \not\geq 0$ for any i.

<u>Proof:</u> (1) is clearly false; consider the case where

$$A = \begin{pmatrix} I \\ Z \end{pmatrix},\ \alpha(0) = (2,2,\text{---},2)^T, \text{ and } \beta(0) = (1,1,\text{---},1)^T.$$

Then
$$\begin{aligned} y(0) &= A\,\alpha(0) - \beta(0) \\ &= (2,2,\text{---},2)^T - (1,1,\text{---},1)^T \\ &= (1,1,\text{---},1)^T \\ &\geq 0 \end{aligned}$$

The "proof" is based on the erroneous "fact" that $(AA^{\#} - I) \leq 0$. The example on page 9 together with the vectors (1,0,-1) and (-2,0,-1) show that $(AA^{\#} - I)$ need be neither positive semi-definite nor negative semi-definite.

In case $y(i) = A\,\alpha(i) - \beta(i) \geq 0$, for the value $\frac{1}{2}$ for ρ, as suggested on page 8, we arrive at a solution in the next iteration:

$$\beta(i+1) = \beta(i) + \frac{1}{2}\,(A\,\alpha(i) - \beta(i) + |A\,\alpha(i) - \beta(i)|) = A\,\alpha(i)$$

$$\alpha(i+1) = A^{\#}A\,\alpha(i)$$

$$A\,\alpha(i+1) - \beta(i+1) = AA^{\#}A\,\alpha(i) - A\,\alpha(i) = 0.$$

Lemma (cont)

Proof: (2) Assume $\exists\, i \in : y(i) \leq 0$. Since (6) is consistent, $\exists \alpha^*, \beta^* > 0 \;:\; A\alpha^* = \beta^* > 0$.

(7) Then $y^T(i)\ \beta^* < 0$.

But

$$\begin{aligned} A^T y(i) &= A^T[A(\alpha(i)) - \beta(i)] \\ &= A^T[AA^{\#}\beta(i) - \beta(i)] \\ &= A^T(AA^{\#} - I)\ \beta(i) \\ &= (A^T AA^{\#} - A^T)\beta(i) \\ &= (A^T - A^T)\ \beta(i) \\ &= 0. \end{aligned}$$

Also,

$$\begin{aligned} A^T(y(i)) = 0 &\Rightarrow (\alpha^*)^T A^T y(i) = 0 \\ &\Rightarrow [(\alpha^*)^T A^T y(i)]^T = 0 \\ &\Rightarrow y(i)^T A\alpha^* = 0 \\ &\Rightarrow y(i)^T \beta^* = 0. \end{aligned}$$

But this contradicts (7).

Therefore, if (6) is consistent, then $y(i) \not\leq 0$ for any i.

Proposition Consider the set of inequalities

(6) $A\,\alpha \geq 0$ and the algorithm (5) to solve them. Let $V(y(i)) = \|y(i)\|^2$.

$\textcircled{1}$ $\textcircled{a}$ If (6) is consistent then

$$\lim_{i\to\infty} V(y(i)) = 0,$$ implying convergence to a solution.

Note: this proof corrects the error that $\Delta V(y(i))\text{"="}-\|y(i) + |y(i)|\|^2\{\rho^2 AA^{\#} + (\rho - \rho^2)I\}$

Proof:

$$\begin{aligned} \Delta V(y(i)) &= V(y(i+1)) - V(y(i)) \\ &= \|y(i+1)\|^2 - \|y(i)\|^2 \\ &= \|y(i) + \rho(AA^{\#} - I)[y(i) + |y(i)|]\|^2 - \|y(i)\|^2 \\ &= [y(i) + \rho(AA^{\#} - I)[y(i) + |y(i)|]]^T[y(i) + \rho(AA^{\#} - I)[y(i) + |y(i)|]] \\ &\qquad - [y(i)]^T[y(i)] \\ &= \{\rho(AA^{\#} - I)\{y(i) + |y(i)|]\}^T\ y(i) + \end{aligned}$$

$$[y(i)]^T\{\rho(AA^{\#}- I)[y(i) + |y(i)|]\}$$

$$+ \{\rho(AA^{\#}- I)[y(i) + |y(i)|]\}^T\{\rho(AA^{\#}- I)[y(i) + |y(i)|]\}$$

$$= \rho[y(i) + |y(i)|]^T(AA^{\#}- I)^T y(i)$$

$$+ \rho y(i)^T(AA^{\#}- I)[y(i) + |y(i)|]$$

$$+ \rho^2[y(i) + |y(i)|]^T(AA^{\#}- I)^T(AA^{\#}- I)[y(i) + |y(i)|]$$

$$= \rho[y(i) + |y(i)|]^T(AA^{\#}- I)y(i)$$

$$+ \rho y(i)^T(AA^{\#}- I)[y(i) + |y(i)|]$$

$$+ \rho^2[y(i) + |y(i)|]^T(AA^{\#}- I)[y(i) + |y(i)|],$$

since $(AA^{\#}- I)$ is symmetric and idempotent.

<u>Note:</u> $AA^{\#}y(i) = AA^{\#}(AA^{\#}- I)\beta(i)$

$$= AA^{\#}AA^{\#}\beta(i) - AA^{\#}\beta(i)$$

$$= AA^{\#}\beta(i) - AA^{\#}\beta(i)$$

$$= 0$$

and $y(i)^T AA^{\#} = (AA^{\#}y(i))^T = 0^T = 0.$

$$\rho[y(i) + |y(i)|]^T(AA^{\#}- I)y(i)$$

$$= \rho y(i)^T(AA^{\#}- I)y(i)$$

$$+ \rho|y(i)|^T(AA^{\#}- I)y(i)$$

$$= \rho y(i)^T AA^{\#}y(i) - \rho y(i)^T I\ y(i)$$

$$+ \rho|y(i)|^T AA^{\#}y(i) - \rho|y(i)|^T I\ y(i)$$

$$= -\rho\ \|y(i)\|^2 - \rho|y(i)|^T y(i)$$

Also $\rho\ y(i)^T(AA^{\#}- I)[y(i) + |y(i)|]$

$$= \rho\ y(i)^T AA^{\#}[y(i) + |y(i)|] - \rho\ y(i)^T I[y(i) + |y(i)|]$$

$$= -\rho\ y(i)^T[y(i) + |y(i)|]$$

$$= -\rho\ \|y(i)\|^2 - \rho\ y(i)^T|y(i)|$$

Also, $\| y(i) + |y(i)| \|^2 = [y(i) + |y(i)|]^T [y(i) + |y(i)|]$

$$= y(i)^T y(i) + |y(i)|^T y(i) + |y(i)|^T |y(i)| + y(i)^T |y(i)|$$

$$= \| y(i) \|^2 + |y(i)|^T y(i) + \| y(i) \|^2 + y(i)^T |y(i)|$$

Hence, $\Delta V(y(i)) = -\rho\Big(\|y(i)\|^2 + |y(i)|^T y(i) \Big) - \rho\Big(\|y(i)\|^2 + y(i)^T |y(i)| \Big)$

$$+ \rho^2 [y(i) + |y(i)|^T (AA^{\#} - I)[y(i) + |y(i)|]$$

$$= -\rho \|y(i) + |y(i)|\|^2 + \rho^2 [y(i)+|y(i)|]^T (AA^{\#}-I)[y(i)+|y(i)|]$$

$$= -\rho \|y(i) + |y(i) + |y(i)| \|^2$$
$$+\rho^2 [y(i) + |y(i)|]^T AA^{\#} [y(i) + |y(i)|] - \rho^2 \|y(i) + |y(i)| \|^2$$

Also, $[y(i) + |y(i)|]^T AA^{\#} [y(i) + |y(i)|]$

$$= y(i)^T AA^{\#} [y(i) + |y(i)|]$$

$$+ |y(i)|^T AA^{\#} y(i) + |y(i)|^T AA^{\#} |y(i)|$$

$$= |y(i)|^T AA^{\#} |y(i)|$$

$= 0$, since $AA^{\#} y(i) = 0$.

Therefore, $\Delta V(y(i)) = -(\rho + \rho^2) \|y(i) + |y(i)| \|^2$

Thus, for $\rho > 0$, $\Delta V(y(i)) \le 0$, for all i

$\Delta V(y(i)) = 0$ iff $y(i) = 0$ or $y(i) \le 0$.

By the lemma, $y(i) \nleq 0$.

Therefore, $\Delta V(y(i)) \left\{ \begin{array}{ll} < 0 & \forall y(i) \neq 0 \\ = 0 & \text{if } y(i) = 0. \end{array} \right.$

By Lyapunov's stability theorem for discrete systems, $y(i+1) = y(i) + \rho(AA^{\#}-I)$ $(y(i) + |y(i)|)$ is globally asymptatically stable.

Therefore, $\lim_{i \to \infty} \|y(i)\| = 0$.

Proposition ①ⓑ: If $A\alpha > 0$ is consistent, then

$$\Delta V(y(i)) = V(y(i+1)) - V(y(i)) < -\lambda_o V(y(i)) \quad \text{with} \quad \lambda_o > 0$$

showing exponential convergence.

The "proof" given was based on the erroneous fact that $\Delta V(y(i))$ "=" $-\| y(i) + |y(i)| \|^{2}\{\rho^2 AA^{\#} + (\rho-\rho^2)I\}$.

Hence the non-zero eigenvalues of the matrix

$$C(i)\ \{\rho^2 AA^{\#} + (\rho-\rho^2)I\}\ C(i)$$

where $C(i)$ is the diagonal matrix defined by

$$C_{jj}(i) = \begin{array}{ll} 2 & \text{if } y_j(i) \geq 0 \\ 0 & \text{if } y_j(i) < 0 \end{array}$$

are irrelavant to this discussion.

The "proof" cannot be corrected by using the correct value of $\Delta V(y(i))$, $-(\rho+\rho^2)\|y(i) + |y(i)|\|$,

$$\begin{aligned} -\Delta V(y(i)) &= (\rho+\rho^2)V(C(i)y(i)) \\ &\leq (\rho+\rho^2)V(2y(i)) \\ &\leq 4(\rho+\rho^2)V(y(i)) \end{aligned}$$

and hence $\Delta V(y(i)) \geq -4(\rho+\rho^2)V(y(i))$.

Fortunately, ① ⓒ is not only a stronger statement than ① ⓑ, it is also proven independently and correctly.

Proposition ①ⓒ: Consider the set of inequalities (6) $A\alpha \geq 0$ and the algorithm (5) to solve them. If (6) is consistent, then a solution is obtained in a finite number of steps.

Proof: Recalling that $\beta(i+1) = \beta(i) + \rho[y(i) + |y(i)|]$, $\rho > 0$ we observe that β is a non-decreasing vector. That is, each coordinate of β is non-decreasing.

Thus, choosing $\beta(0)^T = (1,1,\text{---},1)$, every coordinate of $\beta(i) \geq 1$ for all i.

Since $V(y(i)) \longrightarrow 0, \exists\, N \ni: i > N \Rightarrow V(y(i)) < 1$.

But $V(y(i)) < 1 \Rightarrow$ each coordinate of $|y(i)| < 1$.

Therefore, $A\,\alpha(i) = \beta(i) + y(i) > 0, \; \forall\; i > N$.

Therefore, a solution to $A\,\alpha \geq 0$ is obtained in a finite number of steps.

<u>Proposition 2:</u> If (6) is inconsistent, then there exists a positive integer i^* such that

$$\begin{array}{ll} \Delta V(y(i)) < -\lambda_o V(y(i)) & \text{if } i < i^* \\ \Delta V(y(i)) = 0 & \text{if } i \geq i^* \\ y(i) \nleq 0 & \text{if } i < i^* \\ y(i) = y(i^*) \leq 0 & \text{if } i \geq i^* \\ \alpha(i) = \alpha(i^*) & \text{if } i \geq i^* \\ \beta(i) = \beta(i^*) & \text{if } i \geq i^* \end{array}$$

Unfortunately, his entire proof is based on the misconception that (after showing that $\Delta V(y(i)) \leq 0$), "since $y(i)$ and hence $V(y(i))$ cannot become zero for any i [since (6) is assumed to be inconsistent], there must exist a value of i, say i^*, such that $\Delta V(y(i)) < 0$ for $i < i^*$

$\Delta V(y(i)) = 0$ for $i = i^*$"

However, it does follow from part ② of the lemma, that the verbal explanation of proposition 2 is correct: "In other words, the occurance of a nonpositive vector $y(i)$ at any stage terminates the algorithm and indicates the inconsistency of (6)." This is possible because the verbal explanation is not equivalent to the statement of the proposition.

Further implications of the existance of i such that

$$y(i^*) \le 0 \text{ follow:}$$

Then $y(i^*) + |y(i^*)| = 0$.

$$y(i^* + 1) = y(i^*) + \rho(AA^{\#} - I)(y(i^*) + |y(i^*)|) = y(i^*).$$

Similiarly $\beta(i^* + 1) = \beta(i^*) + \rho[y(i^*) + |y(i^*)|] = \beta(i^*)$ and

$$\alpha(i^* + 1) = \alpha(i^*) + \rho A^{\#}[y(i^*) + |y(i^*)|] = \alpha(i^*)$$

Hence $y(i) = y(i^*)$ for $i \ge i^*$

$\beta(i) = \beta(i^*)$ for $i \ge i^*$

$\alpha(i) = \alpha(i^*)$ for $i \ge i^*$

also $\Delta V(y(i)) = 0$ for $i > i^*$

This proceedure is compared with other algorithms, but unfortunately, this algorithm is "rewritten as":

$$\begin{aligned} \alpha(i + 1) &= \alpha(i) + \rho(A^TA)^{-1}\{|A\alpha(i) - \beta(i)| - (A\alpha(i)\beta(i))\} \\ &= \alpha(i) + \rho(A^TA)^{-1}\{|y(i)| - y(i)\} \\ \beta(i + 1) &= \beta(i) + \rho\{|A\,\alpha(i) - \beta(i)| + (A\,\alpha(i) - \beta(i))\} \\ &= \beta(i) + \rho\{|y(i)| + y(i)\} \end{aligned}$$

This is a change from

$$\alpha(i + 1) = \alpha(i) + \rho\ A^{\#}\{|y(i)| + y(i)\}.$$

Clearly these two expressions are not in general equivalent, even if the sign for $y(i)$ in the new expression is made consistent with all of the other expressions of this type.

When implementing this algorithm, one standard initialization is to let $\beta(0)$ be the vector composed of all $+1$'s and to let $\rho = 1/2$. The latter compensates for the multiple of two arising from $[y(i) + |y(i)|]$.

The algorithm was applied to the matrix used to illustrate the numerical computation of a generalized inverse in the MSC Internal Techinal Note MSC - IN - 64 - ED6, The Concept of Generalized Inversion of Arbitrary Complex Matrices by Henry P. Decell, Jr. For ease of calcualtion, only two place accuracy was used.

$$A = \begin{pmatrix} 4 & -1 & -3 & 2 \\ -2 & 5 & -1 & -3 \\ 2 & 3 & -9 & -5 \end{pmatrix}_{3x4}$$

$$(AA^{\#} - I) = \begin{pmatrix} 0 & .6 \text{ D-14} & .27 \text{ D-13} \\ .58 \text{ D-14} & 0 & 69 \text{ D-14} \\ -.13 \text{ D-13} & .92 \text{ D-14} & 0 \end{pmatrix}_{3x3}$$

$$A^{\#} = \begin{pmatrix} .19 & .60 \text{ D-1} & -.36 \text{ D-1} \\ .\ 2 & .\ 3 & -.\ 1 \\ -.56 \text{ D-2} & .53 \text{ D-1} & -.90 \text{ D-1} \\ .21 & .11 & -.11 \end{pmatrix}_{4x3}$$

Let $\beta(0) = (1, 1, 1)^T$. Let $\rho = 1/2$.

$$\alpha(0) = A^{\#}\beta(0) = \begin{pmatrix} .19 & & +.06 & & -.036 & \\ .\ 2 & & .\ 3 & & -.\ \ 1 & \\ -.056 & D\text{-}1 & .53 & D\text{-}1 & -.90 & D\text{-}1 \\ .21 & & .11 & & -.11 & \end{pmatrix}_{4\times 1} = \begin{pmatrix} .21 & \\ .\ 4 & \\ .43 & D\text{-}1 \\ .21 & \end{pmatrix}_{4\times 1}$$

$$A\,\alpha(0) = \begin{pmatrix} 4 & -1 & -3 & 2 \\ -2 & 5 & -1 & -3 \\ 2 & 3 & -9 & -5 \end{pmatrix}_{3\times 4} \begin{pmatrix} .21 \\ .\ 4 \\ -.043 \\ .21 \end{pmatrix}_{4\times 1}$$

$$= \begin{pmatrix} .84 & -\ .4 & +.129 & +\ .42 \\ -.42 & +2.0 & +.043 & -\ .63 \\ .42 & +1.2 & +.387 & -1.05 \end{pmatrix}_{3\times 1}$$

$$= \begin{pmatrix} .99 \\ .99 \\ .94 \end{pmatrix}_{3\times 1}$$

$A\,\alpha(0) \geq 0$. The algorithm arrives at a solution on the zeroth iteration!

This algorithm has the distinct advantage of being finite, but has no bound on the number of iterations. There is a flag signaling the inconsistancy of the set of equations, but unfortunately there is no guarantee that the flag will occur if the equations are inconsistent.

There are only two matrices that have to be calculated, namely $A^{\#}$ and $(AA^{\#}-I)$, and these only have to be calculated once. This has the strong advantage of minimizing both computation time and storage requirements. This method has one other disadvantage - the primary iteration does not yield the desired vector, so two iterations must be continued concurrently (unless it is preferable to store several to many vectors and perform the second iteration after is it known that the desired vector exists). This disadvantage is mimimized by the similiarity in the algorithms: $y(i+1) = y(i) + \rho(AA^{\#}- I)[y(i) + |y(i)|]$ and $\alpha(i+1) = \alpha(i) + \rho A^{\#}[y(i) + |y(i)|]$. That is, only the vector $[y(i) + |y(i)|]$ need be computed, and then used in both of the algorithms.

REFERENCES

(1) An algorithm for Linear Inequalities and its applications Y-C Ho and R. L. Kashyap IEEE Transactions on Electronic Computers Oct. 1965.

(2) The Concept of Generalized Inversion of Arbitrary Complex Matrices Henry P. Decell, Jr. MSC - IN - 64 - ED6, July 16, 1964.

(3) The Theory of Generalized Inverses by Pat O'Dell and Boullion, John Wiley and Sons.

(4) A Generalized Inverse for Matrices, R. Penrose Proceedings of the Cambridge Philosophical Society, 1955.

(5) The Relaxation Method for Linear Inequalities, Shmuel Agmon.

(6) The Efficiency of a Linear Discriminant Function for Arbitrary Distributions Yu. N. Zhezhel Engineering Cybernatics No. 6, 1968

(7) Generalized Inverse Approach to adaptive Multi-class Pattern Classification W. G. Wee IEEE Transactions on Computers, Dec. 1968.

(8) An Optimal Discriminant Plane, J. W. Sammon, Jr. IEEE Transactions on Computers, Sept. 1970.

(9) The Use of an Adaptive Threshold Element to Design a Linear Optimal Pattern Classifier J. S. Koford and G. F. Groner IEEE Transactions on Information Theory, Jan, 1966

K- 1-7

DIVERGENCE

Some Necessary Conditions For An Extremum

J. A. Quirein
University of Houston
Mathematics Department

November, 1972

NAS-9-12777

#12

K-1-a

Introduction - One of the important problems in pattern recognition is that of feature extraction or selection. Tou and Heydorn (1967) proposed a procedure for two pattern classes to find a dimension reducing transformation matrix B that maximizes the divergence in the reduced dimension. C.C. Babu (1972) extended the above procedure to the multi-class problem by maximizing the average divergence in the reduced dimension. Both of the above papers present necessary conditions for the divergence in the reduced dimensional space to be an extremum. Neither of the papers present an explicit solution for obtaining B, and both suggest that B be obtained numerically. Baba's expression for the gradient of the average divergence with respect to B is rather lengthy and numerically unattractive, since it is expressed in terms of many eigenvalues and vectors, which of course must be obtained. Tou's expression, in addition to being numerically unattractive, is valid only in the case of two distinct classes.

In this paper, a comparitively simple expression for the gradient of the average divergence with respect to B is developed. The developed expression for the gradient contains no eigenvectors or eigenvalues; also, all matrix inversions necessary to evaluate the gradient are available from computing the average divergence.

SECTION 1 - THREE FUNDAMENTAL LEMMAS.

Let

B ; k by n matrix of rank $k \leq n$

Λ ; n by n symmetric matrix of rank n

S ; n by n symmetric matrix

and define

$$\psi = \frac{1}{2}\,\mathrm{tr}\{(B\;\Lambda\;B^T)^{-1}(BSB)^T)\}$$

We prove the following Lemma

Lemma 1

$$\left(\frac{\partial\psi}{\partial B}\right)^T = [SB^T - \Lambda B^T(B\;\Lambda\;B^T)^{-1}(B\;S\;B^T)](B\;\Lambda\;B^T)^{-1}$$

Proof: Taking the differential of ψ, it is easily verified

$$d\psi = F + G, \quad \text{where}$$

$$\begin{aligned} F &= \frac{1}{2}\,\mathrm{tr}\{(B\;\Lambda\;B^T)^{-1}(dB\;S\;B^T + B\;S\;dB^T)\} \\ &= \frac{1}{2}\,\mathrm{tr}\{(dB\;S\;B^T)(B\;\Lambda\;B^T)^{-1}\} + \frac{1}{2}\,\mathrm{tr}\{B\;\Lambda\;B^T)^{-1}(B\;S\;dB^T)\} \\ &= \frac{1}{2}\,\mathrm{tr}\{(dB\;S\;B^T)(B\;\Lambda\;B^T)^{-1}\} + \frac{1}{2}\,\mathrm{tr}\{[(dB\;S\;B^T)(B\;\Lambda\;B^T)^{-1}]^T\} \\ &= \mathrm{tr}\{(dB\;S\;B^T)(B\;\Lambda\;B^T)^{-1}\} \end{aligned}$$

and

$$G = -\frac{1}{2}\,\mathrm{tr}\{(B \wedge B^T)^{-1}(dB \wedge B^T + B \wedge dB^T)(B \wedge B^T)^{-1}(B\ S\ B^T)\}$$

$$= -\frac{1}{2}\,\mathrm{tr}\{(dB \wedge B^T)(B \wedge B^T)^{-1}(B\ S\ B^T)(B \wedge B^T)^{-1}\}$$

$$-\frac{1}{2}\,\mathrm{tr}\{(B \wedge B^T)^{-1}(B\ S\ B^T)(B \wedge B^T)^{-1}(B \wedge dB^T)\}$$

$$= -\mathrm{tr}\{(dB \wedge B^T)(B \wedge B^T)^{-1}(B\ S\ B^T)(B \wedge B^T)^{-1}\}$$

thus

$$d\psi = F + G$$

$$= \mathrm{tr}\{dB[SB^T - \Lambda B^T(B \wedge B^T)^{-1}(B\ S\ B^T)](B \wedge B^T)^{-1}\}$$

Now, define

$$H = [SB^T - \Lambda B^T(B\Lambda B^T)^{-1}(BSB^T)](B\Lambda B^T)^{-1}$$

so that

$$d\psi = \mathrm{tr}\{dBH\}$$

and

$$\frac{\partial\psi}{\partial b_{ij}} = \mathrm{tr}\{\frac{\partial B}{\partial b_{ij}}\ H\} = h_{ji}$$

where h_{ji} is the element in the j^{ith} row and i^{ith} column of H. Since $\frac{\partial\psi}{\partial b_{ij}}$ is the element in the i^{ith} row and j^{ith} column of $\partial\psi/\partial B$, it follows that

$$\left(\frac{\partial\psi}{\partial B}\right)^T = H$$

Q.E.D.

<u>Lemma 2</u>

$$B\left(\frac{\partial\psi}{\partial B}\right)^T = 0$$

Proof: Immediate from Lemma 1.

<u>Remark 1</u> - Note that when $k = n$, so that B is non-singular, Lemma 2 shows that ψ is in variant under a non-singular transformation, ie

$$\frac{\partial\psi}{\partial B} = 0$$

<u>Remark 2</u> - If $n \geq 3$ and $k = n-1$, then the column vectors of $(\partial\psi/\partial B)^T$ are linearly dependent, since by Lemma two, the rank of $(\partial\psi/\partial B)^T$ is at most 1.

<u>Lemma 3</u> : Let Q be a non-singular k by k matrix. Let $\hat{B} = QB$. Then

$$\left(\frac{\partial\psi}{\partial B}\right)^T = 0 \text{ implies} \left(\frac{\partial\psi}{\partial \hat{B}}\right)^T = 0$$

Proof: By Lemma 1

$$\begin{aligned}\left(\frac{\partial\psi}{\partial \hat{B}}\right)^T &= [S\hat{B}^T - \Lambda\hat{B}^T(\hat{B}\ \Lambda\ \hat{B}^T)^{-1}(\hat{B}\ S\ \hat{B}^T)](\hat{B}\ \Lambda\ \hat{B}^T)^{-1} \\ &= [SB^TQ^T - \Lambda B^TQ^T(Q^T)^{-1}(B\ \Lambda\ B^T)^{-1}Q^{-1}Q(BSB^T)Q^T](Q^T)^{-1}(B\ \Lambda\ B^T)Q^{-1} \\ &= \left(\frac{\partial\psi}{\partial B}\right)^TQ^{-1} \\ &= (0)\end{aligned}$$

Q.E.D.

SECTION 2

B-AVERAGE INTERCLASS DIVERGENCE - A NECESSARY CONDITION FOR AN EXTREMUM.

Assume the existence of m distinct classes with means and covariances

μ_i n-dimensional mean vector for class i.

Λ_i n by n covariance for class i, assumed to be positive definite.

Let $\delta_{ij} = \mu_i - \mu_j$ so that $\delta_{ij}\, \delta_{ij}^T = \delta_{ji}\, \delta_{ji}^T$

The interclass divergence between classes i and j is defined in Reference 1 as

$$D(i,j) = \frac{1}{2}\, \mathrm{tr}\{\Lambda_i^{-1}(\Lambda_j + \delta_{ij}\, \delta_{ij}^T)\} + \frac{1}{2}\, \mathrm{tr}\{\Lambda_j^{-1}(\Lambda_i + \delta_{ij}\, \delta_{ij}^T)\} - n$$

Note that when $\Lambda_i = \Lambda_j$ and $\mu_i = \mu_j$,

$$D(i,j) = 0$$

so that $D(i,j)$ is in a sense, a measure of the degree of difficulty of distinguishing between classes i and j, with the larger the value of $D(i,j)$, the less the degree of difficulty of distinguishing between classes i and j.

There is a discussion in Reference 2 of a natural generalization of the interclass divergence i.e., the average interclass divergence, defined by

$$D = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} D(i,j)$$

$$= \frac{1}{2}\, \mathrm{tr}\{\sum_{i=1}^{m} \Lambda_i^{-1}(\sum_{\substack{j=1 \\ j \neq i}}^{m} [\Lambda_j + \delta_{ij}\ \delta_{ij}^{T}])\} - \frac{m(m-1)}{2}\, n$$

$$= \frac{1}{2}\, \mathrm{tr}\{\sum_{i=1}^{m} \Lambda_i^{-1} S_i\} - \frac{m(m-1)}{2}\, n$$

where

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^{m} [\Lambda_j + \delta_{ij}\ \delta_{ij}^{T}]$$

We are interested in performing the transformation

$$y = Bx$$

where

x ; an n-dimensional observation vector

B ; a k by n matrix of rank k, with $k \leq n$

y ; the k-dimensional transformed observation vector

It is shown in Reference 3 that corresponding to the transformation $y = Bx$, the means transforms,

$$\mu_i \longrightarrow B\mu_i$$

and the covariances transforms,

$$\Lambda_i \longrightarrow B\Lambda_i B^{T}$$

Thus subsequent to performing the transformation $y = Bx$, we can assume the existence of m classes with means and covariances—

$B\mu_i$; k-dimensional mean vector for class i

$B\Lambda_i B^T$; k by k covariance for class i, which is positive definite by the assumptions on B and Λ_i.

Thus in k-dimensional space, the B-induced interclass divergence $D_B(i,j)$, is, by definition of the interclass divergence;

$$D_B(i,j) = \frac{1}{2}\,\mathrm{tr}\{(B\Lambda_i B^T)^{-1} B(\Lambda_j + \delta_{ij}\,\delta_{ij}^T)B^T\}$$

$$+ \frac{1}{2}\,\mathrm{tr}\{(B\Lambda_j B^T)^{-1} B(\Lambda_i + \delta_{ij}\,\delta_{ij}^T)B^T\} - k$$

Similarly, in k-dimensional space, we can define the B-average interclass divergence, D_B, as

$$D_B = \sum_{i=1}^{m-1}\sum_{j=i+1}^{m} D_B(i,j)$$

$$= \frac{1}{2}\,\mathrm{tr}\{\sum_{i=1}^{m} [(B\Lambda_i\, B^T)^{-1}(BS_i B^T)]\} - \frac{m(m-1)}{2}\,k$$

where, as defined previously

$$S_i = \sum_{\substack{j=1\\ j\neq i}}^{m} [\Lambda_j + \delta_{ij}\,\delta_{ij}^T]$$

Note that in performing the transformation $y = Bx$, the dimension of each observation is reduced from n to k, so that in a sense, information is lost. It is shown in Reference 2 that a measure of the information lost is given by the difference

$$D - D_B \geq 0$$

We are interested in minimizing the information lost, as measured by the average interclass divergence. Thus, it is desired to maximize the B-average interclass divergence, or equivalently, minimize $- D_B$. We prove the following theorem;

<u>THEOREM 1</u> - Let a k by n matrix B of rank k extremize D_B. Then it is necessary that B satisfy an equation of the form

$$\left(\frac{\partial D_B}{dB}\right)^T = \sum_{i=1}^{m} [S_i B^T - \Lambda_i B^T (B\Lambda_i B^T)^{-1} (BS_i B^T)] (B\Lambda_i B^T)^{-1} = 0$$

Also, if $\hat{B} = QB$, where Q is a non-singular k by k matrix,

$$\left(\frac{\partial D_{\hat{B}}}{\partial \hat{B}}\right)^T = \left(\frac{\partial D_B}{dB}\right)^T Q^{-1}$$

So that B is unique up to a non-singular k by k linear transformation.

Proof: Immediate from the definitions of B and D_B, and Lemmas 1 and 3.

Q.E.D.

<u>Remark 1</u> - The expression $\frac{\partial D_B}{dB}$ is the gradient of the B-average interclass divergence with respect to B. Note that the expressions for D_B and $\partial D_B/\partial B$ are rather easily evaluated.

<u>THEOREM 2</u> - Let B be a k by n matrix of rank k such that $BB^T = I$, and satisfying

$$(B^TB)S_i = S_i(B^TB) \quad \text{and} \quad (B^TB)\Lambda_i = \Lambda_i(B^TB)$$

$$i = 1,2,\ldots,m$$

then $\left(\frac{\partial D_B}{dB}\right)^T = 0$

Proof: By the above commutivity and since $BB^T = I$, it is readily verified

$$(BS_iB^T)^{-1} = BS_i^{-1}B^T \quad \text{and} \quad (B\Lambda_iB^T)^{-1} = B\Lambda_i^{-1}B^T$$

Note that $\left(\frac{\partial D_B}{dB}\right)^T$ can be written as

$$\left(\frac{\partial D_B}{dB}\right)^T = \sum_{i=1}^{m} [S_iB^T(BS_iB^T)^{-1}-\Lambda_iB^T(B\Lambda_iB^T)^{-1}](BS_iB^T)(B\Lambda_iB^T)^{-1}$$

$$= \sum_{i=1}^{m} [B^TBB^T - B^TBB^T](BS_iB^T)(B\Lambda_iB^T)^{-1}$$

$$=(0)$$

Q.E.D.

Remark 1 - In general, such a B satisfying the hypotheses of Theorem 2 will not exist. However, it will be shown in Remark 3 that the hypotheses of theorem 2 is satisfied when $m = 2$ and the classes have equal means. Although this case has no practical value, it is of interest since here a class of matrices which extremize D_B are readily available analytically.

Note that under the hypotheses of Theorem 2, it is true that

$$(B\Lambda_i B^T)^{-1} = B\Lambda_i^{-1} B^T$$

This is just a special case of the more general result:

$$(B\Lambda_i B^T)^{-1} = B\Lambda_i^{-1} B^T + B\Lambda_i^{-1}(I - B^T B)Y$$

for some Y and any B of rank k satisfying $BB^T = I$.

Remark 2: Note that if B satisfies $BB^T = I$ and if $(B^T B)$, S_i, and Λ_i $(i = 1,2,\ldots,m)$ are all diagonal matrices, then

$$\left(\frac{\partial D_B}{\partial B}\right)^T = 0$$

An example of a B satisfying $B^T B$ is a diagonal matrix is given by any selection of k out of n channels. Mathematically, B must satisfy $BB^T = I$, with elements b_{ij} satisfying

$$b_{ij}^2 = b_{ij}$$

Remark 3: Consider the particular case where $m = 2$ and $S_{12} = 0$. Then there exists an n by n nonsingular matrix P such that

$$P\Lambda_1 P^T = I_n \quad \text{and} \quad P\Lambda_2 P^T = W_2$$

where I_n is the n by n identity matrix and W_2 is a diagonal matrix. Then any matrix B such that $BB^T = I$ and with elements $b_{ij}^2 = b_{ij}$ satisfies

$$\left(\frac{\partial D_{(BP)}}{\partial (BP)} \right)^T = 0$$

SECTION 3 - A COMPARISON OF EXPRESSIONS

The following Theorem is proved in Reference 4, with the notation of Reference 4 being changed to agree with the notation of this note.

THEOREM - If two pattern classes π_1 and π_2 are normally distributed according to $N(\mu_1,\Lambda_1)$ and $N(\mu_2,\Lambda_2)$ respectively, then a necessary condition for the B-induced interclass divergence $D_B(i,j)$ to be an extremum is that the matrix B satisfy the following equation:

$$\sum_{i=1}^{k} (1 - \beta_i^{-2})(\Lambda_1 B^T - \beta_i \Lambda_2 B^T)\bar{b}_i \bar{b}_i^T$$
$$+ (\delta_{12}\delta_{12}^T B^T - \beta_{k+1}\Lambda_1 B^T)\bar{b}_{k+1}\bar{b}_{k+1}^T$$
$$+ (\delta_{12}\delta_{12}^T B^T - \beta_{k+2}\Lambda_2 B^T)\bar{b}_{k+2}\bar{b}_{k+2}^T$$

where β_i and $\bar{b}_i$ are the eigenvalues and eigenvectors of $(B\Lambda_2 B^T)^{-1}(B\Lambda_1 B^T)$;

β_{k+1}, $\bar{b}_{k+1}$ and β_{k+2}, $\bar{b}_{k+2}$ are the eigenvalues and eigenvectors of

$$(B\Lambda_1 B^T)^{-1}(B\,\delta_{12}\,\delta_{12}{}^T B^T) \quad \text{and} \quad (B\Lambda_2 B^T)^{-1}(B\,\delta_{12}\,\delta_{12}{}^T B^T)$$

respectively.

While the above expression is not too complicated, one is still faced with the bothersome task of obtaining the eigenvalues and eigenvectors (compare with theorem 1).

Finally, we present Babu's condition for the B-average interclass divergence to be an extremum (Reference 5). Again, the notation of Reference 5 has been changed to agree with the notation of this report.

<u>THEOREM</u> - Let a k by n matrix B of rank k extremize $D_B = \sum_{i=1}^{m}\sum_{j=1}^{m} D_B(i,j)$. Then it is necessary that B satisfy an equation of the form

$$\Gamma = \sum_{j=1}^{k}[(\sum_{i=1}^{m}\Lambda_i)B^T - \lambda_j(\sum_{i=1}^{m}\Lambda_i^{-1})^{-1}B^T]e_j e_j{}^T$$

$$+ \sum_{i=1}^{m}[\sum_{j=1}^{k}(A_i B^T - \lambda_{ij}\Lambda_i B^T)e_{ij}e_{ij}^T = 0,$$

where λ_j and e_j are the eigenvalues and their correspondingg eigenvectors of:

$$[\sum_{i=1}^{m}(B\Lambda_i B^T)^{-1}][\sum_{i=1}^{m}B\Lambda_i B^T]$$

and λ_{ij} and e_{ij} are the eigenvalues and their corresponding eigenvectors of:

$$(B\Lambda_i B^T)^{-1}\sum_{j=1}^{m}B\delta_{ij}\delta_{ij}{}^T B^T$$

and

$$A_i = \sum_{j=1}^{m} \delta_{ij}\delta_{ij}^T$$

Again, a comparison of the above Theorem with Theorem 1 suggests the desirability of using Theorem 1 to compute the gradient. Note that S_i and Λ_i $(i=1,2,\ldots,m)$ appearing in Theorem 1 are constant and need to be computed only once.

In addition, Babu's expression for Γ appears to be incorrect. In deriving the expression for Γ, Babu essentially assumes

$$[\sum_{i=1}^{m} (B\Lambda_i B^T)^{-1}] = B\left(\sum_{i=1}^{m} \Lambda_i^{-1}\right)^{-1} B^T \qquad (1)$$

(Equations (7) and (12) of Reference 5) to be true for arbitrary B. That the above identity is not true in general is evidenced by the following counter example; let

$$\Lambda_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \Lambda_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \qquad \Lambda_2^{-1} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix}$$

$$B = \alpha^T = (1 \quad 1)$$

The left side of equation 1 is

$$\frac{1}{\frac{1}{2} + \frac{1}{3}} = \frac{6}{5}$$

The right side of equation 1 is

$$\frac{2}{3} + \frac{1}{2} = \frac{7}{6}$$

SUMMARY

It has been shown that for m distinct classes with means μ_i and covariances Λ_i, upon performing the transformation $y = Bx$ where B is a k by n matrix of rank k, the average divergence in the space of reduced dimension may be written as

$$D_B = \frac{1}{2} \operatorname{tr}\{\sum_{i=1}^{m} (B\Lambda_i B^T)^{-1} (BS_i B^T)\} - \frac{(m)(m-1)}{2} k$$

where

$$S_i = \sum_{\substack{j=1 \\ j\neq 1}}^{m} [\Lambda_j + (\mu_i - \mu_j)(\mu_i - \mu_j)^T]$$

Also, if $\frac{\partial D_B}{\partial B}$ denotes the matrix whose i-j th element is $\frac{\partial D_B}{db_{ij}}$, where b_{ij} is the i-j th element of B, then

$$\left(\frac{\partial D_B}{\partial B}\right)^T = \sum_{i=1}^{m} [S_i B^T - \Lambda_i B^T (B\Lambda_i B^T)^{-1} (BS_i B^T)] (B\Lambda_i B^T)^{-1}$$

and

$$B\left(\frac{\partial D_B}{\partial B}\right)^T = 0$$

REFERENCES

1. Kullback, Solomon, Information Theory and Statistics, 1968 Dover Publications, New York.

2. Quirein, J. A., "Sufficients Statistics for the Divergence and Probability of Misclassification" Due to be published March 1, 1973

3. Anderson, T. W., An Introduction to Multivariate Statistical Analysis, 1958 John Wiley and Sons, Inc., New York

4. Tou, J. T., and Heydorn, R. P., 1967, in Computer and Information Sciences, Vol. 2, edited by J. T. Tou (New York:Academic Press)

5. Babu, C. C., and Kalra, S. N., "On Feature Extraction in Multiclass Pattern Recognition", Int. J. Control, 1972, Vol. 15, No. 3.

N73-29576 1

UNIVERSITY OF HOUSTON · Founded 1927

DEPARTMENT OF MATHEMATICS

UNIVERSITY OF HOUSTON HOUSTON, TEXAS

AN EVALUATION OF AN ALGORITHM FOR LINEAR INEQUALITIES AND ITS APPLICATION TO PATTERN CLASSIFICATION

JOHN JURGENSEN

SEPT. 1972

PREPARED FOR
EARTH OBSERVATION DIVISION , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004

N73-29577

DEPARTMENT OF MATHEMATICS

UNIVERSITY OF HOUSTON HOUSTON, TEXAS

DIVERGENCE AND NECESSARY CONDITIONS FOR EXTREMUMS

JOHN QUIREIN

NOV. 1972

PREPARED FOR
EARTH OBSERVATION DIVISION , JSC
UNDER
CONTRACT NAS-9-12777

3801 CULLEN BLVD.
HOUSTON, TEXAS 77004