SPACE RESEARCH COORDINATION CENTER

# AN IMPROVED GENERALIZED INVERSE ALGORITHM FOR LINEAR INEQUALITIES AND ITS APPLICATIONS

BY

L. C. GEARY AND C. C. LI

DEPARTMENT OF
ELECTRICAL ENGINEERING

SRCC REPORT NO. 122

UNIVERSITY OF PITTSBURGH
PITTSBURGH, PENNSYLVANIA

11 MARCH 1970

The Space Research Coordination Center, established in May, 1963, has the following functions: (1) it administers predoctoral and postdoctoral fellowships in space-related science and engineering programs; (2) it makes available, on application and after review, allocations to assist new faculty members in the Division of the Natural Sciences and the School of Engineering to initiate research programs or to permit established faculty members to do preliminary; work on research ideas of a novel character; (3) in the Division of the Natural Sciences it makes an annual allocation of funds to the interdisciplinary Laboratory for Atmospheric and Space Sciences; (4) in the School of Engineering it makes a similar allocation of funds to the Department of Metallurgical and Materials Engineering and to the program in Engineering Systems Management of the Department of Industrial Engineering; and (5) in concert with the University's Knowledge Availability Systems Center, it seeks to assist in the orderly transfer of new space-generated knowledge in industrial application. The Center also issues periodic reports of space-oriented research and a comprehensive annual report.

The Center is supported by an Institutional Grant (NsG-416) from the National Aeronautics and Space Administration, strongly supplemented by grants from the A. W. Mellon Educational and Charitable Trust, the Maurice Falk Medical Fund, the Richard King Mellon Foundation and the Sarah Mellon Scaife Foundation. Much of the work described in SRCC reports is financed by other grants, made to individual faculty members.

# AN IMPROVED GENERALIZED INVERSE ALGORITHM FOR LINEAR INEQUALITIES AND ITS APPLICATIONS

by

L. C. Geary and C. C. Li
Department of Electrical Engineering
University of Pittsburgh
Pittsburgh, Pennsylvania

CONTENTS

1970 Spring Joint Computer Conference

# An improved generalized inverse algorithm for linear inequalities and its applications

*by* L. C. GEARY* and C. C. LI

*University of Pittsburgh*
Pittsburgh, Pennsylvania

## INTRODUCTION

A great amount of research for the solution of linear inequalities has been undertaken in the past ten years. One of the reasons for this research is the development of linear separation approaches to pattern recognition[1–5,8–16] and threshold logic problems.[6,7,9] Both of these problems require the determination of a decision function or decision functions which, in the case of linear separation, involve a system of linear inequalities.

In this paper, an improved iterative algorithm will be developed for the solution of the set of linear inequalities which is written in the following equation:

$$Aw > 0. \tag{1}$$

This algorithm is an improvement of the Ho-Kashyap algorithm by choosing a criterion function

$$J(y) = 4 \sum_{i=1}^{N} (\cosh \tfrac{1}{2} y_i)^2 \tag{2}$$

*Presently with Gulf Research & Development, Company, Pittsburgh, Pa.

to be minimized where $y_i$ is the $i$th component of the $N$ by 1 vector $y$ defined below

$$y = Aw - b, b > 0. \tag{3}$$

The improvement lies in an acceleration of the Ho-Kashyap algorithm caused by a steeper gradient of $J(y)$ as can be seen when a comparison is made between the two criterion functions. Let $J_{hk}(y)$ designate the criterion function used in the Ho-Kashyap algorithm,

$$J_{hk}(y) = \| y \|^2 = \sum_{i=1}^{N} y_i^2. \tag{4}$$

Since $J(y)$ and $J_{hk}(y)$ reach their respective minimum when each $(\cosh \frac{1}{2}y_i)^2$ and each $y_i^2$ are respectively minimized, one can simply compare $J(y_i)$ and $J_{hk}(y_i)$, the convex functions of one variable only. Taking the gradients of $J(y_i)$ and $J_{hk}(y_i)$ with respect to $y_i$, one obtains

$$\frac{\partial J(y_i)}{\partial y_i} = 2y_i + \frac{2}{3!} y_i^3 + \frac{2}{5!} y_i^5 + \cdots \tag{5}$$

and

$$\frac{\partial J_{hk}(y_i)}{\partial y_i} = 2y_i. \tag{6}$$

It is clear that the absolute value of $\partial J(y_i)/\partial y_i$ is greater than the absolute value of $\partial J_{hk}(y_i)/\partial y_i$ everywhere except at $y_i = 0$ where they are equal. In general, the gradient $\partial J(y)/\partial y$ is greater than the gradient $\partial J_{hk}(y)/\partial y$ everywhere except at the origin $y = \mathbf{0}$. Since the gradient descent procedure is used in both algorithms, and since $y$ and $b$, or $y$ and $w$, are linearly related, it is conceivable that the proposed algorithm may have a higher convergence rate for a solution $w$.

As mentioned previously, $J(y)$ reaches a minimum when each term $(\cosh \frac{1}{2}y_i)^2$, $(i = 1, \ldots, N)$, is minimized. For each $(\cosh \frac{1}{2}y_i)^2$ to be a minimum, each $y_i$, $(i = 1, \ldots, N)$, must equal zero and $y = 0$ gives a desired solution. Since the $b_i$'s are only constrained to be positive, $J(y)$ can be minimized with respect to both $w$ and $b$ subject to the condition that $b > 0$. Note that it is not necessary to attain the minimum value of $J(y)$; in fact, a solution $w^*$ is obtained whenever $y \geq 0$ with $b > 0$ from which follows $Aw^* \geq b > 0$.

## DEVELOPMENT OF THE TWO-CLASS ALGORITHM

Let the matrix $A$, whose transpose is

$$A^t = [{}_1x_1, \ldots {}_{n_1}x_1, {}_1x_2, \ldots {}_{n_2}x_2],$$

be represented as

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ a_{N1} & a_{N2} & \cdots & a_{Nn} \end{bmatrix}, \tag{7}$$

where ${}_jx_i$ is an $n$ by 1 augmented pattern vector, $n = r + 1$, and $N = n_1 + n_2$, then the gradient of $J(y)$ with respect to $w$ is given by

$$\frac{\partial J(y)}{\partial w} = 2A^t s(y) \tag{8}$$

where

$$s^t(y) = [\sinh y_1, \cdots, \sinh y_N],$$

and the gradient of $J(y)$ with respect to $b$ is given by

$$\frac{\partial J(y)}{\partial b} = -2s(y), \tag{9}$$

where the derivative of a scalar with respect to a column vector is a column vector. Since $w$ is not constrained in any way $\partial J(y)/\partial w = 0$ implies $s(y) = 0$ which, in turn, implies $y_i = 0$ for all $i = 1, 2, \cdots, N$. Therefore, for a fixed $b > 0$, minimizing $J(y)$ with respect to $w$ gives

$$y = Aw - b = 0.$$

Solving the above equation for $w$, one obtains

$$w = A^\# b \tag{10}$$

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

On the other hand, for a fixed $w$, $\partial J(y)/\partial b = 0$ with $b > 0$ dictates a descent procedure of the following form, with $k$ denoting the iteration number:

$$b(k + 1) = b(k) + \Delta b(k) \tag{11}$$

where the components of $\Delta b_i(k)$, $i = 1, 2, \cdots, N$, of $\Delta b(k)$ are governed by

$$\Delta b_i(k) \propto \begin{cases} -\left(\dfrac{\partial J(y(k))}{\partial b}\right)_i = 2 \sinh y_i & \text{if } y_i > 0, \\ 0 & \text{if } y_i \leq 0. \end{cases} \tag{12}$$

Introduce a positive scalar $p(k)$ as the proportionality constant and rewrite equation (12) in the vector form,

$$\Delta b(k) = p(k) h(k), \tag{13}$$

where

$$h(k) = [h_i(k)] = [\sinh y_i(k) + |\sinh y_i(k)|]. \quad (i = 1, 2, \cdots, N) \tag{14}$$

As can be shown later, $p(k)$ may be chosen as equal to

$$p(k) = \frac{1}{\cosh y_{\max}(k)} \tag{15}$$

where

$$y_{\max}(k) = \operatorname*{Max}_i | y_i(k) | . \tag{16}$$

Substituting (13) into (11) and, from (10), writing

$$w(k+1) = w(k) + p(k)A^{\#}h(k), \tag{17}$$

one obtains the following algorithm:

$$\begin{cases} w(0) = A^{\#}b(0), b(0) > 0 \text{ but otherwise arbitrary} \\ y(k) = Aw(k) - b(k) \\ b(k+1) = b(k) + p(k)h(k) \\ w(k+1) = w(k) + p(k)A^{\#}h(k) \end{cases} \tag{18}$$

where $h(k)$ and $p(k)$ are given by equations (14) and (15) respectively. Note that in this algorithm $p(k)$ varies at each step and is a nonlinear function of $y(k)$. A recursive relation in $y(k)$ can also be obtained from (18),

$$y(k+1) = y(k) + p(k)(AA^{\#} - I)h(k). \tag{19}$$

Just like the Ho-Kashyap algorithm, it can be shown that the above algorithm (18) converges to a solution $w^*$ of the system of linear inequalities in a finite number of steps provided that a solution exists, and simultaneously acts as a test for the inconsistency of the linear inequalities. These properties are formally stated in Theorem I given in the next section.

## THEOREM I

Before discussing the main theorem, a lemma to be used in the proof of the theorem will be given first.

*Lemma 1:* Let one consider the set of linear inequalities (1) and the algorithm (18) to solve this set. Then

1) $y(k) \not\geq 0$ for any $k$;

and

2) if the set of linear inequalities is consistent,

then

$$y(k) \not\leq 0 \qquad \text{for any k.}$$

This lemma is the same as the one given by Ho and Kashyap[8] except that the iterative algorithm is different. The proof of the lemma is not given here since it is similar to the proof of Ho-Kashyap lemma. Recall again the notation used in the lemma: $y(k) \leq 0$ means that $y_i(k) \leq 0$ for all $i$ but $y$ possesses at least one negative component. This lemma is a rigorous statement that with a consistent set of linear inequalities $Aw > 0$, the elements of the vector $y(k)$ cannot be all non-positive.

*Theorem I:* Consider the set of linear inequalities (1) and the algorithm (18) to solve these inequalities, and let $V[y(k)] = \| y(k) \|^2$.

1) If the set of linear inequalities is consistent then

a) $\Delta V[y(k)] \triangleq V[y(k+1)] - V[y(k)] < 0$ and $\lim_{k\to\infty} V[y(k)] = 0$ implying convergence to a solution in an infinite number of steps; and

b) actually, a solution is obtained in a finite number of steps.

2) If the set of linear inequalities is inconsistent, then there exists a positive integer $k^*$ such that

$$\Delta V[y(k)] < 0 \qquad \text{for} \quad k < k^*$$
$$\Delta V[y(k)] = 0 \qquad \text{for} \quad k \geq k^*,$$

and

$$y(k) \not\leq 0 \qquad \text{for} \quad k < k^*$$
$$y(k) = y(k^*) \leq 0 \qquad \text{for} \quad k \geq k^*$$

and

$$w(k) = w(k^*) \qquad \text{for} \quad k \geq k^*$$
$$b(k) = b(k^*) \qquad \text{for} \quad k \geq k^*.$$

In other words, the occurrence of a nonpositive vector $y(k)$ at any step terminates the algorithm and indicates the inconsistency of the given set of linear inequalities.

*Proof:*

Part 1: Since the algorithm (18) can be rewritten as a recursive relation in $y(k)$ given by (19), and[8]

$$V[y(k)] = \| y(k) \|^2 > 0 \qquad \text{for all} \quad y(k) \neq 0 \tag{20}$$

$V[y(k)]$ can be considered as a Liapunov function for the nonlinear difference equation (19). Thus

$$\begin{aligned} \Delta V[y(k)] &\triangleq V[y(k+1)] - V[y(k)] \\ &= \| y(k+1) \|^2 - \| y(k) \|^2 \\ &= y^t(k+1)y(k+1) - y^t(k)y(k) \\ &= p(k)h^t(k)(AA^{\#} - I)^t y(k) \\ &\quad + p(k)y^t(k)(AA^{\#} - I)h(k) \\ &\quad + p^2(k)h^t(k)(AA^{\#} - I)^t(AA^{\#} - I)h(k). \end{aligned}$$

Since $(AA^{\#} - I)$ is hermitian idempotent, and $AA^{\#}y(k) = 0$, $\Delta V[y(k)]$ reduces to

$$\Delta V[y(k)] = -2p(k)h^t(k)y(k) + p^2(k)h^t(k)(I - AA^{\#})h(k). \tag{21}$$

Further simplification leads to

$$\Delta V[y(k)] = -[y(k) + |y(k)|]^t[p(k)R(k) + p^2(k)R(k)(AA^\# - I)R(k)][y(k) + |y(k)|]$$
$$= -\|y(k) + |y(k)|\|^2_{\{p^2(k)R(k)AA^\#R(k) + p(k)R(k) - p^2(k)R^2(k)\}}. \quad (22)$$

where $R = \operatorname{diag}\left[\frac{\sinh y_1}{y_1}, \cdots, \frac{\sinh y_N}{y_N}\right]$.

For $\Delta V[y(k)]$ to be negative semidefinite, $\Delta V[y(k)] = 0$ only if $y(k) = 0$ or $y(k) \leq 0$, the matrix

$$[p^2(k)R(k)AA^\#R(k) + p(k)R(k) - p^2(k)R^2(k)]$$

must be positive definite. $AA^\#$ is positive semidefinite because $AA^\#$ is hermitian idempotent, $x^tAA^\#x \geq 0$ for any $x$; it follows that $z^tRAA^\#Rz \geq 0$ for any $z$; hence $RAA^\#R$ is also positive semidefinite. Now one can choose a $p(k)$ such that $[p(k)R(k) - p^2(k)R^2(k)]$ is positive definite. $[p(k)R(k) - p^2(k)R^2(k)]$ is positive definite if

$$[p(k)r_{ii}(k) - p^2(k)r_{ii}^2(k)] > 0 \quad \text{for all} \quad i = 1, 2, \cdots, N. \quad (23)$$

Since $r_{ii}(k) = \sinh y_i/y_i > 0$ for all $i$ and $p(k)$ is restricted to be positive, the above condition reduces to the condition,

$$1 - p(k)r_{ii}(k) > 0 \quad \text{for all} \quad i = 1, 2, \cdots, N. \quad (24)$$

For $p(k)$ chosen in equation (15),

$$p(k) = \frac{1}{\cosh y_{\max}(k)}$$

$$p(k)r_{ii}(k) = \frac{1}{\cosh y_{\max}(k)} \frac{\sinh y_i(k)}{y_i(k)}$$
$$= \frac{\sinh y_i(k)}{y_i(k) \cosh y_{\max}(k)}$$
$$= \frac{\sum_{n=0}^{\infty} \frac{y_i^{2n}(k)}{(2n+1)!}}{\sum_{n=0}^{\infty} \frac{y_{\max}^{2n}(k)}{(2n)!}} < 1.$$

Thus the condition (24) is satisfied and $[p(k)R(k) - p^2(k)R^2(k)]$ is positive definite for

$$p(k) = \frac{1}{\cosh y_{\max}(k)}.$$

Then $\Delta V[y(k)]$ has the desired property of negative semidefinite for $p(k) = 1/\cosh y_{\max}(k)$ and for any finite $y(k)$.

From equation (22) one notes that $\Delta V[y(k)]$ equals zero if and only if $y(k) = 0$ or $y(k) \leq 0$. Since it is assumed that the set of linear inequalities (1) is consistent, and from the lemma $y(k) \nleq 0$, therefore

$$\Delta V[y(k)] < 0 \quad \text{for all} \quad y(k) \neq 0 \quad (25)$$
$$= 0 \quad \text{if} \quad y(k) = 0.$$

By Liapunov's stability criterion, the equilibrium state $y = 0$ of the discrete system (19) can be reached asymptotically, i.e., $\lim_{k\to\infty} \| y(k) \|^2 = 0$, which corresponds to a solution $w^{**}$ with $Aw^{**} = b > 0$. This completes the proof of Part 1(a).

To prove the convergence of the algorithm (18) in a finite number of steps, one notes that $b(k)$ is a nondecreasing vector. Let $b^t(0) = [1, 1, \cdot\cdot 1]$, then

$$b^t(k) \geq b^t(0) \geq [1, 1, \cdots, 1] \quad \text{for any} \quad k > 0.$$

Since $Aw(k) = b(k) + y(k)$, $| y^t(k) | < [1, 1, \cdots, 1]$ implies $Aw^*(k) > 0$ when a solution $w^*$ is reached. But $V[y(k)] \leq 1$ implies $| y^t(k) | < [1, 1, \cdots, 1]$. Since $V[y(k)]$ converges to zero in infinite time, it must converge to the region $V[y(k)] = 1$ in finite time, hence $| y^t(k) | < [1, 1, \cdots, 1]$, $Aw(k) > 0$, and a solution $w^* = w(k)$ is obtained in a finite number of steps. This completes the proof of Part 1(b).

*Part 2:* It has been proved in Part 1 that $V[y(k)]$ is negative semidefinite independent of the consistency of the linear inequalities. Now, if the set of linear inequalities (1) is inconsistent, one notes that $y(k)$ cannot be 0 and hence $V[y(k)]$ cannot become zero for any $k > 0$. There must exist a value of $k$, called $k^*$, such that

$$\Delta V[y(k)] < 0 \quad \text{for} \quad 0 \leq k < k^*$$
$$= 0 \quad \text{for} \quad k = k^*,$$
$$y(k) \nleq 0 \quad \text{for} \quad 0 \leq k < k^*.$$

But $V[y(k^*)] = 0$ if either $y(k^*) = 0$ or $y(k^*) \leq 0$. Since $y(k^*) \neq 0$, this implies $y(k^*) \leq 0$ and hence, from (14), $h(k^*) = 0$. Equation (19) indicates that

$$y(k) = y(k^*) \leq 0 \quad \text{for all} \quad k \geq k^*$$

As a consequence, one obtains

$$\Delta V[y(k)] = 0 \qquad \text{for all} \quad k \geq k^*$$

$$h(k) = 0 \qquad \text{for all} \quad k \geq k^*$$

$$w(k) = w(k^*) \qquad \text{for all} \quad k \geq k^*$$

$$b(k) = b(k^*) \qquad \text{for all} \quad k \geq k^*$$

This completes the proof of the theorem.

*An Optimum Choice of* The Scalar p(k)

The choice of $p(k) = 1/\cosh y_{\max}(k)$ in the previous section is only one of many possible choices of $p(k)$ for the convergence of the algorithm (18). The convergence rate may be further improved by choosing a $p(k)$ such that the decrease in the Lyapunov function $V[y(k)]$ is maximized at every step, that is, $-\Delta V[y(k)]$ is maximized with respect to $p(k)$. Taking the partial derivative of $\Delta V[y(k)]$ in equation (22) with respect to $p(k)$ leads to an optimum value of $p(k)$ given by

$$p(k) = \frac{[y(k) + |y(k)|]^t R(k)[y(k) + |y(k)|]}{2[y(k) + |y(k)|]^t R(k) \cdot [I - AA^{\#}]R(k)[y(k) + |y(k)|]} \tag{26}$$

provided that $I - AA^{\#} > 0$. For this value of $p(k)$, $\Delta V[y(k)]$ is negative definite in $[y(k) + |y(k)|]$ which is required in the convergence proof of the algorithm (18). A flow chart summarizing the above procedure is shown in Figure 1.

EXAMPLES

The algorithm (18) has been applied to pattern recognition and switching theory problems. For switching theory problems the generalized inverse of the $N$ by $n$ pattern matrix $A$ is simplified to

$$A^{\#} = 2^{-(n-1)}A^t.$$

Two example problems will be presented, one in switching theory and the other in pattern recognition.

*Example 1:* Consider a Boolean function of eight binary variables which corresponds to the separation of the two classes:

Class $C_1$ = (127, 191, 215, 217 to 255)

Class $C_2$ = (0 to 126, 128 to 190, 192 to 214, 216).

Here $m = 2^r = 256$ and $n = r + 1 = 9$, where $r$ is the number of binary variables. For

$$b^t(0) = [0.1, 0.1, 0.1, \cdots, 0.1, 0.1, 0.1]$$

and $p(k)$ given in equation (26), the algorithm terminates after the tenth iteration and gives a solution weight vector $w$ for the switching function,

$$w^t = [0.3732, 0.2278, 0.2278, 0.1654, 0.0769, 0.0569, 0.0247, 0.0247, 0.0247],$$

The same example was solved using the Ho-Kashyap algorithm.[8] It required 229 iterations with the same initial $b(0)$. The solution weight vector $w$ for the Ho-Kashyap algorithm is

$$w^t = [0.5741, 0.3447, 0.3447, 0.2425, 0.1155, 0.1080, 0.0436, 0.0436, 0.0436].$$

The computing time for the proposed algorithm was 50 seconds on IBM 7090 with a cost of $1.50, while the Ho-Kashyap algorithm required 80 minutes with a cost of $23.50. Thus the proposed algorithm not only reduced the number of required iterations but also the computing time and cost to solve the problem. It was observed, that for $0.5 \geq b_i(0) \geq 0.001$ and $p(k)$ given by equation (26), for all examples tried by the authors that the number of iterations was less than or equal to the number of iterations required by the Ho-Kashyap algorithm. In some cases the number of iterations was reduced by a factor of 25.[17]

*Example 2:* The proposed algorithm was also applied to a preliminary study of a biomedical pattern recognition problem. The problem is to investigate whether or not a change exists in the diurnal cycle of an individual person upon a change in his environmental condition or physiological state and if such a change may be used to diagnose physical ailments under strictly controlled conditions by measuring the amounts of electrolytes present in urine samples every three hours.[18] The data used in this example consisted of thirteen sample patterns under two different conditions. Each pattern has eight components which represent the mean excretion rates of an electrolyte for each three-hour period of the twenty-four hour cycle. Thus $N = 13$ and $n = r + 1 = 8 + 1 = 9$; the size of the pattern matrix $A$ is 13 by 9. The pattern matrix $A$ is shown in Table 1. Let $b^t(0) = [0.1, 0.1, \cdots 0.1]$. For this problem the Ho-Kashyap algorithm with $p = 1$ required 7 iterations to determine the separability. However, the proposed algorithm with $p(k)$ given by equation (26) required only two iterations, where $p(1) = 5.270684$ and $p(2) = 3.197152$. The problem is linearly separable and a solution weight vector $w$ obtained by the proposed algorithm is

$$w^t(2) = [-13.6089, 2.5915, 1.6847, 2.2314, 0.3414, 3.0077, 1.8428, 1.6559, 0.0096]$$

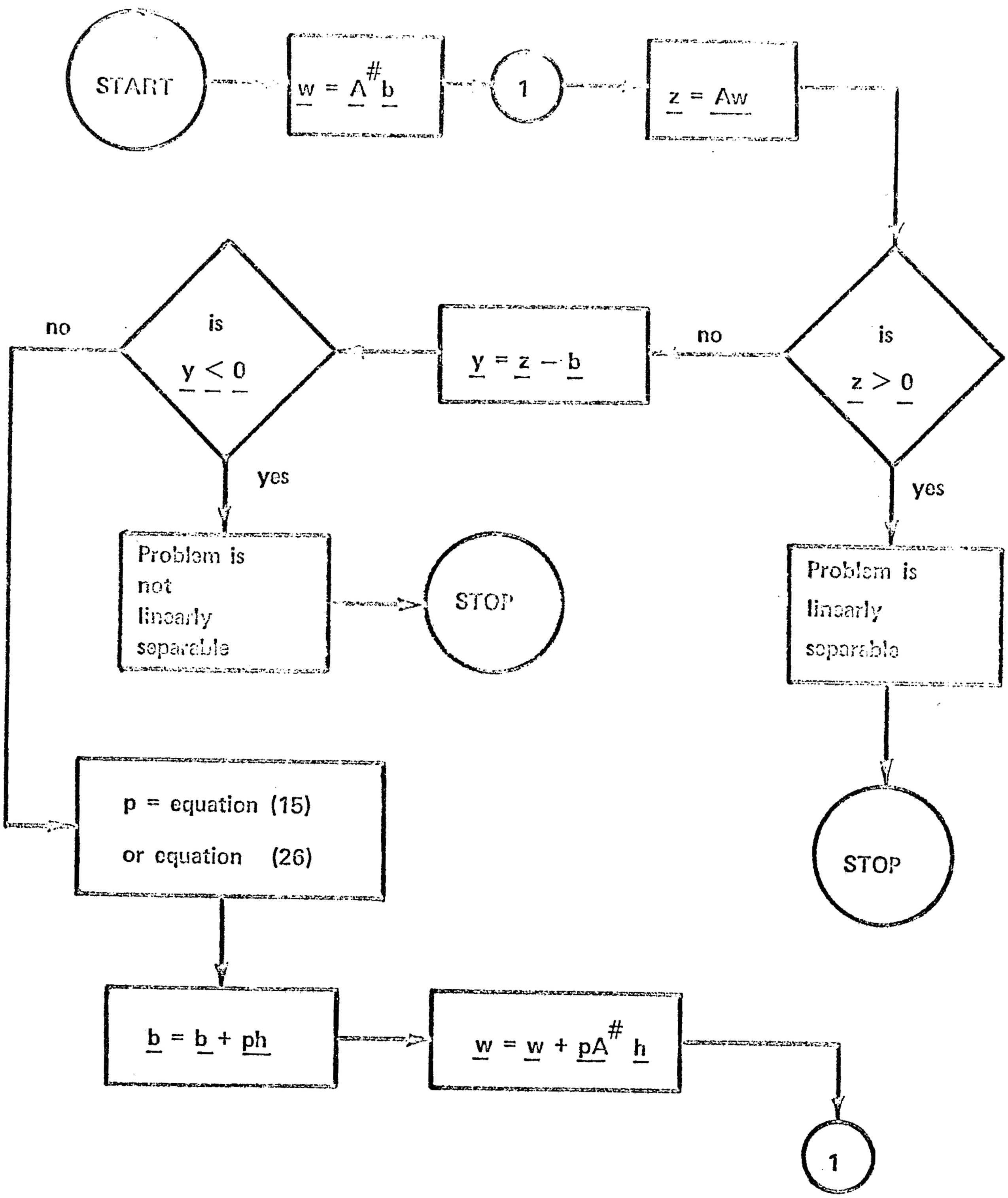


Figure 1—Flow chart of the proposed 2-class algorithm

TABLE 1—The Pattern Matrix $A$ for Example 2

| $x_0$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ |
|---|---|---|---|---|---|---|---|---|
| 1.00 | .96 | 1.19 | 1.35 | .75 | 1.12 | .94 | .73 | .97 |
| 1.00 | .75 | 1.19 | 1.35 | 1.06 | 1.07 | .97 | .81 | .81 |
| 1.00 | .80 | 1.13 | .85 | .90 | 1.14 | 1.27 | 1.01 | .88 |
| 1.00 | .66 | 1.40 | 1.25 | 1.09 | 1.54 | .79 | .27 | .00 |
| 1.00 | 2.04 | 1.14 | 1.10 | .57 | .62 | .66 | .47 | 1.39 |
| 1.00 | 1.02 | 1.32 | 1.06 | 1.03 | 1.07 | 1.16 | .77 | .57 |
| −1.00 | −.48 | −1.01 | −.68 | −.72 | −1.76 | −1.25 | −.62 | −1.47 |
| −1.00 | −.55 | −.55 | −1.04 | −.91 | −1.40 | −1.17 | −1.28 | −1.09 |
| −1.00 | −.87 | −.79 | −1.34 | −.86 | −.44 | −2.15 | −.82 | −.74 |
| −1.00 | −.09 | −.70 | −.67 | −.80 | −1.93 | −1.29 | −1.14 | −1.39 |
| −1.00 | −1.12 | −1.75 | −.51 | −.72 | −1.25 | −.46 | −.89 | −1.29 |
| −1.00 | −1.20 | −1.47 | −.60 | −.96 | −1.13 | −.89 | −.74 | −1.00 |
| −1.00 | −1.43 | −1.79 | −.68 | −.75 | −.82 | −.56 | −.94 | −1.04 |

## EXTENSION TO THE MULTICLASS ALGORITHM

The problem of multiclass patterns classification is that it must be determined to which of the $R$ different classes, $C_1, C_2, \cdots, C_R$, a given pattern vector, $x$, belongs. If the $R$-class patterns are linearly separable, there exist $R$ weight vectors $w_j$ to construct $R$ discriminant functions $g_j(x)$, $(j = 1, 2, \cdots, R)$, such that

$$g_j(x) = x^t w_j > x^t w_i = g_i(x) \quad \text{for all } i \neq j,\ x \in C_j. \qquad (27)$$

Chaplin and Levadi[10] have formulated another set of inequalities which can be considered as a representation of linear separation of $R$-class patterns. This set of inequalities is

$$\| x^t U - e_j^t \| < \| x^t U - e_i^t \| \text{ for all } i \neq j,\ x \in C_j \quad (28)$$
$$\text{for all } j = 1, 2 \cdots, R$$

where $U$ is an $n \times (R-1)$ weight matrix and the vectors $e_j$'s are the vertex vectors of a $R-1$ dimensional equilateral simplex with its centroid at the origin. If each $e_j$ is associated with one class, $x$ is classified according to the nearest neighborhood of the mapping $x^t U$ to the vertices. Inequalities (28) are, in fact, equivalent to inequalities (27) with

$$w_j = U e_j, \qquad (j = 1, 2, \cdots, R) \qquad (29)$$

Let the $N \times n$ pattern matrix $A$ be defined in the following manner,

$$A = \begin{bmatrix} A_1 \\ -- \\ \cdot \\ \cdot \\ \cdot \\ -- \\ A_j \\ -- \\ \cdot \\ \cdot \\ \cdot \\ -- \\ A_R \end{bmatrix} \overset{\Delta}{=} \begin{bmatrix} {}_1x_1^t \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_1}x_1^t \\ --- \\ \cdot \\ \cdot \\ \cdot \\ --- \\ {}_1x_j^t \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_j}x_j^t \\ --- \\ \cdot \\ \cdot \\ \cdot \\ --- \\ {}_1x_R^t \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_R}x_R^t \end{bmatrix} \qquad (30)$$

where $A_j$ is an $n_j \times n$ submatrix having as its rows $n_j$ transposed pattern vectors of class $C_j$,

$${}_lx_j^t, \qquad (l = 1, 2, \cdots, n_j),$$

where the right subscript denotes the pattern class and the left subscript denotes the $l$th pattern in that class, and $N = n_1 + n_2 + \cdots + n_R$. Designate the $n \times (R-1)$ weight matrix $U$ as composed of $(R-1)$ column vectors $u_q$, $(q = 1, 2, \cdots, R-1)$,

$$U = [u_1 \cdots u_q \cdots u_{R-1}]. \qquad (31)$$

Also define an $N \times (R-1)$ matrix $B$ as

$$B = \begin{bmatrix} B_1 \\ -- \\ \cdot \\ \cdot \\ \cdot \\ -- \\ B_j \\ -- \\ \cdot \\ \cdot \\ \cdot \\ -- \\ B_R \end{bmatrix} \overset{\Delta}{=} \begin{bmatrix} {}_1b_1^t \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_1}b_1^t \\ --- \\ \cdot \\ \cdot \\ \cdot \\ --- \\ {}_1b_j^t \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_j}b_j^t \\ --- \\ \cdot \\ \cdot \\ \cdot \\ --- \\ {}_1b^t \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_R}b_R^t \end{bmatrix} \qquad (32)$$

whose row vectors ${}_lb_j^t$, $(j = 1, 2, \cdots, R; l = 1, 2, \cdots, n_j)$, correspond to the class groupings in the $A$ matrix and satisfy the following inequalities

$${}_lb_j^t(e_j - e_i) > 0 \qquad \text{for all} \quad i \neq j \qquad (33)$$
$$\text{for all} \quad j = 1, 2, \cdots, R.$$

$B_j$ is an $n_j \times (R-1)$ submatrix of $B$, $j = 1, 2, \cdots, R$. Let an $N \times (R-1)$ matrix $Y$ be defined as

$$Y \overset{\Delta}{=} AU - B. \qquad (34)$$

The representation of $Y$ may be in the form of either an array of $(R-1)$ column vectors, $y_q$, $(q = 1, 2, \cdots, R-1)$,

$$Y = [y_1 \cdots y_q \cdots y_{R-1}] \qquad (35)$$

or an array of $N$ row vectors ${}_lY_j$, $(j = 1, 2, \cdots, R;$ $l = 1, 2, \cdots, n_j)$, corresponding to the class groupings in the $A$ matrix,

$$Y = \left[\begin{array}{c} Y_1 \\ \hline \cdot \\ \cdot \\ \cdot \\ \hline Y_j \\ \hline \cdot \\ \cdot \\ \cdot \\ \hline Y_R \end{array}\right] \triangleq \left[\begin{array}{c} {}_1Y_1 \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_1}Y_1 \\ \hline \cdot \\ \cdot \\ \cdot \\ \hline {}_1Y_j \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_j}Y_j \\ \hline \cdot \\ \cdot \\ \cdot \\ \hline {}_1Y_R \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_R}Y_R \end{array}\right] \tag{36}$$

where $Y_j$ is an $n_j \times (R-1)$ submatrix of $Y$,

$$Y_j = A_jU - B_j \tag{37}$$

or

$$ {}_lY_j = {}_lx_j{}^tU - {}_lb_j{}^t \qquad j = 1, 2, \cdots, R$$
$$l = 1, 2, \cdots, n_j.$$

The set of linear inequalities which will be discussed in this paper is

$$A_jU(e_j - e_i) > 0 \qquad \text{for all} \quad i \neq j \tag{38}$$
$$\text{for all} \quad j = 1, 2, \cdots, R$$

Associated with it is another set of linear inequalities

$$Y_j(e_j - e_i) = (A_jU - B_j)(e_j - e_i) > 0 \tag{39}$$
$$\text{for all} \quad i \neq j$$
$$\text{for all} \quad j = 1, 2, \cdots, R$$

or

$${}_lY_j(e_j - e_i) = ({}_lx_j{}^tU - {}_lb_j{}^t)(e_j - e_i) > 0$$
$$\text{for all} \quad i \neq j$$
$$\text{for all} \quad j = 1, 2, \cdots R$$
$$\text{for all} \quad l = 1, 2, \cdots n_j.$$

Since, by (33), $B_j(e_j - e_i)$ is constrained to have positive components for all $i \neq j$, inequalities (39) implies the inequalities (38) and hence (27) or (28). When inequalities (38) are satisfied for all $i \neq j$ and for all $j = 1, 2, \cdots, R$, a solution weight matrix $U$ is reached which will give linear classification of $R$-class patterns; that is, if

$$x^tU(e_j - e_i) > 0 \qquad \text{for all} \quad i \neq j$$

then $x$ is classified as of class $C_j$.

## DEVELOPMENT OF THE MULTI-CLASS ALGORITHM

For the notational simplicity in the derivation of the gradient function to be developed below, let the matrices $A$, $U$, $B$, and $Y$ in equations (30), (31), (32), and (35) be represented respectively as

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{1n} \\ \cdot\cdot\cdot & \cdot\cdot\cdot & \cdot\cdot\cdot \\ a_{N1} & a_{N2} & a_{Nn} \end{bmatrix} \tag{40}$$

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{1,R-1} \\ \cdot\cdot\cdot & \cdot\cdot\cdot & \cdot\cdot\cdot \\ u_{n1} & u_{n2} & u_{n,R-1} \end{bmatrix} \tag{41}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{1,R-1} \\ \cdot\cdot\cdot & \cdot\cdot\cdot & \cdot\cdot\cdot \\ b_{N1} & b_{N2} & b_{N,R-1} \end{bmatrix} \tag{42}$$

and

$$Y = \begin{bmatrix} y_{11} & y_{12} & y_{1,R-1} \\ \cdot\cdot\cdot & \cdot\cdot\cdot & \cdot\cdot\cdot \\ y_{N1} & y_{N2} & y_{N,R-1} \end{bmatrix} \tag{43}$$

Substituting these into equation (34), one obtains

$$y_{ij} = \sum_{k=1}^{n} a_{ik}u_{kj} - b_{ij}. \tag{44}$$

Let $C(Y)$ be an $N \times (R-1)$ matrix defined by

$$C(Y) = [c_{ij}] \triangleq [\cosh \tfrac{1}{2}y_{ij}] \tag{45}$$
$$(i = 1, \cdots, N; j = 1, \cdots, R-1).$$

The criterion function $J(Y)$ to be minimized is chosen as the trace of $4C^t(Y)C(Y)$,

$$J(Y) \triangleq \mathrm{Tr}\,(4C^tC) = \sum_{i=1}^{N}\sum_{j=1}^{R-1} J_{ij}(Y) \tag{46}$$

where

$$J_{ij}(Y) = 4\,(\cosh \tfrac{1}{2}y_{ij})^2.$$

Determine the gradients of $J(Y)$ with respect to both $U$ and $B$,

$$\frac{\partial J(Y)}{\partial U} = 2A^tS(Y) \tag{47}$$

$$\frac{\partial J(Y)}{\partial B} = -2S(Y) \tag{48}$$

where $S(Y)$ is an $N \times (R-1)$ matrix with the following representation

$$S(Y) \triangleq [\sinh y_{ij}]$$

$$(i = 1, 2, \cdots, N; j = 1, \cdots, R-1)$$

$$= \begin{bmatrix} S_1(Y) \\ ----- \\ S_j(Y) \\ ----- \\ \cdot \\ \cdot \\ \cdot \\ ----- \\ S_R(Y) \end{bmatrix} \triangleq \begin{bmatrix} {}_1S_1(Y) \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_1}S_1(Y) \\ ------ \\ {}_1S_j(Y) \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_j}S_j(Y) \\ ------ \\ \cdot \\ \cdot \\ \cdot \\ ------ \\ {}_1S_R(Y) \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_R}S_R(Y) \end{bmatrix} \quad (49)$$

and ${}_lS_j(Y)$ is a row vector of the following form

$${}_lS_j(Y) = [{}_lS_{j1}(Y),\ {}_lS_{j2}(Y), \cdots, {}_lS_{j(R-1)}(Y)]$$
$$= [\sinh y_{(n_{j-1}+l),1}, \cdots, \sinh y_{(n_{j-1}+l),R-1}]. \quad (50)$$

Since $U$ is not constrained in any manner, $\partial J(Y)/\partial U = 0$ implies that $S(Y) = 0$, which, in turn, implies that $\sinh y_{ij} = 0$ and hence $y_{ij} = 0$ for all $i = 1, \cdots, N$ and $j = 1, 2, \cdots, R-1$. Therefore, for $\partial J(Y)/\partial U = 0$ and a fixed $B$,

$$Y = AU - B = 0$$

which gives a least square fit of

$$U = A^{\#}B. \quad (51)$$

On the other hand, for a fixed $U$ and the constraint $B_j(e_j - e_i) > 0$ for all $i \neq j$ as given in (33), one might attempt to increment $B$ according to the following gradient descent procedure to reduce $J(Y)$ at each step,

$$B(k+1) = B(k) + \delta B(k) \quad (52)$$

where the $q$th element, $\delta[{}_lb_{jq}(k)]$, of $\delta[{}_lb_j{}'(k)]$ in $\delta B_j(k)$ is given by

$$\delta[{}_lb_{jq}(k)] = \begin{cases} -p(k)\ {}_l\left[\dfrac{\partial J(Y)(k)}{\partial B}\right]_{jq} = 2p(k)\ {}_lS_{jq}(Y(k)), & \text{if } {}_lY_j(k)(e_j - e_q) > 0 \text{ for any } q \neq j \\ 0 & \text{if } {}_lY_j(k)(e_j - e_q) \leq 0 \text{ for any } q \neq j. \end{cases}$$

However, ${}_lY_j(k)(e_j - e_q) > 0$ does not imply ${}_lS_j(Y(k))(e_j - e_q) > 0$. In order to make $\delta[{}_lb_j{}'(k)] \cdot (e_j - e_q) \geq 0$ so that (33) can be satisfied at each step, a modified gradient descent procedure, similar to the one adopted in Teng and Li's generalization of the Ho-Kashyap algorithm [16], is to be used. Let a $(R-1) \times (R-1)$ non-singular matrix $E_j$ be defined as[16]

$$E_j = [e_j - e_1, \cdots, e_j - e_{j-1}, e_j - e_{j+1}, \cdots, e_j - e_R]. \quad (53)$$

Also define

$$Z_j = Y_jE_j \qquad \text{for all} \quad j = 1, 2, \cdots, R. \quad (54)$$

The increment $\delta[{}_lb_{jq}(k)]$ is then given in terms of

$$\delta[{}_lb_j{}'(k)E_j]_q = \begin{cases} 2p(k)\ {}_lS_{jq}(Z(k)) = p(k)[{}_lS_{jq}(Z(k)) + {}_l\Lambda_{jq}(k)] & \text{if } {}_lZ_{jq}(k) = {}_lY_j(k)(e_j - e_q) > 0 \\ 0 & \text{if } {}_lZ_{jq}(k) = {}_lY_j(k)(e_j - e_q) \leq 0 \end{cases} \quad (55)$$

where

$${}_l\Lambda_{jq}(k) = {}_lS_{jq}(Z(k)) \operatorname{Sgn}({}_lZ_{jq}(k)) \quad (56)$$

and, following (50),

$${}_lS_{jq}(Z(k)) = \operatorname{Sinh} {}_lZ_{jq}(k). \quad (57)$$

Putting into vector representation,

$$\delta[{}_lb_j(k)E_j] = p(k)[{}_lS_j(Z(k)) + {}_l\Lambda_j(k)] \quad (58)$$

or

$$\delta[{}_lb_j(k)] = p(k)[{}_lS_j(Z(k)) + {}_l\Lambda_j(k)]E_j^{-1}$$
$$= p(k)\ {}_lH_j(Y(k)) \quad (59)$$

where

$${}_lH_j(Y(k)) \triangleq [{}_lS_j(Z(k)) + {}_l\Lambda_j(k)]E_j^{-1}. \quad (60)$$
$$H_j(Y(k)) = [S_j(Z(k)) + \Lambda_j(k)]E_j^{-1}$$

$$H(Y(k)) = \begin{bmatrix} H_1(Y(k)) \\ \cdot \\ \cdot \\ \cdot \\ H_j(Y(k)) \\ \cdot \\ \cdot \\ \cdot \\ H_R(Y(k)) \end{bmatrix} = \begin{bmatrix} {}_1H_1(Y(k)) \\ \cdot \\ \cdot \\ \cdot \\ {}_lH_j(Y(k)) \\ \cdot \\ \cdot \\ \cdot \\ {}_{n_R}H_R(Y(k)) \end{bmatrix}$$

$$= [h_1(Y(k))\cdots h_q(Y(k))\cdots h_{R-1}(Y(k))]. \tag{61}$$

It follows from (58) and (56) that

$$\delta[{}_lb_j(k)](e_j - e_i) \geq 0 \qquad \text{for all} \quad i \neq j \qquad \text{and for all } j.$$

Then, from (59),

$$\delta[B(k)] = p(k)H(Y(k)).$$

Substituting the above equation into (52), one has

$$B(k+1) = B(k) + p(k)H(Y(k)) \tag{62}$$

Using the above equation in (51), one has

$$\begin{aligned} U(k+1) &= A^{\#}B(k+1) \\ &= A^{\#}\{B(k) + p(k)H[Y(k)]\} \\ &= U(k) + p(k)A^{\#}H[Y(k)] \end{aligned} \tag{63}$$

Therefore, an iterative algorithm to solve for $U$ can be proposed in the following:

$$\begin{cases} U(0) = A^{\#}B(0) \\ Y(k) = AU(k) - B(k), \qquad Z_j(k) = Y_j(k)E_j \\ B(k+1) = B(k) + p(k)H[Y(k)], \\ H_j(Y(k)) = [S_j(k) + \Lambda_j(k)]E_j^{-1} \\ U(k+1) = U(k) + p(k)A^{\#}H[Y(k)] \end{cases} \tag{64}$$

where $p(k)$ may be chosen as equal to

$$p(k) = \frac{\sum_{j=1}^{R}\sum_{l=1}^{n_j}\{{}_l\epsilon_j(k) + {}_lH_j(Y(k))(E_j{}^t)^{-1}R^{-1} \cdot ({}_lZ_j(k))E_j{}^t{}_lH_j(Y(k))\}}{2\sum_{q=1}^{R-1} h_q{}^t(I - AA^{\#})h_q} \tag{65}$$

provided that

$$\sum_{j=1}^{R}\sum_{l=1}^{n_j}\{{}_l\epsilon_j(k) + {}_lH_j(Y(k))(E_j{}^t)^{-1}R({}_lZ_j(k))E_j^t \cdot H_j(Y(k))\} > 0 \tag{66}$$

where[17]

$$R({}_lZ_j) \triangleq \operatorname{diag}[r_{11}({}_lZ_j), \cdots, r_{R-1,R-1}({}_lZ_j)] \tag{67}$$

$$(j = 1, 2, \cdots, R; \qquad l = 1, 2, \cdots, n_j)$$

$$r_{qq}({}_lZ_j) \triangleq \frac{\operatorname{Sinh} {}_lZ_{jq}}{{}_lZ_{jq}} \geq 1, \qquad (q = 1, \cdots, R-1). \tag{68}$$

$${}_l\epsilon_j \triangleq [{}_lZ_jR({}_lZ_j) + {}_l\Lambda_j](E_j{}^tE_j)^{-1}R^{-1}({}_lZ_j) \cdot [{}_lZ_jR({}_lZ_j) - {}_l\Lambda_j]^t \geq 0 \qquad \text{for all} \quad j \qquad \text{and all} \quad l. \tag{69}$$

The initial $B$ matrix, $B(0)$, may be chosen from

$$B^t(0) = \beta[e_1, \cdots e_1 \mid \cdots \mid e_j, \cdots, e_j \mid \cdots \mid e_R, \cdots, e_R], \qquad \beta > 0 \tag{70}$$

A recursive relation in $Y(k)$ is also obtained as follows:

$$Y(k+1) = Y(k) + p(k)(AA^{\#} - I)H[Y(k)] \tag{71}$$

This algorithm is a convergent algorithm for the solution $U$ of the set of linear inequalities (38). The nonlinear separability of the multiclass patterns can also be detected by observing at a certain step $k^*$

$$Y_j(k^*)(e_j - e_i) \leq 0 \qquad \text{for all} \quad i \neq j \qquad \text{for all} \quad j = 1, 2, \cdots, R.$$

## CONVERGENCE PROOF OF THE MULTI-CLASS ALGORITHM

The convergence of the proposed multi-class algorithm can be proved in the following steps.

*Lemma 2.* Consider the set of inequalities (38) and the algorithm (64) to solve it. Then

1) $Y_j(k)(e_j - e_i) \not\geq 0$ for all $i \neq j$
for all $j = 1, 2, \cdots, R$
for any $k$

2) If (38) is consistent, then
$Y_j(k)(e_j - e_i) \not\leq 0$ for all $i \neq j$
for all $j = 1, 2, \cdots, R$
for any $k$

This lemma can be proved by contradiction.[17,16]

*Theorem II:* Consider the set of linear inequalities (38) and the algorithm (64) to solve it, and let

$$V[Y(k)] = \| Y(k) \| \triangleq \mathrm{Tr}\,[Y^t(k)Y(k)]$$
$$= \sum_{q=1}^{R-1} \| y_q(k) \|^2 = \sum_{j=1}^{R} \sum_{i=1}^{n_j} \| {}_iY_j(k) \|^2 \quad (72)$$

1) If the set of linear inequalities is consistent, then

   a) $\Delta V[Y(k)] \triangleq V[Y(k+1)] - V[Y(k)] < 0$

   and

   $\lim_{k\to 0} V[Y(k)] = 0$ implying convergence to a solution in an infinite number of iterations; and

   b) a solution is obtained in a finite number of steps.

2) If the set of linear inequalities is inconsistent, then there exists a positive integer $k^*$ such that

$$V[Y(k)] < 0 \qquad \text{for} \quad k < k^*$$
$$V[Y(k)] = 0 \qquad \text{for} \quad k \geq k^*$$
$${}_lY_j(k)(e_j - e_i) \nleq 0 \quad \text{for } k < k^* \quad \text{for all } i \neq j \quad \text{for all } j = 1, 2, \cdots, R$$
$${}_lY_j(k)(e_j - e_i) = Y_j(k^*)(e_j - e_i) \leq 0 \quad \text{for all } k \geq k^* \quad \text{for all } i \neq j \quad \text{for all } j = 1, 2, \cdots, R$$

and

$$U(k) = U(k^*) \qquad \text{for} \quad k \geq k^*$$
$$B(k) = B(k^*) \qquad \text{for} \quad k \geq k^*.$$

That is, the occurrence of a matrix $Y(k)$ with all nonpositive elements of $Y(k)(e_j - e_i)$ for all $i \neq j$ and all $j$ at any step terminates the algorithm and indicates the nonlinear separability of the $R$-class patterns.

*Proof:* Making substitution of the recurrence relation of $Y(k)$ in (71) and simplification, it can be shown that

$$\Delta V[Y(k)] = \mathrm{Tr}\,[Y^t(k+1)Y(k+1) - Y^t(k)Y(k)]$$
$$= -2p(k) \sum_{j=1}^{R} \sum_{l=1}^{n_j} {}_lH_j(Y(k))\, {}_lY_j^t(k)$$
$$+ p^2(k) \sum_{q=1}^{R-1} h_q^t(Y(k))(I - AA^\#)h_q(Y(k)) \quad (73)$$

From (57), (50) and (67),

$${}_lS_j(Z) = {}_lZ_jR({}_lZ_j) \quad (74)$$

Substituting (74) into (60) gives

$${}_lH_j(Y(k)) = [{}_lZ_j(k)R({}_lZ_j(k)) + {}_l\Lambda_j(k)]E_j^{-1}. \quad (75)$$

Substitute (75) and (54) into the following expression,

$$-2p \sum_{j=1}^{R} \sum_{l=1}^{n_j} {}_lH_j\, {}_lY_j^t$$
$$= -p \sum_j \sum_l {}_lH_j(Y(k))(E_j^t)^{-1}R^{-1}({}_lZ_j)E_j^t\, {}_lH_j^t(Y(k))$$
$$- p \sum_j \sum_l [{}_lZ_jR({}_lZ_j) + {}_l\Lambda_j](E_j^tE_j)^{-1}R^{-1}({}_lZ_j) \cdot [{}_lZ_jR({}_lZ_j) - {}_l\Lambda_j]^t. \quad (76)$$

It has been shown that the off-diagonal elements in $(E_j^tE_j)^{-1}$ are negative,[16] and, from (67) and (68), $R^{-1}({}_lZ_j)$ is a diagonal matrix with all positive diagonal elements. It follows that the off diagonal elements of $(E_j^tE_j)^{-1}R^{-1}({}_lZ_j)$ are also negative. From (56), (60), and (74), the elements of $[{}_lZ_jR({}_lZ_j) + {}_l\Lambda_j]$ are either positive or zero, and the corresponding elements of $[{}_lZ_jR({}_lZ_j) - {}_l\Lambda_j]$ are either zero or negative. Hence, the last term in (76), which is equal to $-{}_l\epsilon_j$ as defined in (69), is shown to be non-positive. Substituting (69) into (76), which, in turn, is substituted into (73), one obtains

$$\Delta V[Y(k)] = -p(k) \sum_{j=1}^{R} \sum_{l=1}^{n_j} {}_lH_j(Y(k))\{(E_j^t)^{-1} \cdot [R^{-1}({}_lZ_j) - p(k)I]E_j^t\}\, {}_lH_j^t(Y(k))$$
$$- p(k) \sum_{j=1}^{R} \sum_{l=1}^{n_j} {}_l\epsilon_j(k)$$
$$- p^2(k) \sum_{q=1}^{R-1} h_q^t(Y(k))AA^\# h_q(Y(k)), \quad (77)$$

$\Delta V(Y(k))$ is negative definite if the right hand side of the above equation is negative definite in ${}_lH_j(Y(k))$ or in $[{}_lZ_jR({}_lZ_j) + {}_l\Lambda_j]$. The last two terms on the right hand side are negative semi-definite. If a value of $p(k)$ can be found such that

$$\sum_{j=1}^{R} \sum_{l=1}^{n_j} {}_lH_j(Y(k))\{(E_j^t)^{-1}[R^{-1}({}_lZ_j) - p(k)I]E_j^t\} \cdot {}_lH_j^t(Y(k)) > 0$$

then $\Delta V(Y(k))$ is negative definite in $[{}_lZ_jR({}_lZ_j) + {}_l\Lambda_j]$. Note that if

$$p(k) = \frac{1}{\cosh Y_{\max}(k)}, \qquad Y_{\max}(k) = \max_{j,l,q} | {}_lY_{jq}(k) |,$$

$[R^{-1}({}_lZ_j) - p(k)I]$ is positive definite and has real eigenvalues as can be shown by following (67) and (68); but it is not certain that $(E_j{}^t)^{-1}[R^{-1}({}_lZ_j) - p(k)I]E_j{}^t$ can be positive definite for all $j$ and all $l$. Let $p(k)$ be so chosen as to maximize $-\Delta V[Y(k)]$ at each step, one obtains a choice of $p(k)$ as given in (65), provided the condition (66) is satisfied to make sure that $p(k) > 0$. For this value of $p(k)$,

$$\Delta V(Y(k)) = -\frac{\left[\sum_{j=1}^{R}\sum_{l=1}^{n_j}\{{}_l\epsilon_j(k) + {}_lH_j(Y(k))(E_j{}^t)^{-1} \cdot R({}_lZ_j(k))E_j{}^t\ {}_lH_j(Y(k))\}\right]^2}{4\sum_{q=1}^{R-1} h^t{}_q(Y(k))(I - AA^\#)h_q(Y(k))}$$

$$\leq 0 \text{ for } {}_lH_j(Y(k)) \neq 0 \text{ or } [{}_lZ_jR({}_lZ_j) + {}_l\Lambda_j] \neq 0 \quad \text{for all } l \text{ and } j.$$

Hence, $\Delta V[Y(k)]$ is negative definite in $[{}_lZ_jR({}_lZ_j) + {}_l\Lambda_j]$. Note that ${}_lZ_jR({}_lZ_j) + {}_l\Lambda_j = 0$ for all $j$ and all $l$ only if ${}_lZ_j \leq 0$, that is, only if $Y(k) = 0$ or ${}_iY_j(k) \cdot (e_j - e_i) \leq 0$ for all $i \neq j$ and for all $j$. Since it is assumed that the set of the inequalities (38) is consistent, from the lemma $Y_j(k)(e_j - e_i) \nleq 0$ for all $i \neq j$ and for all $j$; therefore,

$$\Delta V[Y(k)] < 0 \qquad \text{for all} \quad Y(k) \neq 0$$
$$= 0 \qquad \text{if} \quad Y(k) = 0$$

and the solution $Y = 0$ of the equation (71) can be reached asymptotically, that is,

$$\lim_{k\to\infty} \| Y(k) \|^2 = 0$$

which corresponds to a solution $U^{**}$ with $AU^{**} = B$ such that $A_jU^{**}(e_j - e_i) = B_j(e_j - e_i) > 0$ for all $i \neq j$ and for all $j$. This is the proof of Part 1(a).

Note that for $B(0)$ given in (70), and $\delta > 0$,

$${}_lb_j{}^t(k+1)E_j = {}_lb_j{}^t(k)E_j + p(k)[{}_lS_j(Z(k)) + {}_l\Lambda_j(k)]$$
$$> \beta(1+\delta)e_j{}^tE_j > 0 \qquad \text{for all} \quad l \quad \text{and} \quad j.$$

For a sufficiently large but finite $k$, $V[Y(k)] < 1$ such that $\| {}_lY_j(k) \|^2 < 1$ and

$${}_lY_j(k)E_j > -\beta e_j{}^tE_j \qquad \text{for all} \quad l \neq j \qquad \text{and all} \quad j.$$

It follows then

$$A_jU(k)E_j = B_j(k)E_j + Y_j(k)E_j$$
$$> (1+\delta)B_j(0)E_j - B_j(0)E_j > \delta B_j(0)E_j > 0 \qquad \text{for all} \quad j$$

which indicates a solution $U^* = U(k)$ is obtained in a finite number of iteration steps. This is the proof of part 1(b).

Part 2 can be proved in the same way as that in the Ho-Kashyap theorem.[17]

## CONCLUSION

A new generalized inverse algorithm for $R$-class pattern classification is proposed which is parallel to the one given by Teng and Li. In the case of $R = 2$, the algorithm is reduced to the improved dichotomization algorithm developed in the beginning; except here $A_2$ is composed of transposes of augmented pattern vectors without change of sign and $B_2$ is a column vector consisting of elements all equal to $e_2 = -1$. This corresponds to the reformulation of the Ho-Kashyap algorithm as mentioned by Wee and Fu.[15] The proposed 2-class algorithm has a higher rate of convergence than previous methods for a certain range of initial $b$ vector or vectors. A comparison has been made between this improved algorithm with $p(k)$ given by equation (26) and the Ho-Kashyap algorithm with $p = 1$, the convergence rate may be greatly increased for $.001 \leq b_i(0) \leq 0.5$ ($i = 1, 2, \cdots, N$), as verified by the computer results of several switching theory and pattern classification problems. For problems where a large number of iterations, for example, greater than twenty, were required for the Ho-Kashyap algorithm, the proposed algorithm reduced this number of iterations by a factor of 20 or more. Even though the cost per iteration for the proposed algorithm is 10 to 20 per cent greater than the Ho-Kashyap algorithm, the total cost is reduced. For problems where a small number of iterations were required by the Ho-Kashyap algorithm, less than twenty, the proposed algorithm reduced the number of iterations by as much as 30 percent. Experimental results suggest that the proposed algorithm is advantageous for problems requiring a large number of iterations by the Ho-Kashyap algorithm.

## ACKNOWLEDGMENT

This work is based partially on a Ph.D. dissertation submitted by the first author in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the University of Pittsburgh. During the course of this study, the first author was supported by a NASA Traineeship as well as the Learning Research and Development Center at the University. This work was also supported in part by NASA Grant NsG-416. The authors would like to thank Professor J. G. Castle, Professor T. W. Sze, Dr. T. L. Teng and Major T. E. Brand for their helpful discussions.

## REFERENCES


1 N J NILSSON
*Learning machines*
McGraw-Hill Inc New York N Y 1965

2 G S SEBESTYEN
*Decision-making processes in pattern recognition*
MacMillan Co New York N Y 1962

3 A G ARKADEV E M BRAVERMAN
*Computers and pattern recognition*
Thompson Book Co Washington D C 1967

4 L UHR
*Pattern recognition*
John Wiley and Sons Inc New York N Y 1966

5 C K CHOW
*An optimum character recognition system using decision functions*
IRE Transactions on Electronic Computers Vol EC-6 No 4 1957

6 R O WINDER
*Threshold logic*
Ph D Dissertation Princeton University Princeton N J 1962

7 P M LEWIS C L COATES
*Threshold logic*
John Wiley ane Sons New York N Y 1967

8 Y C HO R L KASHYAP
*An algorithm for linear inequalitites and its applications*
IEEE Transactions on Electronic Computers Vol EC-14 No 5 1965

9 R L KASHYAP
*Pattern classification and switching theory*
Ph D dissertation Harvard University Cambridge Mass 1965

10 W G CHAPLIN V S LEVADI
*A generalization of the linear threshold decision algorithm to multiple classes*
Computer and Information Sciences-II edited by J T Tou
Academic Press New York N Y 1967

11 C C BLAYDON
*Recursive algorithms for pattern recognition*
Ph D dissertation Harvard University Cambridge Mass 1967

12 K S FU W G WEE
*On generalizations of adaptive algorithms and application of the fuzzy sets concept to pattern classification*
TR-EE67-7 School of Electrical Engineering Purdue University Lafayette Ind 1967

13 I P DEVYATERIKOV A I PROPOI Y Z TSYPKIN
*Iterative learning algorithms for pattern recognition*
Automation and Remote Control Vol 28 No 1 1967

14 W G WEE
*Generalized inverse approach to adaptive multiclass pattern classification*
IEEE Transactions on Computers Vol C-17 No 12 1968

15 W G WEE K S FU
*An extension of the generalized inverse algorithm to multiclass pattern classification*
IEEE Transactions on Systems Science and Cybernetics Vol SSC-4 No 2 1968

16 T L TENG C C LI
*On a generalization of the Ho-Kashyap algorithm to multiclass pattern classification*
Proceedings of the Third Annual Princeton Conference on information Sciences and Systems 1969

17 L C GEARY
*An improved algorithm for linear inequalities in pattern recognition and switching theory*
Ph D Dissertation University of Pittsburgh Pittsburgh Pa 1968

18 A G NIELSEN A H VAGNUCCI C C LI
*Application of pattern recognition to electrolyte circadian cycles*
Proceedings of the 8th International Conference on Medical and Biological Engineering Chicago Ill 1969

Proceedings of the Third Hawaii International Conference
On System Sciences, Part 1, 1970

# ON AN IMPROVED GENERALIZED INVERSE ALGORITHM FOR LINEAR INEQUALITIES

by
L. C. Geary* and C. C. Li**
Department of Electrical Engineering, University of Pittsburgh
Pittsburgh, Pennsylvania 15213

INTRODUCTION. Both pattern recognition and threshold logic problems require the determination of a decision function or decision functions which, in the case of linear separation, involve a set of linear inequalities(1-4). An improved generalized inverse algorithm has been developed for a solution $\underline{w}$ of a set of linear inequalities $\underline{A}\ \underline{w} > \underline{0}$. The algorithm is an improvement of the Ho-Kashyap algorithm by choosing a different criterion function,

$$J(\underline{y}) = 4 \sum_{i=1}^{N} (\cosh \frac{1}{2} y_i)^2, \tag{1}$$

to be minimized where $y_i$ is the ith component of the Nx1 vector $\underline{y} = \underline{A}\ \underline{w} - \underline{b}$, $\underline{b} > \underline{0}$. The improvement lies in the acceleration of the Ho-Kashyap algorithm caused by a steeper gradient of $J(\underline{y})$.

THE PROPOSED TWO-CLASS ALGORITHM. Let the matrix $\underline{A}$, whose transpose is $\underline{A}^t = [{}_1\underline{x}_1, \ldots, {}_{n_1}\underline{x}_1, -{}_1\underline{x}_2, \ldots, -{}_{n_2}\underline{x}_2]$, be an Nxn matrix of augmented sample pattern vectors ${}_\ell\underline{x}_i$ of dimension nx1 where the subscript on the right denotes the pattern class and the subscript on the left denotes the $\ell$th sample pattern in that class. Note that $N = n_1 + n_2$. The gradients of $J(\underline{y})$ with respect to $\underline{w}$ and $\underline{b}$ are

$$\frac{\partial J(\underline{y})}{\partial \underline{w}} = 2\ \underline{A}^t\ \underline{s}(\underline{y}), \qquad \frac{\partial J(\underline{y})}{\partial \underline{b}} = 2\ \underline{s}(\underline{y}) \tag{2}$$

where $\underline{s}^t(\underline{y}) = [\sinh y_1, \ldots, \sinh y_N]$. Since $\underline{w}$ is not constrained in any way, $\frac{\partial J(\underline{y})}{\partial \underline{w}} = \underline{0}$ implies $\underline{s}(\underline{y}) = \underline{0}$, which, in turn, implies $y_i = 0$ for all $i=1,2,\ldots,N$. Solving $\underline{y} = \underline{A}\ \underline{w} - \underline{b} = \underline{0}$, one obtains $\underline{w} = \underline{A}^{\#}\underline{b}$, where $\underline{A}^{\#}$ is the generalized inverse of $\underline{A}$. On the other hand, for a fixed $\underline{w}$, $\frac{\partial J(\underline{y})}{\partial \underline{b}} = \underline{0}$ with $\underline{b} > \underline{0}$ dictates a descent procedure of the following form, with k denoting the iteration number: $\underline{b}(k+1) = \underline{b}(k) + p(k)\ \underline{h}(k)$, where

---

*Presently at Gulf Research Center, Pittsburgh, Pennsylvania 15230.

**The work was supported in part by the NASA Grant NsG-416.

$$\underline{h}(k) = [h_i(k)] = [\sinh y_i(k) + |\sinh y_i(k)|], \quad (i=1,2,\ldots,N). \tag{3}$$

It can be shown that p(k) may be chosen to equal

$$p(k) = \frac{1}{\cosh y_{max}(k)} \tag{4}$$

where $y_{max}(k) = \underset{i}{\text{Max}} |y_i(k)|$. An optimum value of p(k) is

$$p(k) = \frac{[\underline{y}(k) + |\underline{y}(k)|]^t \underline{R}(k) \; [\underline{y}(k) + |\underline{y}(k)|]}{2[\underline{y}(k) + |\underline{y}(k)|]^t \underline{R}(k) \; [\underline{I} - \underline{A}\,\underline{A}^{\#}] \underline{R}(k) [\underline{y}(k) + |\underline{y}(k)|]} \tag{5}$$

provided that $\underline{I} - \underline{A}\,\underline{A}^{\#} > 0$.

Combining the above relationship one obtains the following algorithm:

$$\begin{aligned} &\underline{w}(0) = \underline{A}^{\#}\underline{b}(0), \; \underline{b}(0) > \underline{0}; \qquad \underline{y}(k) = \underline{A}\,\underline{w}(k) - \underline{b}(k); \\ &\underline{b}(k+1) = b(k) + p(k)\,\underline{h}(k); \; \underline{w}(k+1) = \underline{w}(k) + p(k)\,\underline{A}^{\#}\underline{h}(k). \end{aligned} \tag{6}$$

Note that in this algorithm p(k) varies at each step and is a non-linear function of $\underline{y}(k)$. Just like the Ho-Kashyap algorithm, it can be shown that the above algorithm (6) converges to a solution $\underline{w}^*$ of the system of linear inequalities in a finite number of steps provided that a solution exists, and simultaneously acts as a test for the inconsistency of the linear inequalities(6).

THE PROPOSED MULTICLASS ALGORITHM. The problem of multiclass pattern classification is that it must be determined to which of the R different classes, $C_1, C_2, \ldots, C_R$, a given pattern vector, $\underline{x}$, belongs. If the R-class patterns are linearly separable, there exist R weight vectors $\underline{w}_j$ to construct R discriminant functions $g_j(\underline{x})$, (j=1,2,...,R), such that $g_j(\underline{x}) = \underline{x}^t\underline{w}_j > \underline{x}^t\underline{w}_i = g_i(\underline{x})$ for all $i \neq j$, $\underline{x} \in C_j$. Chaplin and Levadi have formulated another set of inequalities which can be considered as a representation of linear separation of R-class patterns(5). This set of inequalities is $||\underline{x}^t\underline{U} - \underline{e}_j^t|| < ||\underline{x}^t\underline{U} - \underline{e}_i^t||$ for all $i \neq j$, $\underline{x} \in C_j$ and all j=1,2,...,R, where $\underline{U}$ is an nx(R-1) weight matrix and the vectors $\underline{e}_j$'s are the vertex vectors of a R-1 dimensional equilateral simplex with its centroid at the origin. If each $\underline{e}_j$ is associated with one class, $\underline{x}$ is classified according to the nearest neighborhood of the mapping $\underline{x}^t\underline{U}$ to the vertices. These two representations are, in fact, equivalent with $\underline{w}_j = \underline{U}\ \underline{e}_j$, $(j=1,2,\ldots,R)$.

The generalization of the improved two-category algorithm applicable to multi-class pattern classification problems has been developed and its convergence proved[6]. The algorithm solves for an $n x (R-1)$ solution matrix $\underline{U}$ of a set of linear inequalities $\underline{A}_j \underline{U}\ (\underline{e}_j - \underline{e}_i) > \underline{0}$, (for all $i \neq j$ and $j=1,2,..,R$), which in turn generates the weight vectors $\underline{w}_j = \underline{U}\ \underline{e}_j$, where $\underline{A}_j$ is the jth block in $\underline{A}$ composed of $n_j$ training augmented pattern vectors of class $C_j$. Let $\underline{B}$ be an $Nx(R-1)$ matrix whose row vectors ${}_\ell\underline{b}_j^t$ correspond to the class grouping in $\underline{A}$ and satisfy the following inequalities ${}_\ell\underline{b}_j^t\ (\underline{e}_j - \underline{e}_i) > 0$ for all $i \neq j$, and $j=1,2,\ldots,R$. Let also $\underline{Y}$ be an $Nx(r-1)$ matrix composed of row vectors, ${}_\ell\underline{Y}_j$. Then the generalized multi-class algorithm is given in the following equations:

$$\underline{U}(0) = \underline{A}^{\#}\underline{B}(0),\ \underline{Y}(k) = \underline{A}\ \underline{U}(k) - \underline{B}(k),\ \underline{Z}_j(k) = \underline{Y}_j(k)\ \underline{E}_j,$$
$$\underline{B}(k+1) = \underline{B}(k) + p(k)\underline{H}[\underline{Y}(k)],\ \underline{H}_j[\underline{Y}(k)] = [\underline{S}_j(\underline{Z}(k)) + \underline{\Lambda}_j(k)]\underline{E}_j^{-1},$$
$$\underline{U}(k+1) = \underline{U}(k) + p(k)\ \underline{A}^{\#}\underline{H}[\underline{Y}(k)], \tag{7}$$

where

$$\underline{E}_j = [\underline{e}_j - \underline{e}_1,\ldots,\underline{e}_j - \underline{e}_{j-1},\ \underline{e}_j - \underline{e}_{j+1},\ldots,\ \underline{e}_j - \underline{e}_R],$$
$$\underline{S}_j(\underline{Z}(k)) \triangleq [{}_\ell S_{jq}(\underline{Z}(k))] = [\sinh\ {}_\ell Z_{jq}(k)],\quad (\ell=1,\ldots,n_j)$$
$$\underline{\Lambda}_j(k) = [{}_\ell\Lambda_{jq}(k)],\ {}_\ell\Lambda_{jq}(k) \triangleq {}_\ell S_{jq}(\underline{Z}(k))\ \mathrm{Sgn}\ ({}_\ell Z_{jq}(k)),$$

and $\underline{B}_j(0)$ = the jth block in $\underline{B}(0)$ composed of $n_j$ row vectors $\underline{e}_j^t$. $p(k)$ can be expressed by

$$p(k) = \frac{\sum_{j=1}^{R}\sum_{\ell=1}^{n_j} \{{}_\ell\varepsilon_j(k) + {}_\ell\underline{H}_j(\underline{Y}(k))(\underline{E}_j^{\ t})^{-1}R^{-1}({}_\ell\underline{Z}_j(k))\underline{E}_j^{\ t}\ {}_\ell\underline{H}_j(\underline{Y}(k))\}}{2\sum_{q=1}^{R-1} \underline{h}_q^{\ t}\ (\underline{I} - \underline{A}\ \underline{A}^{\#})\ \underline{h}_q}, \tag{8}$$

where

$${}_\ell\varepsilon_j(k) \triangleq [{}_\ell\underline{Z}_j(k)\ \underline{R}({}_\ell\underline{Z}_j(k)) + {}_\ell\underline{\Lambda}_j(k)](\underline{E}_j^{\ t}\underline{E}_j)^{-1}\underline{R}^{-1}({}_\ell\underline{Z}_j(k))$$
$$\cdot\ [{}_\ell\underline{Z}_j(k)\underline{R}({}_\ell\underline{Z}_j(k)) - {}_\ell\underline{\Lambda}_j(k)]^t,$$
$$\underline{R}({}_\ell\underline{Z}_j(k)) \triangleq \text{a diagonal matrix } [r_{qq}({}_\ell\underline{Z}_j(k))],$$

and

$$r_{qq}({}_\ell\underline{Z}_j(k)) = \frac{\mathrm{Sinh}\ {}_\ell Z_{jq}}{{}_\ell Z_{jq}},\quad (q=1,2,\ldots,R-1).$$

The proof of convergence of this multiclass algorithm utilizes the concept of mapping the pattern classes into vertices of the equilateral simplex[6], similar to the procedure used by Teng and Li[7]. The criterion function $J(\underline{Y})$ to be minimized with respect to $\underline{U}$ and $\underline{B}$ is the trace of $4\ \underline{C}^t(\underline{Y})\underline{C}(\underline{Y})$, $J(\underline{Y}) \triangleq \mathrm{Tr}\ (4\ \underline{C}^t\underline{C}) = \sum_{i=1}^{N} \sum_{j=1}^{R-1} J_{ij}(\underline{Y})$, where $J_{ij}(\underline{Y}) = 4\ (\cosh \frac{1}{2} y_{ij})^2$, $\underline{C}(\underline{Y})$ is an Nx(R-1) matrix defined by $\underline{C}(\underline{Y}) = [c_{ij}] \triangleq [\cosh \frac{1}{2} y_{ij}]$, and $y_{ij} = \sum_{k=1}^{n} a_{ij}\ u_{kj} - b_{ij}$, $(i=1,\ldots,N;\ j=1,\ldots,R-1)$.

CONCLUSION. Experimental results have shown that the rate of convergence and the computer time can be substantially reduced with the proposed algorithm when compared to the Ho-Kashyap algorithm, especially when the later required a large number of iterations.

REFERENCES


1. Nilsson, N. J., Learning Machines, McGraw-Hill, Inc., New York, 1965.
2. Ho, Y. C. and R. L. Kashyap, "An Algorithm for Linear Inequalities and its Applications", IEEE Trans. on Electronic Computers, Vol. EC-14, pp. 683-688, 1965.
3. Lewis,P. M.II. and C. L. Coates, Threshold Logic, John Wiley & Sons., Inc., New York, 1967.
4. Fu, K. S. and W. G. Wee, "On Generalizations of Adaptive Algorithms and Application of the Fuzzy Sets Concept to Pattern Classification", TR-EE67-7, School of Electrical Engineering, Purdue University, June 1967.
5. Chaplin, W. G. and V. S. Levadi, "A Generalization of the Linear Threshold Decision Algorithm to Multiple Classes", in Computer and Information Sciences-II, edited by J. T. Tou, Academic Press, New York, 1967, pp. 337-355.
6. Geary, L. C., "An Improved Algorithm for Linear Inequalities in Pattern Recognition and Switching Theory," Ph.D. Dissertation, University of Pittsburgh, 1968.
7. Teng, T. L. and C. C. Li, "On A Generalization of the Ho-Kashyap Algorithm to Multi-Class Pattern Classification", Proc. Third Annual Princeton Conference on Information Sciences and Systems, pp. 209-213, 1969.