Pattern Recognition, Vol 23, No 3/4, pp 325–335, 1990
Printed in Great Britain

0031-3203/90 $3.00 + .00
Pergamon Press plc

GENERALIZED GUARD-ZONE ALGORITHM (GGA) FOR LEARNING: AUTOMATIC SELECTION OF THRESHOLD

A. PAL (PATHAK) and S. K. PAL

Electronics and Communication Sciences Unit, Indian Statistical Institute, 203 B T Road, Calcutta 700035, India

(*Received* 27 *October* 1988; *in revised form* 7 *April* 1989; *received for publication* 16 *May* 1989)

Abstract—This work is a continuation of our earlier work on the Generalized Guard-zones Algorithm (GGA) for self-supervised parameter learning. An attempt is made here for the automatic determination of the guard-zone parameter λ_n (i.e. the threshold used for discarding doubtful or mislabeled samples) at every instant of learning, for the general m-class N-feature pattern recognition problem. This is done by minimizing the mean squared error (MSE) of the estimate, under a simple probabilistic model which takes into consideration the presence of mislabeled training samples. Under the assumptions of normality, it is found that the estimates for λ_n so obtained are distribution-free, that is, they do not depend on the parameters of the distribution. They are functions of N, the iteration number n and certain percentage points of the beta distribution with parameters N and $n - N$. The effectiveness of the automatic selection of guard-zone dimension is further demonstrated on a bivariate three-class data set to show the improvement in performance of the GGA.

GGA Learning Optimum dimension/threshold

1. INTRODUCTION

A Generalized Guard-zone Algorithm (GGA) was described by Pathak and Pal[1] for learning class parameters using a restricted updating program, together with investigation of its stochastic convergence for optimum learning. Basically, the aim of the GGA is to detect mislabeled training samples and outliers and to reject them from the parameter updating procedure. The algorithm is a generalization of some existing ones[2,3] which were found to be useful for practical data but without mathematical formulation of their various features (e.g. convergence, optimum dimension for guard zone/threshold, performance in presence of mislabeling, etc.)

Recently, it was reported[4] that the guard zone parameter (λ_n) of GGA lies between certain bounds and the recognition rate increases when the guard zone is made "dynamic" by making its zone-controlling parameter dependent on current estimates. It should be mentioned here that the zone-controlling parameter was kept constant in the algorithms/experiments of Pal *et al.*[2] and Chien.[3]

However, the problem of automatic selection of the guard-zone parameter λ_n was not greatly facilitated by the above study, and continued to be an impediment in the practical implementation of the GGA. It became necessary therefore, to tackle this problem from a different viewpoint, that is, by using criteria other than stochastic convergence. This led to the present work, which attempts to determine the values of guard-zone dimension at every instant of learning automatically based on mean squared error (MSE). The explicit expressions for the MSE are obtained for both the GGA and the non-GGA (i.e. the usual unsupervised stochastic approximation learning algorithm not based on guard-zone) using the model of Chittineni,[6] involving mislabeled training samples. An approximation for the guard-zone parameter λ_n was obtained for which the MSE for the GGA is smaller than that for the non-GGA. In other words, the value of λ_n selected automatically by the system makes the GGA discard the doubtful (mislabeled) samples from the training procedure, thus improving its performance *vis-à-vis* the non-GGA for self-supervised learning.

This feature is further exemplified with the help of a two-feature three-class normally distributed data set.

2. THE GENERALIZED GUARD-ZONE ALGORITHM (GGA)[1]

Let us consider a general m-class pattern recognition problem, where C_i, $i = 1, \ldots, m$, denotes the ith class. For this purpose, let the feature vector selected be

$$\mathbf{X} = [x_1, x_2, \ldots, x_N]', \qquad \mathbf{X} \in \mathbb{R}^N.$$

Let us assume that:

(A1) The probability densities $P_k(\mathbf{X})$ of $\mathbf{X}$ for the classes C_k, $k = 1, \ldots, m$, are continuous and of the same parametric family.

(A2) The densities P_k (.) involve a q-dimensional

parameter vector θ_k, which needs to be learned, either wholly or partly.

(A3) The densities $p_k(.)$ admit of moments of the first two orders, i.e.

$$E(\mathbf{X}|C_k) = \mu_k \quad \text{and}$$
$$\text{Disp}(\mathbf{X}|C_k) = \Sigma_k$$

exist.

(A4) An unbiased statistic exists for the parameter vector θ.

Let us suppose that for the purpose of learning θ_k, a set of samples

$$\{\mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \ldots, \mathbf{X}_{nk}^{(k)}\}$$

is provided, for $k = 1, \ldots, m$, where the superscripts k denote the labels "given" to the respective samples, as opposed to their true labels.

We assume that:

(A5) The training samples for any given class are all statistically independent.

The GGA for estimating (θ_k) recursively is as follows:

$$\hat{\theta}_n^{(k)} = \mathbf{f}(\mathbf{X}_n^{(k)}) \text{ for } n = 1$$
$$\hat{\theta}_{n-1}^{(k)} - a_n \mathbf{Y}_n^{(k)} \text{ for } n > 1, \tag{1a}$$

where

$$\mathbf{Y}_n^{(k)} = \hat{\theta}_{n-1}^{(k)} - \mathbf{f}(\mathbf{X}_n^{(k)}) \text{ if } \mathbf{X}_n^{(k)} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n) \tag{1b}$$

$\hat{\theta}_n^{(k)}$ is the nth-stage estimate of θ_k,
$\{a_n\}$ is a sequence of positive numbers,
$\mathbf{f}: \mathbb{R}^N \to \mathbb{R}^q$ is a continuous map, defining an unbiased statistic for θ_k,
$\hat{\mu}_{n-1}^{(k)}$ is the $(n-1)$th-stage GGA estimate of μ_k,
$G(\hat{\mu}_{n-1}^{(k)}, \lambda_n) = \{\mathbf{X}: \mathbf{X} \in \mathbb{R}^N, d(\mathbf{X}, \hat{\mu}_{n-1}^{(k)}) \leq \lambda_n\}$,
$d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})' \mathbf{B}_n (\mathbf{x} - \mathbf{y})$,
$\mathbf{B}_n$ is a symmetric positive definite matrix, which may or may not be a function of the training samples $\mathbf{X}_n^{(k)}$ (some examples are given in Refs (1) and (5)),
λ_n is a positive number, suitably chosen.

Incidentally, $G(\mathbf{a}, r)$ is the guard-zone and, clearly, is nothing but a closed ball centred at $\mathbf{a}$ and having radius r. In essence, this algorithm allows only those training samples to be used for the updating program which lie within the corresponding guard-zone centred at the preceding estimate of the mean. Training samples which lie outside it are ignored, and at the corresponding stages the estimate is kept unchanged.

The choice of the various parameters of the algorithm, namely $\{a_n\}$, λ_n and $\mathbf{B}_n$, is governed by a number of factors, which depend on the criterion of performance chosen. This point was discussed to some extent by Pal and Pal.[5]

3. MODELLING MISLABELED TRAINING SAMPLES

A very simple but realistic model, inspired by Chittineni,[6] is adopted to describe the situation in which there may be mislabeled training samples. Let w and $\hat{w}$ denote, respectively, the true and the given labels. Clearly,

$$w, \hat{w} \in \{1, 2, \ldots, m\} = \Omega_c, \text{ say.}$$

Let $\pi_k = P(w = k)$ denote the *a priori* probability for the class C_k, $k = 1, \ldots, m$. Further, let $p_k(\mathbf{X}) = p(\mathbf{X}|w = k)$ be the class-conditional probability density of the feature vector $\mathbf{X}$. Also, let α_{kj} denote the probability that a sample from C_j is given the label k, i.e.

$$\alpha_{kj} = P(\hat{w} = k | w = j), \quad \text{for } j, k = 1, \ldots, m. \tag{2}$$

Clearly,

$$\sum_{k=1}^{m} \alpha_{kj} = 1.$$

Under this model, it can be shown that for any subset $A_k(n)$ of the sample space, the probability density of a sample labeled k is

$$p(\mathbf{X}_n^{(k)}) = p(\mathbf{X}_n|\hat{w} = k) = p(\mathbf{X}|\hat{w} = k)$$
$$= \sum_{j=1}^{m} \beta_{kj}(n) p(\mathbf{X}|w = j) \text{ if } \mathbf{X}_n^{(k)} \in A_k(n), \text{ given } \hat{w} = k \tag{3a}$$
$$= \sum_{j=1}^{m} \beta_{kj}^*(n) p(\mathbf{X}|w = j), \text{ otherwise,} \tag{3b}$$

where $A_k(n) = \{\mathbf{x}: \mathbf{x} \in G(\hat{\mu}_{n-1}^{(k)}, \lambda_n)\}$,

$$\beta_{kj}(n) = P(A_k(n)|\mathbf{X}, \hat{w} = k, w = j)\alpha_{kj}\pi_j / P(\hat{w} = k) \tag{3c}$$

$$\beta_{kj}^*(n) = P(A_k^c(n)|\mathbf{X}, \hat{w} = k, w = j)\alpha_{kj}\pi_j / P(\hat{w} = k), \tag{3d}$$

provided we are prepared to assume that:

(A6) $p(\mathbf{X}|\hat{w} = k, w = j) = p(\mathbf{X}|w = j)$ for all $j, k = 1.2, \ldots, m$

(A7) $P(\hat{w} = k, A_k(n)) > 0$ for all k, n

(A8) $P(\hat{w} = k, A_k^c(n)) > 0$ for all k, n.

A proof can be found in Ref. (5).

It is not difficult to observe that $\beta_{kj}(n)$, $\beta_{kj}^*(n) \in [0, 1]$ for all $k, j = 1.2, \ldots, m$, as it is known that

$$P(\hat{w} = k) = \sum_{j=1}^{m} P(w = k, w = j)$$
$$= \sum_j \pi_j \alpha_{kj}.$$

4. PERFORMANCE OF THE GGA RELATIVE TO THAT OF THE NON-GGA

Before a comparison of the performances of the two algorithms can be made, it is necessary to introduce a suitable measure of the quality of learning. Ideally, such a function should estimate, at each instant n, the distance between the current state $\hat{\theta}_n$ and the optimal state θ. One convenient performance index of learning is the MSE of the estimate at each instant, namely,

$$D(\hat{\theta}_n) = E\|\hat{\theta}_n - \theta\|^2 \tag{4}$$

for discrete algorithms of learning.

In the following sections, we shall drop the suffix k denoting the class, for convenience, unless it is required for the sake of clarity.

Performance index of the GGA

The GGA is defined as

$$\text{for } n = 1,\ \hat{\boldsymbol{\theta}}_n = \mathbf{f}(\mathbf{X}_1) \tag{5a}$$

and for $n > 1$, $\hat{\boldsymbol{\theta}}_n =$

$$\begin{cases} (1 - a_n)\hat{\boldsymbol{\theta}}_{n-1} + a_n\mathbf{f}(\mathbf{X}_n) \text{ with probability } p_n & \text{(5b)} \\ \hat{\boldsymbol{\theta}}_{n-1} \text{ with probability } q_n = 1 - p_n, & \text{(5c)} \end{cases}$$

where all symbols are as in Section 3 and

$$\begin{aligned} p_n &= P(\mathbf{X}_n \in G(\hat{\boldsymbol{\mu}}_{n-1}^{(k)}, \lambda_n)) \\ &= P(A_k(n)), \text{ say.} \end{aligned}$$

Let $\mathbf{P}_n = \hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}$.

Then $\mathbf{P}_1 = \mathbf{f}(\mathbf{X}_1) - \boldsymbol{\theta}$, (6a)

and, for $n > 1$,

$$\mathbf{P}_n = \begin{cases} \bar{a}_n\mathbf{P}_{n-1} + a_n(\mathbf{f}(\mathbf{X}_n) - \boldsymbol{\theta}) \text{ with probability } p_n & \text{(6b)} \\ \mathbf{P}_{n-1} \text{ with probability } q_n, & \text{(6c)} \end{cases}$$

where $\bar{a}_n = 1 - a_n$.

Thus, $E(\hat{\boldsymbol{\theta}}_n) = p_nE(\bar{a}_n\hat{\boldsymbol{\theta}}_{n-1} + a_n\mathbf{f}(\mathbf{X}_n)) + q_nE(\hat{\boldsymbol{\theta}}_{n-1})$

$$\begin{aligned} &= (\bar{a}_np_n + q_n)E(\hat{\boldsymbol{\theta}}_{n-1}) + a_np_nE(\mathbf{f}(\mathbf{X}_n)). \\ &= (\bar{a}_np_n + q_n)[(\bar{a}_{n-1}p_{n-1} + q_{n-1})E(\hat{\boldsymbol{\theta}}_{n-2}) \\ &\quad + a_{n-1}p_{n-1}E(\mathbf{f}(\mathbf{X}_{n-1}))] + a_np_nE(\mathbf{f}(\mathbf{X}_n)) \\ &= \ldots \\ &= \prod_{i=2}^{n}(\bar{a}_ip_i + q_i)E(\hat{\boldsymbol{\theta}}_1) \\ &\quad + \sum_{j=2}^{n}\prod_{i=j+1}^{n}(\bar{a}_ip_i + q_i)a_jp_jE(\mathbf{f}(\mathbf{X}_j)) \end{aligned} \tag{7a}$$

if we follow the convention that

$$\prod_{i=m}^{n}(\bar{a}_ip_i + q_i) = 1 \;\forall\, m > n.$$

Writing

$$A_{i,j} = \prod_{k=i}^{j}(\bar{a}_kp_k + q_k) = \prod_{k=i}^{j}(1 - a_kp_k),$$

we have, from equation (7a),

$$E(\hat{\boldsymbol{\theta}}_n) = A_{2,n}E(\hat{\boldsymbol{\theta}}_1) + \sum_{j=2}^{n}A_{j+1,n}a_jp_jE(\mathbf{f}(\mathbf{X}_j)), \tag{7b}$$

Similarly, if we define

$$Z_n = \mathbf{P}_n'\mathbf{P}_n = \|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}\|^2,$$

then

$$Z_1 = \|\mathbf{f}(\mathbf{X}_1) - \boldsymbol{\theta}\|^2 \tag{8a}$$

and, for $n > 1$,

$$\begin{aligned} Z_n &= \bar{a}_n^2Z_{n-1} + T_n \text{ with probability } p_n & \text{(8b)} \\ &= Z_{n-1} \text{ with probability } q_n, & \text{(8c)} \end{aligned}$$

where $T_n = a_n^2\mathbf{Q}_n'\mathbf{Q}_n + 2a_n\bar{a}_n\mathbf{P}_{n-1}'\mathbf{Q}_n$,

$\mathbf{Q}_n = \mathbf{f}(\mathbf{X}_n) - \boldsymbol{\theta}$.

Thus $Z_1 = \mathbf{Q}_1'\mathbf{Q}_1$, and

$$E(Z_n) = B_{2,n}E(Z_1) + \sum_{j=2}^{n}B_{j+1,n}p_jE(T_j),$$

where $B_{i,j} = \prod_{k=i}^{j}(\bar{a}_k^2p_k + q_k)$.

As $E(T_j) = a_j^2E(\mathbf{Q}_j'\mathbf{Q}_j) + 2a_j\bar{a}_jE(\mathbf{P}_{j-1}'\mathbf{Q}_j)$

and $E(\mathbf{P}_{j-1}'\mathbf{Q}_j) = 0$,

on account of our assumption (A5) regarding the independence of the observations,

we have

$$E(T_j) = a_j^2E(\mathbf{Q}_j'\mathbf{Q}_j).$$

Thus

$$\begin{aligned} E(Z_n) &= B_{2,n}E(Z_1) \\ &\quad + \sum_{j=2}^{n}B_{j+1,n}a_j^2p_jE(\mathbf{Q}_j'\mathbf{Q}_j), \end{aligned} \tag{9a}$$

where $\bar{a}_j = 1 - a_j$.

We observe that equation (5a) is actually equivalent to equation (5b) with

$$\begin{aligned} a_1 &= 1 \\ p_1 &= 1. \end{aligned}$$

Equation (9) is therefore equivalent to

$$E(Z_n) = \sum_{j=1}^{n}B_{j+1,n}a_j^2p_jE_{jj}, \tag{9b}$$

where

$$\begin{aligned} E_{jj} &= E(\mathbf{Q}_j'\mathbf{Q}_j) \\ &= E[(\mathbf{f}(\mathbf{X}_j) - \boldsymbol{\theta}_k)'(\mathbf{f}(\mathbf{X}_j) - \boldsymbol{\theta}_k)|\hat{w} = K] \\ &= \sum_{i=1}^{m}\beta_{ki}(j)E[(\mathbf{f}(\mathbf{X}_j) - \boldsymbol{\theta}_k)'(\mathbf{f}(\mathbf{X}_j) - \boldsymbol{\theta}_k)|w = i] \\ &\quad + \sum_{i=1}^{m}\beta_{ki}^{*}(j)E[(\mathbf{f}(\mathbf{0}) - \boldsymbol{\theta}_k)'(\mathbf{f}(\mathbf{0}) - \boldsymbol{\theta}_k)|w = i] \end{aligned}$$

on aeeount of equation (3).

Let us now assume that

(A6) $r_j = E[\|\mathbf{f}(\mathbf{X}) - \boldsymbol{\theta}_j\|^2|w = j]$ exists,

so that

$$\begin{aligned} E_{jj} &= \sum_{i=1}^{m}\beta_{ki}(j)[r_i + \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_k\|^2] \\ &\quad + \sum_{i=1}^{m}\beta_{ki}^{*}(j)[\|f(\mathbf{0}) - \boldsymbol{\theta}_i\|^2 + \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_k\|^2]. \end{aligned} \tag{10}$$

Performance index of the non-GGA

The non-GGA is defined as[7]

$$\begin{aligned} \hat{\hat{\boldsymbol{\theta}}}_n &= \mathbf{f}(\mathbf{X}_1) \text{ for } n = 1 & \text{(11a)} \\ &= \bar{a}\hat{\hat{\boldsymbol{\theta}}}_{n-1} + a_n\mathbf{f}(\mathbf{X}_n) \text{ for } n > 1. & \text{(11b)} \end{aligned}$$

Clearly, proceeding as before,

$$E(\hat{\hat{\theta}}_n) = A^*_{2,n}E(\hat{\hat{\theta}}_1) + \sum_{i=1}^{m} a_j A^*_{j+1,n}E(\mathbf{f}(\mathbf{X}_j)), \quad (12)$$

where

$$A^*_{i,j} = \prod_{k=i}^{j} \bar{a}_k = \prod_{k=i}^{j} (i - a_k).$$

Writing

$$Z^*_n = \|\hat{\hat{\theta}}_n - \theta\|^2 \text{ if } n = 1$$
$$= \bar{a}_n{}^2 Z^*_{n-1} + T^*_n \text{ if } n > 1,$$

where

$$T^*_j = a_j^2 \mathbf{Q}'_j \mathbf{Q}_j + 2a_j a_j \mathbf{P}'_{j-1}\mathbf{Q}_j,$$
$$P^*_k = \hat{\hat{\theta}}_k - \theta$$

so that

$$E^*(T^*_j) = a_j{}^2 E^*(\mathbf{Q}_j) \text{ as } E^*(\mathbf{P}'_{j-1}\mathbf{Q}_j) = 0,$$

we have

$$E^*(Z^*_n) = B^*_{2,n}E^*(Z^*_1) + \sum_{j=2}^{n} B^*_{j+1,n}a_j{}^2 E^*(\mathbf{Q}'_j\mathbf{Q}_j),$$

where E denotes expectation under the probability model given by Pathak-Pal and Pal,[7] and

$$B^*_{i,j} = \prod_{k=i}^{j} \bar{a}_k{}^2.$$

Note. Equations (12) and (13) can also be obtained directly from equations (7) and (9) by specializing the value of p_j as 1 for all j. Clearly, this is because the non-GGA is a special case of the GGA for which $\lambda_n = \infty$.

Under assumption (A5) of independence of the training samples, and writing $a_1 = 1$ and $p_1 = 1$, we have

$$E^*(Z^*_n) = \sum_{j=1}^{n} a_j{}^2 B^*_{j+1,n}E^*(\mathbf{Q}'_j\mathbf{Q}_j) = \sum_{j=1}^{n} a_j{}^2 B^*_{j+1,n}E^*_{jj}, \quad (13)$$

where

$$E^*_{jj} = E^*(\mathbf{Q}'_j\mathbf{Q}_j) = \sum_{i=1}^{m} \varepsilon_{ki}[r_i + \|\theta_i + \theta_k\|^2], \quad (14a)$$

where

$$\varepsilon_{ki} = \alpha_{ki}\pi_i \Big/ \left(\sum_{j=1}^{m} \alpha_{kj}\pi_j\right).$$

As

$$\varepsilon_{ki} = \beta_{ki}(n) + \beta^*_{ki}(n) \quad \text{for all } n > 0,$$

we must have

$$E^*_{jj} = \sum_{i=1}^{m} [\beta_{ki}(j) + \beta^*_{ki}(j)](r_i + \|\theta_i + \theta_k\|^2) \quad (14b)$$

so that

$$E^*_{jj} - E_{jj} = \sum_{i=1}^{m} \beta_{ki}(j)[r_i - \|f(0) - \theta_k\|^2]. \quad (14c)$$

If we assume that

(A7) $f_i(X) > f_i(0)$ for all $i = 1, 2, \ldots, q$

then

$$E^*_{jj} > E_{jj}.$$

From equations (9) and (13) it follows that

$$E(Z_n) < E^*(Z^*_n)$$

if and only if

$$B_{j+1,n}p_j E_{jj} < B^*_{j+1,n}E^*_{jj} \text{ for all } j = 1, 2, \ldots, n. \quad (15)$$

Let us examine this set of necessary and sufficient conditions closely.

First of all, we note that, as

$$0 \le \bar{a}_j{}^2 \le \bar{a}_j{}^2 p_j + q_j \le 1, \text{ whatever } j,$$

we must have, for all i, j,

$$B^*_{i,j} \le B_{i,j}. \quad (16)$$

Also, it is sufficient to consider the case where $E_{jj} > 0$, as condition (15) is always trivially true when $E_{jj} = 0$.

Rewriting the inequality (15) as

$$p_j \le \frac{B^*_{j+1,n}}{B_{j+1,n}} \cdot \frac{E^*_{jj}}{E_{jj}}, \quad j = 1, 2, \ldots, n \quad (17)$$

where $E_{jj} > 0$ for all j,
we have, for $j = n$,

$$p_n \le E^*_{nn}/E_{nn},$$

which is, in effect, redundant, if assumption (A7) holds, by which $E^*_{nn}/E_{nn} \ge 1$ necessarily.

Let us write

$$R_{j,k} = B^*_{j,k}/B_{j,k}$$

and

$$e_j = E_{jj}/E_{jj}.$$

Then (17) can be written as

$$p_j \leq R_{j+1,n} e_j. \tag{18}$$

However, as $R_{j,k}$ is monotonically non-increasing in k for fixed j, condition (18) is equivalent to

$$R_{2,j} p_j/e_j \leq \lim_{n \to \infty} R_{2,n} = R, \text{ say,} \tag{19}$$

as (17) must hold for all $n > j$.

Let us examine the infinite product

$$R = \prod_{k=2}^{\infty} [\bar{a}_k{}^2/(\bar{a}_k{}^2 p_k + q_k)]$$
$$= \prod_{k=2}^{\infty} (1 - c_k), \text{ say,}$$

where

$$c_k = [(1 - \bar{a}_k{}^2)q_k]/[\bar{a}_k{}^2 p_k + q_k]$$
$$= d_k(1 - p_k)/(1 - d_k p_k),$$

with $d_k = 1 - \bar{a}_k{}^2$.

From standard results on infinite products,[8] it follows that a necessary and sufficient condition for $\prod_{k=2}^{\infty} (1 - c_k)$ to converge is that

$$\sum_{k=2}^{\infty} c_k < \infty.$$

At this stage, we state the following lemma (a proof is provided in the Appendix):

Lemma 4.1

Let $\{x_k\}$ be a sequence of positive numbers $\in (0, 1]$ such that

$$x_k = b_k(1 - c_k)/(1 - b_k c_k),$$

where $b_k, c_k \in (0, 1)$.

Then

$$\sum_{k=1}^{\infty} x_k < \infty$$

if $b_k > b_{k+1}$ and $c_k < c_{k+1}$ for all $k = 1, 2, 3, \ldots$.

This lemma provides sufficient conditions for $\sum_{k=2}^{\infty} (1 - c_k)$ to converge. These sufficient conditions are:

(i) $d_k > d_{k+1}$, i.e. $a_k > a_{k+1}$ for all k (20)

(ii) $p_k < p_{k+1}$ for all k. (21)

Let us return to condition (19). We now have sufficient conditions for R to exist. Our next problem is to find its value, if possible. For this purpose, we make use of the following lemma (a proof is provided in the Appendix):

Lemma 4.2

Let us consider the infinite product

$$\prod_{k=2}^{\infty} (1 - r_k), r_k \in (0, 1],$$

where $r_k = b_k(1 - c_k)/(1 - b_k c_k)$,

such that

(i) $b_k, c_k \in (0, 1)$,
(ii) $b_k \to 0$ as $k \to \infty$,
(iii) $b_k > b_{k+1}$ for all $k = 1, 2, \ldots$
(iv) $c_k < c_{k+1}$ for all $k = 1, 2, \ldots$

If $c_k = b_{k+1}/b_k$ for all k, then

$$\prod_{k=2}^{\infty} (1 - r_k) = (1 - b_2).$$

This lemma can be used directly for the solution of the problem at hand, namely, to find conditions under which the sequences $\{a_n\}$ and $\{p_n\}$ satisfy condition (19). We shall establish later how and why this is possible.

If we apply lemma 4.2, then

$$p_k = d_{k+1}/d_k$$
$$R = 1 - d_2 = a_2{}^2$$
$$R_{2,j} = (1 - d_2)/(1 - d_j p_j),$$

so that condition (19) becomes equivalent to

$$p_j \leq e_j(1 - d_j p_j).$$

Obviously, a sufficient condition for (19) to hold is, therefore,

$$d_{j+1} \leq d_j e_j/(1 + d_j e_j). \tag{22}$$

All the major conclusions arrived at in this section can be formally stated as follows:

Theorem 4.1

Let $(\hat{\theta}_n)$ and $(\hat{\hat{\theta}}_n)$ be sequences of estimates defined by equations (5) and (11) respectively. Let $D(.)$ be as defined in equation (4).

If

(P1) $f: R^N \to R^q$ is such that
$\mathbf{f}_i(X) \geqslant f_i(0), i = 1, 2, \ldots, q$

(P2) $\{a_n\}$ is strictly monotonically decreasing

(P3) $a_n \to 0$ as $n \to \infty$

(P4) $p_n = (1 - \bar{a}_{n+1}^2)/(1 - \bar{a}_n^2)$,
where $\bar{a}_n = 1 - a_n$

(P5) $a_n^2 > [(1 - e_n) - e_n a_n^2]$ for $n = 1, 2, \ldots$,
where $e_j = E_{jj}^*/E_{jj}$, E_{jj}^* and E_{jj} being as in equations (10) and (14) respectively,

then

$$D(\hat{\theta}_n) < D(\hat{\hat{\theta}}_n).$$

Remarks

It is interesting to note that if we take $a_n = 1/n$, then all the requirements (P1)–(P5) are satisfied.

5. AN APPROXIMATION TO λ_n

In this section we shall show that it is possible to obtain certain approximations to the zone-controlling parameter λ_n, if the following assumptions are made:

(L1) For every $k = 1, 2, \ldots, m$, the distribution of $X^{(k)}$, i.e. the feature vector having the "given" label k (as opposed to the true label), is an N-variate normal with mean vector

$$\bar{\boldsymbol{\mu}}_k = \sum_{j=1}^{m} \varepsilon_{kj}\boldsymbol{\mu}_j$$

and dispersion matrix

$$\bar{\boldsymbol{\Sigma}}_k = \sum_{j=1}^{m} \varepsilon_{kj}\boldsymbol{\Sigma}_j,$$

where

$$\varepsilon_{ki} = \alpha_{ki}\pi_i \Big/ \left(\sum_{j=1}^{m} \alpha_{kj}\pi_j \right).$$

Let $\bar{\boldsymbol{\mu}}_k(n)$ be as in Ref. (6), that is, let $\bar{\boldsymbol{\mu}}_k(n) = \sum_{j=1}^{m} \beta_{kj}(n)\boldsymbol{\mu}_j$, where

$$\beta_{kj}(n) = P(A_k(n) | \mathbf{X}, \hat{w} = k, w = j)\varepsilon_{kj}.$$

(L2) $\bar{\boldsymbol{\mu}}_k(n)$ can be approximated by $\bar{\boldsymbol{\mu}}_k$.

Remark

One situation in which conditions (L1) and (L2) can surely be expected to hold is in the ideal case, i.e. where there is no mislabeling. In such a situation these conditions become respectively equivalent to

(L1)′ For every $k = 1, 2, \ldots, m$, the distribution of $\mathbf{X}^{(k)}$, i.e. the feature vector having the "given" label k (as opposed to the true label), is N-variate normal with mean vector

$$\bar{\boldsymbol{\mu}}_k = \boldsymbol{\mu}_k$$

and dispersion matrix

$$\bar{\boldsymbol{\Sigma}}_k = \boldsymbol{\Sigma}_k;$$

(L2)′ $\bar{\boldsymbol{\mu}}_k$ is equal to $\bar{\boldsymbol{\mu}}_k(n)$ (1)

Thus whenever we assume that (L1) and (L2) hold, we are, perhaps, assuming the absence of mislabeling in the training set, an assumption which may not always be justified, so any results based on these assumptions will, at best, be approximate. However, we had to resort to them to make the problem and its treatment sufficiently tractable to yield useful results.

We have the following result:

Theorem 5.1

Let $\hat{\boldsymbol{\mu}}_n$ and $\hat{\bar{\boldsymbol{\mu}}}_n$ be as in sections 2 and 4 and let assumptions (A1)–(A7), (L1) and (L2) hold, under the set-up assumed in Section 3. Also, let

$$a_n = 1_n$$

and

$$\mathbf{B}_n^{-1} = (1/n) \sum_{j=1}^{n} (\mathbf{X}_j - \hat{\boldsymbol{\mu}}_n)(\mathbf{X}_j - \hat{\boldsymbol{\mu}}_n).$$

Then for $n > N$, a large-sample approximation to λ_n is given by

$$\lambda_n^2 = [n(n+1)/(n-1)]u_p/(1-u_p),$$

where

$c = N/(n-N)$

u_p is the lower p-percentage point of the beta distribution with degrees of freedom $N/2$ and $(n-N)/2$, so that

$$p = [B(N/2, (n-N)/2)]^{-1} \int_0^u u^{(n/2)-1}(1-u)^{((n-N)/2)-1}\,du$$

$$B(N/2, (n-N)/2) = \int_0^1 u^{(n/2)-1}(1-u)^{((n-N)/2)-1}\,du.$$

Proof. As assumptions (A6) and (A8) hold, it follows from Refs. (7) and (6) respectively that

$$\hat{\bar{\boldsymbol{\mu}}}_n \xrightarrow{a.s.} \bar{\boldsymbol{\mu}}_k$$

and

$$\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n) \xrightarrow{a.s.} 0,$$

implying that

$$(\hat{\bar{\boldsymbol{\mu}}} - \bar{\boldsymbol{\mu}}) - (\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n)) \xrightarrow{a.s.} 0. \tag{23}$$

Also, as $a_n = 1/n$, it follows that $\hat{\bar{\boldsymbol{\mu}}}_n$ is nothing but the arithmetic mean of the n observations $\mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \ldots, \mathbf{X}_n^{(k)}$, so that from well-known results of statistical sampling theory, it follows, on account of (L1), that

$$\hat{\bar{\boldsymbol{\mu}}}_n \sim N_N(\bar{\boldsymbol{\mu}}_k, (1/n)\bar{\boldsymbol{\Sigma}}_k). \tag{24}$$

Relations (23) and (24) together imply that

$$\sqrt{n}(\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n)) \xrightarrow{L} N_N(\mathbf{0}, \bar{\boldsymbol{\Sigma}}_k), \tag{25}$$

where the notation $\xrightarrow{L}$ denotes convergence in distribution or L_{aw}, "$\sim$" denotes "is distributed as" and $N_N(.,.)$ is the N- variate normal variable.

By (A5), $\mathbf{X}_{n+1}^{(k)}$ is independent of $\mathbf{X}_1^{(k)}, \mathbf{X}_2^{(k)}, \ldots, \mathbf{X}_n^{(k)}$ and hence of $\sqrt{n}(\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n))$, so that

$$[1 + 1/n]^{-1}(\mathbf{X}_{n+1} - (\hat{\boldsymbol{\mu}}_n - \hat{\boldsymbol{\mu}}_k(n))) \xrightarrow{L} N_N(\bar{\boldsymbol{\mu}}_k, \bar{\boldsymbol{\Sigma}}_k). \tag{26}$$

Also, for $i < n+1$,

$$[1 - 1/n]^{-1}(\mathbf{X}_i - (\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n))) \xrightarrow{L} N_N(\bar{\boldsymbol{\mu}}_k, \bar{\boldsymbol{\Sigma}}_k), \tag{27}$$

since, by (23), $\operatorname{cov}(\mathbf{X}_i, \hat{\boldsymbol{\mu}}_n) - \operatorname{cov}(\mathbf{X}_i, \hat{\bar{\boldsymbol{\mu}}}_k(n)) \to 0$.

Also it can easily be observed that

$$E[(\mathbf{X}_i - \bar{\boldsymbol{\mu}}_k) - (\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n))][(\mathbf{X}_j - \bar{\boldsymbol{\mu}}_k) - (\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n))]' = (1 - 1/n)\bar{\boldsymbol{\Sigma}}_k \quad \text{if } i = j$$

$$= -(1/n)\bar{\boldsymbol{\Sigma}}_k \quad \text{if } i \neq j. \tag{28}$$

Applying (L2) to (26) and (27) gives

$$\sqrt{[n/(n+1)]}(\mathbf{X}_{n+1} - \hat{\boldsymbol{\mu}}_n) \xrightarrow{L} N_N(\mathbf{0}, \bar{\boldsymbol{\Sigma}}_k) \tag{29}$$

and, for $i < n+1$,

$$\sqrt{[n/(n+1)]}(\mathbf{X}_i - \hat{\boldsymbol{\mu}}_n) \xrightarrow{L} N_N(\mathbf{0}, \bar{\boldsymbol{\Sigma}}_k). \tag{30}$$

The relation (28) implies that

$$[n/(n-1)] \sum_{i=1}^{n} E[(\mathbf{X}_i - \bar{\boldsymbol{\mu}}_k) - (\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n))][(\mathbf{X}_i - \bar{\boldsymbol{\mu}}_k) - (\hat{\boldsymbol{\mu}}_n - \bar{\boldsymbol{\mu}}_k(n))]' \sim W_N(\bar{\boldsymbol{\Sigma}}_k | n-1),$$

the N-variate central Wishart distribution with $n-1$ degrees of freedom, so that we are justified in claiming that

$$(n/(n-1)) \sum_{i=1}^{n} (\mathbf{X}_i - \hat{\boldsymbol{\mu}}_n)(\mathbf{X}_i - \hat{\boldsymbol{\mu}}_n)' \xrightarrow{L} W_N(\bar{\boldsymbol{\Sigma}}_k | n-1), \tag{31}$$

i.e.

$$(n^2/(n-1))\mathbf{B}_n \xrightarrow{L} W_N(\bar{\boldsymbol{\Sigma}}_k | n-1).$$

Relations (29) and (31) together imply that

$$[(n-1)^2/n(n+1)](\mathbf{X}_{n+1} - \hat{\boldsymbol{\mu}}_n)'\mathbf{B}_n^{-1}(\mathbf{X}_{n+1} - \hat{\boldsymbol{\mu}}_n) \xrightarrow{L} T^2_{n-1},$$

the N-variate central Hotelling T^2 distribution with $n-1$ degrees of freedom, i.e.

$$[(n-1)^2/n(n+1)]\, d^2(\mathbf{X}^{(k)}_{n+1}, \hat{\boldsymbol{\mu}}_n) \xrightarrow{L} T^2_{n-1}. \tag{30}$$

The theorem follows from this if we remember that

(1) if T is an N-variate central Hotelling T^2-statistic with k degrees of freedom, then

$$[(k-N+1)/N](T/k) \sim F(N, N-k+1),$$

the central F-statistic with $(N, N-k+1)$ degrees of freedom,

(2) if F is a central F-statistic with (m, n) d.f. then

$$U = cF/(1+cF) \sim \text{Beta}(m/2, n/2),$$

the Beta variate with $(m/2, n/2)$ d.f., where $c = m/n$.

Hence the theorem is proved.

Remarks

(1) Karl Pearson tabulated the incomplete beta function

$$I_u(m, n) = \int_0^u x^{m-1}(1-x)^{n-1}\, dx$$

for a large number of values of m and n.[9] It is not difficult to determine the approximate value of n given above with the help of these tables.

(2) These approximations of λ_n depend only on the dimension N of the feature vector, apart from n, so it is possible to tabulate their values for different N for a large number of values of n, for purposes of ready reference.

(3) The point mentioned under (2) actually highlights a distinct advantage of the given method for determining λ_n, as compared with the methods used earlier.[4,6] The latter involves a fair amount of computation, as the eigenvalues of an $N \times N$ matrix have to be computed at each iteration. Further, the values of n have to be computed afresh for every new problem.

(4) A word of caution is necessary here. The given method is only an approximate one and is based mostly on large-sample theory, so it is quite possible that the values obtained may not yield very satisfactory results in small-sample situations.

6. IMPLEMENTATION AND RESULTS

To demonstrate the different features of the proposed method for evaluating λ_n, the GGA is applied to an artificially generated data set for a two-feature three-class pattern recognition problem, the values of λ_n being calculated by the method mentioned in Theorem 5.1. The data set was generated using random normal deviates from Ref. (10), with mean vectors and covariance matrices as given in Table 5. The method used for obtaining a sample (x, y) from the population $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where

$$\boldsymbol{\mu} = (\mu_x, \mu_y)' \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$$

from a pair of random normal deviates (γ_x, γ_y) was based on the following well-known result:

Lemma 5.1.

If (x, y) is distributed in the bivariate normal form with mean vector

$$\boldsymbol{\mu} = (\mu_x, \mu_y)'$$

and dispersion matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix},$$

then $(y|x)$ is also normally distributed with

$$E(y|x) = \mu_{y|x} = \mu_y + (\sigma_y/\sigma_x)(x - \mu_x)$$

and

$$\text{var}(y|x) = \sigma^2_{y|x} = \sigma_y^2(1-\rho^2).$$

Thus, we have $x = \mu_x + \tau_x\sqrt{\sigma_{xx}}$

and $y = \mu_{y|x} + \tau_y\sigma_{y|x}$.

From the sets of samples so obtained for each of the three classes, training sets of size 20 for each of them were obtained by mixing at random the elements of the three sample sets, using the following $((\alpha_{ij}))$-matrix:

$$((\alpha_{ij})) = \frac{1}{20}\begin{bmatrix} 17 & 1 & 2 \\ 1 & 16 & 3 \\ 2 & 3 & 15 \end{bmatrix}.$$

Table 1. Lambda values for feat-

Iteration no. (n)	Lambda value
1	—
2	—
3	11.2450
4	5.5656
5	4.4333
6	3.8401
7	3.5659
8	3.3988
9	3.2876
10	3.2104
11	3.1556
12	3.1153
13	3.0839
14	3.0596
15	3.0415
16	3.0284
17	3.0165
18	3.0078
19	3.0022
20	2.9968

This means, for example, that the training set for class 1 contains 17 samples from class 1, 1 sample from class 2 and 2 samples from class 3, and so on.

The values of λ_n obtained using Theorem 5.1 with $N = 2$ and the help of Ref. (9), are given in Table 1 for $n = 1, 2, \ldots, 20$. The GGA and the non-GGA were implemented on the three training sets for a number of different permutations of the samples within each set. In each case, for each of the classes, we computed, after every iteration and for both algorithms, the "average" distances of the estimates from the two sets of "true" parameter values defined below in the form of their MSEs, as

$$\sqrt{\left[(1/n)\sum_{i=1}^{n}(\boldsymbol{\mu}_i - \boldsymbol{\mu}^*)'(\boldsymbol{\mu}_i - \boldsymbol{\mu}^*)\right]}.$$

where $\boldsymbol{\mu}^*$ is the "true" value chosen.

The two types of "true" parameter values considered are

(1) the "true" sample parameter values obtained with the help of all the correctly labeled training samples of the respective classes, and

(2) the "true" population parameter values.

These "true" parameter values are given in Table 5. The "average" distance (MSE) defined above is simply the square root of the arithmetic mean of the squared Euclidean distances of the estimates from their "true" values. These individual Euclidean distances too, were taken into account for purposes of comparison of the two algorithms.

It was found, in a large majority of cases, that the distances, particularly the MSEs, for the GGA-estimates were smaller than those for the non-GGA-estimates. This was found to be strictly true in the cases where the sample "true" values were the values used as the standard. As a typical example, the complete results for one particular permutation of the training sets using the sample "true" values as the standard, are given in Tables 2, 3 and 4 respectively. The initial and final estimates, as obtained by both the algorithms, are given in Table 5. Also, it should be noted that we have estimated the "uncorrected" second-order moments rather than the central second-order moments. In other words, we have estimated $E(\mathbf{x}\mathbf{x}')$ rather than $E(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})'$. This had to be done because otherwise condition (P1) would not have been satisfied.

Another very important feature observed in Tables 2–4 is that with the set of λ_n values obtained automatically by the method described here, the GGA was able to detect and discard all the mislabeled samples in most of the cases examined. This is definitely a positive feature of the present work.

7. CONCLUSION

In this work we have presented a method for the automatic determination of the threshold values λ_n for a certain class of recursive parameter learning algorithms (GGAs)[1,4] which are used to discard "doubtful" training samples. This has been done under certain normality assumptions. The values obtained are found to be independent of the class parameter values. They depend only on the values of the feature-vector dimension N and the iteration number n through the percentage-points of the beta distribution with parameters $N/2$ and $(n - N)/2$. The effectiveness of the thresholds so obtained has been demonstrated on a three-class two-feature pattern recognition problem.

SUMMARY

This work is a continuation of our earlier work[1,4] on a class of recursive learning algorithms (GGA) which resort to restricted updating to tackle the problem of the presence of "dubious" training samples. Basically, such algorithms operate by computing the distance of the current training sample $\mathbf{X}_n$ from the current estimate $\hat{\boldsymbol{\mu}}_{n-1}$ of the mean vector and compare it with some threshold value λ_n; the current training sample is selected for updating only if the distance is less than the threshold. In this work we have presented a method for the automatic selection of such threshold. This has been done by minimizing the MSE for the GGA with respect to the threshold value λ_n. Making certain assumptions about the normality of the unconditional distribution of the feature vector, we have obtained estimates for λ_n as $\lambda_n^2 = [n(n+1)/(n-1)]u_p/(1-u_p)$,
where

$c = N/(n - N)$,
N is the dimension of the feature vector,
n is the iteration number,
u_p is the lower p-percentage point of the beta distribution with degrees of freedom $N/2$ and $(n - N)/2$, so that

Table 2. Learning of means and covariances for class 1 using GGA and non-GGA

Sample no.	Training sample		True class	Distance	Update?	Euclidean distanees from "true" sample values of: GGA estimates of: Mean vector: Indiv.	GGA: Mean vector: Av.	GGA: Covariance: Indiv.	GGA: Covariance: Av.	non-GGA: Mean vector: Indiv.	non-GGA: Mean vector: Av.	non-GGA: Covariance: Indiv.	non-GGA: Covariance: Av.
1	9.41	14.72	1			1.358	1.358	34.219	34.219	1.358	1.358	34.219	34.219
2	11.20	15.09	1			0.592	1.047	14.265	26.215	0.592	1.047	14.265	26.215
3	12.38	13.80	1	4.12	Y	0.248	0.867	5.975	21.681	0.248	0.867	5.975	21.681
4	10.82	14.34	1	0.57	Y	0.206	0.758	4.939	18.938	0.206	0.758	4.939	18.938
5	10.58	16.67	1	5.40	N	0.206	0.684	4.939	17.082	0.433	0.705	12.468	17.832
6	8.91	15.46	1	8.47	N	0.206	0.630	4.939	15.723	0.539	0.680	14.408	17.309
7	10.38	13.25	1	6.61	N	0.206	0.588	4.939	14.676	0.334	0.642	8.406	16.337
8	5.41	6.48	2	42.40	N	0.206	0.555	4.939	13.839	1.167	0.729	27.726	18.155
9	9.42	11.41	1	19.76	N	0.206	0.528	4.939	13.151	1.388	0.828	35.183	20.749
10	8.48	14.39	1	8.46	N	0.206	0.505	4.939	12.573	1.416	0.904	36.373	22.799
11	8.07	10.19	3	30.29	N	0.206	0.486	4.939	12.080	1.737	1.009	45.636	25.727
12	9.66	13.66	1	7.60	N	0.206	0.469	4.939	11.654	1.706	1.084	45.409	27.902
13	10.33	14.48	1	2.33	Y	0.084	0.451	2.807	11.224	1.599	1.132	42.795	29.318
14	6.06	9.49	3	73.16	N	0.084	0.435	2.807	10.841	1.975	1.212	52.343	31.525
15	10.91	12.82	1	16.95	N	0.084	0.421	2.807	10.499	1.918	1.271	51.555	33.238
16	12.20	14.37	1	4.69	N	0.084	0.408	2.807	10.189	1.746	1.306	47.052	34.265
17	11.99	16.39	1	10.24	N	0.084	0.396	2.807	9.909	1.512	1.319	39.983	34.628
18	13.60	16.48	1	1.65	Y	0.622	0.412	18.223	10.544	1.244	1.315	31.720	34.472
19	11.03	12.96	1	0.72	Y	0.507	0.418	13.452	10.717	1.235	1.311	32.083	34.351
20	11.41	16.42	1	0.74	Y	0.592	0.428	17.098	11.123	1.077	1.300	27.268	34.032

Table 3. Learning of means and covariances for class 2 using GGA and non-GGA

Sample no.	Training sample		True class	Distanee	Update?	Euclidean distances from "true" sample values of: GGA estimates of: Mean vector: Indiv.	GGA: Mean vector: Av.	GGA: Covariance: Indiv.	GGA: Covariance: Av.	non-GGA: Mean vector: Indiv.	non-GGA: Mean vector: Av.	non-GGA: Covariance: Indiv.	non-GGA: Covariance: Av.
1	5.56	4.54	2			1.144	1.144	10.499	10.499	1.144	1.144	10.499	10.499
2	4.68	4.42	2			0.730	0.959	5.900	8.516	0.730	0.959	5.900	8.516
3	6.19	4.72	2	1.73	Y	1.079	1.001	10.056	9.058	1.079	1.001	10.056	9.058
4	5.37	4.02	2	0.96	Y	1.000	1.000	8.966	9.035	1.000	1.000	8.966	9.035
5	0.51	4.19	2	9.34	N	1.000	1.000	8.966	9.021	0.315	0.906	1.975	8.129
6	2.57	4.29	2	11.70	N	1.000	1.000	8.966	9.012	0.472	0.849	2.531	7.493
7	4.11	3.68	2	8.88	N	1.000	1.000	8.966	9.005	0.423	0.802	3.019	7.030
8	2.27	6.19	2	3.18	Y	0.769	0.974	6.648	8.745	0.750	0.796	5.568	6.864
9	6.88	3.11	2	1.36	Y	0.774	0.954	7.088	8.577	0.396	0.762	2.151	6.511
10	3.04	9.57	3	12.41	N	0.774	0.938	7.088	8.440	0.897	0.776	8.992	6.800
11	4.37	4.07	2	0.77	Y	0.645	0.915	5.492	8.216	0.821	0.781	8.166	6.936
12	9.78	8.27	1	5.64	N	0.645	0.895	5.492	8.024	1.024	0.804	12.932	7.618
13	7.96	5.73	2	3.77	N	0.645	0.879	5.492	7.858	1.126	0.833	14.872	8.401
14	2.98	5.00	2	8.88	N	0.645	0.864	5.492	7.714	1.081	0.853	13.688	8.884
15	4.50	8.89	3	17.30	N	0.645	0.851	5.492	7.586	1.325	0.892	16.774	9.613
16	4.26	5.44	2	5.40	N	0.645	0.840	5.492	7.472	1.325	0.925	16.403	10.171
17	4.25	9.97	3	9.93	N	0.645	0.830	4.492	7.370	1.591	0.977	20.214	11.018
18	4.99	3.09	2	1.88	Y	0.560	0.817	4.420	7.238	1.451	1.009	18.604	11.571
19	4.32	5.54	2	2.08	Y	0.559	0.806	4.413	7.117	1.450	1.037	18.301	12.020
20	5.23	4.07	2	0.66	Y	0.562	0.795	4.349	7.005	1.382	1.057	17.406	12.345

$$p = [B(N/2,(n-N)/2)]^{-1}\int_0^u u^{(n/2)-1}(1-u)^{((n-N)/2)-1}\,du$$

$$B(N/2,(n-N)/2) = \int_0^1 u^{(n/2)-1}(1-u)^{((n-N)/2)-1}\,du.$$

It is to be noted that these estimates are distribution-free, that is, they do not involve the parameters of the underlying distribution. They depend only on n, the iteration number, and N, the dimension of the feature vector and are simple functions of certain percentage points of the beta distribution with parameters $N/2$ and $(n-N)/2$. As comprehensive tables[9] for the cumulative probabilities of this distribution are available for a large number of values of the parameters, the determination of λ_n is a simple matter. These approximations to the λ_n values thus have the added advantage that they do not have to be computed afresh for each problem; they can be tabulated for different values of N and n once and for all. To

Table 4. Learning of means and covariances for class 3 using GGA and non-GGA

Sample no.	Training sample		True class	Distance	Update?	Euclidean distances from "true" sample values of: GGA-estimates of: Mean vector: Indiv.	GGA-estimates of: Mean vector: Av.	GGA-estimates of: Covariance: Indiv.	GGA-estimates of: Covariance: Av.	non-GGA estimates of: Mean vector: Indiv.	non-GGA estimates of: Mean vector: Av.	non-GGA estimates of: Covariance: Indiv.	non-GGA estimates of: Covariance: Av.
1	4.30	8.51	3			2.041	2.041	42.445	42.445	2.041	2.041	42.445	42.445
2	6.62	10.81	3			0.779	1.545	15.615	31.980	0.779	1.545	15.615	31.980
3	5.50	10.34	3	2.03	Y	0.587	1.306	11.727	26.975	0.587	1.306	11.727	26.975
4	6.57	11.69	3	2.22	Y	0.606	1.171	9.167	23.806	0.606	1.171	9.167	23.806
5	4.95	6.46	2	7.97	N	0.606	1.082	9.167	21.684	0.924	1.126	15.521	22.396
6	3.59	11.43	3	3.28	Y	0.254	0.993	4.947	19.898	0.510	1.049	9.316	20.795
7	4.56	11.88	3	1.74	Y	0.408	0.932	8.442	18.696	0.212	0.974	4.227	19.319
8	4.72	8.25	3	8.21	N	0.408	0.884	8.442	17.741	0.452	0.925	8.964	18.347
9	4.27	4.62	2	36.76	N	0.408	0.844	8.442	16.962	1.049	0.940	18.380	18.351
10	4.86	11.12	3	0.14	Y	0.455	0.814	9.258	16.355	0.874	0.933	15.390	18.076
11	5.03	4.03	2	20.31	N	0.455	0.788	9.258	15.842	1.367	0.981	22.870	18.563
12	8.61	14.16	1	23.09	N	0.455	0.766	9.258	15.401	0.944	0.978	12.136	18.115
13	6.71	10.64	3	1.42	Y	0.474	0.747	9.320	15.021	0.879	0.970	11.054	17.672
14	8.67	10.02	1	3.12	N	0.474	0.731	9.320	14.688	0.946	0.969	12.134	17.336
15	7.01	7.01	3	10.47	N	0.474	0.717	9.320	14.392	1.139	0.981	15.405	17.214
16	2.54	9.12	3	2.16	Y	0.270	0.697	4.973	13.990	1.066	0.986	15.404	17.106
17	5.58	11.55	3	0.37	Y	0.343	0.682	6.735	13.671	0.949	0.984	13.258	16.904
18	2.56	13.24	3	3.05	N	0.343	0.667	6.735	13.380	0.695	0.970	8.623	16.553
19	2.48	10.92	3	1.58	Y	0.463	0.658	7.445	13.135	0.598	0.955	7.569	16.205
20	9.53	9.07	3	2.83	Y	0.238	0.644	4.435	12.840	0.693	0.943	9.223	15.929

Table 5. True and estimated parameter values for the three classes

	Class 1		Class 2		Class 3	
Population values of						
Mean vector	10.000	15.000	5.000	5.000	5.000	10.000
Covariance matrix*	103.000	152.000	29.000	25.000	30.000	49.000
(uncorrected)	152.000	233.000	25.000	29.000	49.000	103.000
True sample estimates of						
Mean vector	10.747	14.512	4.516	4.069	5.142	10.372
Covariance matrix*	117.200	156.605	23.536	18.100	30.038	52.372
(uueorrected)	156.605	212.656	18.100	18.323	52.372	110.139
Iuitial estimates of						
Mean vector	9.406	14.720	5.558	4.540	4.305	8.511
Covariance matrix*	88.471	138.456	30.891	25.233	18.529	36.636
(uncorrected)	138.456	216.682	25.233	20.612	36.636	72.436
Final GGA estimates of						
Mean vector	11.272	14.787	4.986	4.377	5.237	10.590
Covariance matrix*	128.468	167.198	26.249	21.061	31.055	55.305
(uneorrected)	167.198	219.949	21.061	19.993	55.305	113.306
Fiual non-GGA estimates of						
Mean vector	10.112	13.643	4.691	5.440	5.433	9.743
Covariance matrix*	106.176	141.853	25.991	25.748	33.402	53.318
(uncorrected)	141.853	192.547	25.748	33.765	53.318	101.604

* The matrix $E(XX')$.

our knowledge, this problem has not been tackled before. The efficacy of our method has been demonstrated with the help of a data set for a two-feature three-class pattern recognition problem, for which highly satisfactory results have been obtained. It has also been observed that the use of the threshold-values obtained here also greatly enhances the capability of the GGA to detect and discard mislabeled samples automatically, thus simplifying greatly the task of self-supervised learning.

REFERENCES

1. A. Pathak and S. K. Pal, A generalized learning algorithm based on guard zones, *Pattern Recognition Lett.* 4, 63–69 (1986).
2. S. K. Pal, A. K. Dutta and D. Dutta Majumder, A self-supervised vowel recognition system, *Pattern Recognition*, **12**, 27–34 (1980).
3. Y. T. Chien, The threshold effects of a non-linear learning algorithm for pattern recognition, *Information Sci.* **2**, 351–358 (1970).
4. S. K. Pal, A. Pathak and C. Basu, Dynamic guard zone

for self-supervised learning, *Pattern Recognition Lett.* **7**, 135–144 (1988).
5. A. Pal (Pathak) and S. K. Pal, Effect of wrong samples on the convergenee of learning proeesses—II: a remedy, *Information Sci.* (in press).
6. C. B. Chittineni, Learning with imperfectly labeled samples, *Pattern Recognition* **12**, 281–291 (1980).
7. A. Pathak-Pal and S. K. Pal. Learning with mislabeled training samples nsing stochastic approximation, *IEEE Trans. Syst. Man Cybern.* **SMC-17**, 1072–1077 (1987).
8. K. Knopp, *Theory ond Application of Infinite Series.* Blackie, Glasgow (1959).
9. K. Pearson, *Tables of the Incomplete Beta Function.* Biometrika Trustees, London (1968).
10. H. Wold, *Random Normal Deviates*, Tracts for Computers No. 25. Cambridge University Press, Cambridge (1954).

APPENDIX

Proof of Lemma 4.1

Let ns write

$$g(b, c) = b(1 - c)/(1 - bc).$$

Then

$$\frac{\partial g}{\partial b} > 0 \quad \text{and} \quad \frac{\partial g}{\partial c} < 0,$$

implying that if $b_k > b_{k+1}$ and $c_k < c_{k+1}$, then

$$g(b_k, c) > g(b_{k+1}, c) \text{ whatever } c \text{ may be,}$$

and

$$g(b, c_k) > g(b, c_{k+1}) \text{ whatever } b \text{ may be,}$$

so that

$$g(b_k, c_k) > g(b_k, c_{k+1}) > g(b_{k+1}, c_{k+1})$$

whatever k may be, i.e.

$$x_k > x_{k+1} \text{ for all } k.$$

Hence, by the D'Alembert ratio test for the convergence of any series of positive terms. the lemma follows.

Proof of Lemma 4.2

This is rather obvious, as

$$\prod_{k=2}^{\infty} (1 - r_k) = \lim_{n \to \infty} \prod_{k=2}^{n} (1 - r_k)$$

and

$$\begin{aligned} \prod_{k=2}^{n} (1 - r_k) &= \frac{1 - b_2}{1 - b_2 c_2} \cdot \frac{1 - b_3}{1 - b_3 c_3} \cdots \frac{1 - b_n}{1 - b_n c_n} \\ &= (1 - b_2)/(1 - b_n c_n) \text{ as } c_k = b_{k+1}/b_k \\ &\to (1 - b_2) \text{ as } n \to \infty, \end{aligned}$$

as

$$1 \geq 1 - b_n c_n > 1 - b_n \to 1 \text{ as } n \to \infty,$$

implying that

$$1 - b_n c_n \to 1 \text{ as } n \to \infty.$$

About the Author—AMITA PAL (PATHAK) obtained a B.Sc. degree (Hons) in Statistics from Presidency College, Calcutta, in 1979, and an M.Sc. degree in Statistics from the University of Calcutta in 1981. She is now a lecturer under the FGCS/KBCS project at the Indian Statistical Institute, Calcutta, her Ph.D. (as yet unawarded) work being in the field of learning in pattern recognition systems.

About the Author—DR PAL obtained a B.Sc. (Hons) in Physies, and B.Tech., M.Tech. and Ph.D. degrees in Radiophysics and Electronics, in 1969, 1972, 1974 and 1979 respectively, from the University of Calcutta. In 1982 he received a Ph.D. in Electrical Engineering together with a DIC from Imperial College, University of London. He was a recipient of a Commonwealth Scholarship in 1979 and an MRC (U.K.) Post-doetoral Award in 1981 to work at Imperial College, London. In 1986 he was awarded a Fulbright Post-doctoral Visiting Fellowship to work at the University of California, Berkeley, and the University of Maryland, Collegc Park, and to lecture at U.S. universities and laboratories. He is now a Professor in the Electronics and Communication Sciences Unit and the Professor-in-Charge of the Physical and Earth Sciences Division, Indian Statistical Institnte, Calcutta. He was also a Guest Lectnrer (1983–86) in Computer Science, Calcutta University. His research interests mainly include pattern recognition, image processing, artificial intelligence and fuzzy sets and systems. He is a co-author of the book *Fuzzy Mathematical Approach to Pattern Recognition*, John Wiley (Halsted Press), 1986, which received the Best Prodoction Award in the 7th World Book Fair, New Delhi. He has more than 100 research papers, including nine in edited books and about 60 in international journals to his credit. Dr Pal is a reviewer of the *Mathematical Reviews* (American Mathematical Society) in the fields of fuzzy sets, logic and applications, a Senior Member of the IEEE, a Fellow of the IETE (India) and Treasurer of the Indian Society for Fuzzy Mathematics and Information Processing (ISFUMIP).