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 $\lambda_i$ (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 $\lambda_i$ 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.

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\(^2\) 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 ($\lambda_i$) 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 $\lambda_i$ 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 $\lambda_i$ was obtained for which the MSE for the GGA is smaller than that for the non-GGA. In other words, the value of $\lambda_i$ 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)\(^11\)

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

$$X = [x_1, x_2, \ldots, x_n], \quad X \in \mathbb{R}^n.$$

Let us assume that:

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

(A2) The densities $P_k(\cdot)$ involve a $q$-dimensional
parameter vector $\theta_k$, which needs to be learned, either wholly or partly.

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

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

exist.

(A4) An unbiased statistic exists for the parameter vector $\theta$.

Let us suppose that for the purpose of learning $\theta$, a set of samples

$$\{X_1^{(k)}, X_2^{(k)}, \ldots, X_{n_k}^{(k)}\}$$

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 $\theta_k$ recursively is as follows:

$$\hat{\theta}_n^{(k)} = f(X_i^{(k)}) \quad \text{for} \quad n = 1$$

$$\hat{\theta}_{n-1}^{(k)} - \alpha_n Y_n^{(k)} \quad \text{for} \quad n > 1, \quad (1a)$$

where

$$Y_n^{(k)} = \hat{\theta}_n^{(k)} - f(X_i^{(k)})$$

is the $n$-stage estimate of $\theta_k$, $\{\alpha_n\}$ is a sequence of positive numbers, $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous map, defining an unbiased statistic for $\theta_k$, $\hat{\theta}_n^{(k)}$ is the $n$-th stage GGA estimate of $\theta_k$, $G(\hat{\theta}_n^{(k)}, \lambda_n) = \{X: X \in \mathbb{R}^n, d(X, \hat{\theta}_n^{(k)}) \leq \lambda_n\}$, $d^2(x,y) = (x-y)^T B_n(x-y)$, $B_n$ is a symmetric positive definite matrix, which may or may not be a function of the training samples $X_n^{(k)}$ (some examples are given in Refs (1) and (5)), $\lambda_n$ is a positive number, suitably chosen.

Incidentally, $G(\alpha, \lambda)$ is the guard-zone and, clearly, is nothing but a closed ball centred at $\alpha$ and having radius $\lambda$. 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 $\{\alpha_n\}, \lambda_n$ and $B_n$, 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.\(^{(15)}\)

3. MODELLING MISLABELED TRAINING SAMPLES

A very simple but realistic model, inspired by Chittineni,\(^{(16)}\) 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,

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

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

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

Clearly,

$$\sum_{k=1}^{m} \alpha_j = 1.$$

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

$$p(X_k^{(n)}) = p(X_k|\hat{w} = k) = p(X_k|w = k)$$

$$= \sum_{j=1}^{m} \beta_{kj}(n) p(X_k|w = j) \quad \text{if} \quad X_k^{(n)} \in A_k(n),$$

$$\text{given } \hat{w} = k \quad \text{otherwise,} \quad (3a)$$

where $A_k(n) = \{x: x \in G(\mu_k^{(n)}, \lambda_n)\}$,

$$\beta_{kj}(n) = P(A_k(n)|X_k|\hat{w} = k), \quad \text{for} \quad j, k = 1, 2, \ldots, m. \quad (3c)$$

$$\beta_{kj}(n) = P(A_k(n)|X_k|\hat{w} = k), \quad \text{for} \quad j = 1, 2, \ldots, m. \quad (3d)$$

provided we are prepared to assume that:

(A6) $P(X|\hat{w} = k, w = j) = p(X|w = j)$ for all $j, k = 1, 2, \ldots, m$.

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

(A8) $P(\hat{w} = k, A_2(n)) > 0$ for all $k, n$.

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

It is not difficult to observe that $\beta_{kj}(n), \beta_{kj}^{(n)}(n) \in [0, 1]$ for all $k, j = 1, 2, \ldots, m$, as it is known that

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

for discrete algorithms of learning.

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 $\theta$. One convenient performance index of learning is the MSE of the estimate at each instant, namely,

$$D(\hat{\theta}_n) = E[(\hat{\theta}_n - \theta)^2] \quad (4)$$

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

\[
E(Z_n) = B_{2n}E(Z_1) + \sum_{j=2}^{n} B_{j+1,0} \rho_j E(T_j),
\]

where \( B_{i,j} = \prod_{k=1}^{j} (a_k^n + q_k). \)

Thus \( Z_1 = Q_1 \), and

\[
E(Z_1) = a_1^2 E(Q_1) + 2a_1 \rho_1 E(P'_{-1}Q_1).
\]

As \( E(T_1) = a_1^2 E(Q_1Q_1) + 2a_1 \rho_1 E(P'_{-1}Q_1) \)

and \( E(P'_{-1}Q_1) = 0 \),

on account of our assumption (AS) regarding the independence of the observations, we have

\[
E(T_1) = a_1^2 E(Q_1Q_1).
\]

Thus

\[
E(Z_n) = B_{2n}E(Z_1) + \sum_{j=2}^{n} B_{j+1,0} \rho_j E(Q_1Q_1),
\]

where \( a_j = 1 - a_j \).

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

\[
a_1 = 1,
\]

\[
p_1 = 1.
\]

Equation (9) is therefore equivalent to

\[
E(Z_n) = \sum_{j=1}^{n} B_{j+1,0} a_j^2 \rho_j E(T_j).
\]

where

\[
E(T_j) = a_j^2 E(Q_jQ_j),
\]

and, for \( n > 1 \),

\[
E(T_n) = a_n^2 E(Q_nQ_n) + 2a_n \rho_n E(T_{n-1}),
\]

where \( T_n = a_n^2 Q_nQ_n + 2a_n \rho_n P'_{n-1}Q_n \),

\[
Q_n = f(X_n) - \theta.
\]

Thus \( Z_1 = Q_1 \), and

\[
E(Z_1) = a_1^2 E(Q_1Q_1) + 2a_1 \rho_1 E(P'_{-1}Q_1).
\]

As \( E(T_1) = a_1^2 E(Q_1Q_1) + 2a_1 \rho_1 E(P'_{-1}Q_1) \)

and \( E(P'_{-1}Q_1) = 0 \),

on account of our assumption (AS) regarding the independence of the observations, we have

\[
E(T_1) = a_1^2 E(Q_1Q_1).
\]

Thus

\[
E(Z_n) = B_{2n}E(Z_1) + \sum_{j=2}^{n} B_{j+1,0} \rho_j E(Q_1Q_1),
\]

where \( a_j = 1 - a_j \).

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

\[
a_1 = 1,
\]

\[
p_1 = 1.
\]

Equation (9) is therefore equivalent to

\[
E(Z_n) = \sum_{j=1}^{n} B_{j+1,0} a_j^2 \rho_j E(T_j).
\]

where

\[
E(T_j) = a_j^2 E(Q_jQ_j),
\]

and, for \( n > 1 \),

\[
E(T_n) = a_n^2 E(Q_nQ_n) + 2a_n \rho_n E(T_{n-1}),
\]

where \( T_n = a_n^2 Q_nQ_n + 2a_n \rho_n P'_{n-1}Q_n \),

\[
Q_n = f(X_n) - \theta.
\]

Thus \( Z_1 = Q_1 \), and

\[
E(Z_1) = a_1^2 E(Q_1Q_1) + 2a_1 \rho_1 E(P'_{-1}Q_1).
\]

As \( E(T_1) = a_1^2 E(Q_1Q_1) + 2a_1 \rho_1 E(P'_{-1}Q_1) \)

and \( E(P'_{-1}Q_1) = 0 \),

on account of our assumption (AS) regarding the independence of the observations, we have

\[
E(T_1) = a_1^2 E(Q_1Q_1).
\]

Thus

\[
E(Z_n) = B_{2n}E(Z_1) + \sum_{j=2}^{n} B_{j+1,0} \rho_j E(Q_1Q_1),
\]

where \( a_j = 1 - a_j \).

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

\[
a_1 = 1,
\]

\[
p_1 = 1.
\]

Equation (9) is therefore equivalent to

\[
E(Z_n) = \sum_{j=1}^{n} B_{j+1,0} a_j^2 \rho_j E(T_j).
\]

where

\[
E(T_j) = a_j^2 E(Q_jQ_j),
\]

and, for \( n > 1 \),

\[
E(T_n) = a_n^2 E(Q_nQ_n) + 2a_n \rho_n E(T_{n-1}),
\]

where \( T_n = a_n^2 Q_nQ_n + 2a_n \rho_n P'_{n-1}Q_n \),

\[
Q_n = f(X_n) - \theta.
\]

Thus \( Z_1 = Q_1 \), and

\[
E(Z_1) = a_1^2 E(Q_1Q_1) + 2a_1 \rho_1 E(P'_{-1}Q_1).
\]

As \( E(T_1) = a_1^2 E(Q_1Q_1) + 2a_1 \rho_1 E(P'_{-1}Q_1) \)

and \( E(P'_{-1}Q_1) = 0 \),

on account of our assumption (AS) regarding the independence of the observations, we have

\[
E(T_1) = a_1^2 E(Q_1Q_1).
\]

Thus

\[
E(Z_n) = B_{2n}E(Z_1) + \sum_{j=2}^{n} B_{j+1,0} \rho_j E(Q_1Q_1),
\]

where \( a_j = 1 - a_j \).

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

\[
a_1 = 1,
\]

\[
p_1 = 1.
\]

Equation (9) is therefore equivalent to

\[
E(Z_n) = \sum_{j=1}^{n} B_{j+1,0} a_j^2 \rho_j E(T_j).
\]

where

\[
E(T_j) = a_j^2 E(Q_jQ_j),
\]

and, for \( n > 1 \),

\[
E(T_n) = a_n^2 E(Q_nQ_n) + 2a_n \rho_n E(T_{n-1}),
\]

where \( T_n = a_n^2 Q_nQ_n + 2a_n \rho_n P'_{n-1}Q_n \),

\[
Q_n = f(X_n) - \theta.
\]
Clearly, proceeding as before,

\[ E(\hat{\theta}_n) = A_{\theta, n}E(\hat{\theta}_1) + \sum_{j=1}^{n} a_j E_{j+1, n}E(f(X_j)), \]  

(12)

where

\[ A_{\theta, j} = \prod_{k=j}^{1} \hat{a}_k = \prod_{k=j}^{1} (1 - a_k). \]

Writing

\[ Z_{\theta} = \|\hat{\theta}_n - \theta\|^2 \text{ if } n = 1 \]
\[ = \hat{a}_n^2 Z_{n-1} + T_n \text{ if } n > 1, \]

where

\[ T_j = a_j^2 Q_j Q_j + 2a_j P_{j-1} Q_j, \]
\[ P_{\theta} = \hat{\theta}_n - \theta, \]

so that

\[ E^*(T_j) = a_j^2 E^*(Q_j) \text{ as } E^*(P_{j-1} Q_j) = 0, \]

we have

\[ E^*(Z_{\theta}) = B_{\theta, n}E^*(Z_{\hat{\theta}}) \]
\[ + \sum_{j=2}^{n} B_{j+1, n} a_j^2 E^*(Q_j Q_j), \]

where \( E \) denotes expectation under the probability model given by Pathak-Pal and Pal, (10)

and

\[ B_{\theta, n} = \prod_{k=1}^{n} \hat{a}_k. \]

Note. Equations (12) and (13) can also be obtained directly from equations (7) and (9) by specializing the value of \( a_j = 1 \) for all \( j \). Clearly, this is because the non-GGA is a special case of the GGA for which \( \theta_k = 0 \).

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

\[ E^*(Z_{\theta}) = \sum_{j=1}^{n} a_j^2 B_{j+1, n} E^*(Q_j Q_j) \]
\[ = \sum_{j=1}^{n} a_j^2 B_{j+1, n} E_{j+1, n}, \]  

(13)

where \( E_{j+1, n} = E^*(Q_j Q_j) \)
\[ = \sum_{i=1}^{n} \epsilon_{j, k}[r_j + \|\theta_k + \theta_k\|^2], \]  

(14a)

where

\[ \epsilon_{j, k} = \beta_{j}(n) + \beta_{j}(n) \text{ for all } n > 0. \]

As we must have

\[ E_{j, n} = \sum_{i=1}^{n} [\beta_{j}(n) + \beta_{j}(n)][r_i + \|\theta_i + \theta_k\|^2], \]

so that

\[ E_{j, n} - E_{j, n} = \sum_{i=1}^{n} \beta_{j}(n)[r_i + \|\theta_i + \theta_k\|^2]. \]

(14b)

If we assume that

\( f_j(X) > f_j(0) \) for all \( i = 1, 2, \ldots, q \)

then

\[ E_{j, n} > E_{j, n}. \]

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

\[ E(Z_n) < E^*(Z_{\theta}) \]

if and only if

\[ B_{j+1, n} E_{j+1, n} < B_{j+1, n} E_{j+1, n} \text{ for all } j = 1, 2, \ldots, n. (15) \]

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

First of all, we note that, as

\[ 0 \leq \hat{a}_j^2 \leq \hat{a}_j^2 p_j + q_j \leq 1, \text{ whatever } j, \]

we must have, for all \( i, j, \)

\[ B_{\theta, j} \leq B_{\theta, j}. \]  

(16)

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

Rewriting the inequality (15) as

\[ p_j \leq \frac{E_{j+1, n}}{B_{j+1, n} E_{j+1, n}}, \quad j = 1, 2, \ldots, n \]  

(17)

where \( E_{j+1, n} > 0 \) for all \( j, \)

we have, for \( j = n, \)

\[ p_n \leq \frac{E_{n+1}}{E_{n+1}}. \]

(18)

which is, in effect, redundant, if assumption (A7) holds, by which \( E_{n+1}/E_{n+1} \geq 1 \) necessarily.

Let us write

\[ R_{j, n} = B_{\theta, j}/B_{j, n} \].
and
\[ e_j = E_{jj}/E_{jj}. \]

Then (17) can be written as
\[ p_j \leq R_{j+1} e_{jj}. \]  
(18)

However, as \( R_{j,k} \) is monotonically non-increasing in \( k \) for fixed \( j \), condition (18) is equivalent to
\[ R_{2,j} e_{jj} \leq \lim_{k \to \infty} R_{2,k} = R, \]
(19)
as (17) must hold for all \( n > j \).

Let us examine the infinite product
\[ R = \prod_{k=2}^{\infty} \frac{1}{(1 - c_k) d_k}, \]
where
\[ c_k = \frac{1 - a_k +\gamma_k}{a_k^2 p_k + q_k} = d_k(1 - p_k)/(1 - d_k p_k), \]
with \( d_k = 1 - a_k^2 \).

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
\[ p_k = d_{k+1}/d_k, \]
\[ R = 1 - d_2 = a_2, \]
\[ R_{2,j} = (1 - d_2)/(1 - d_j p_j), \]
so that condition (19) becomes equivalent to
\[ p_j \leq e_{jj}(1 - d_j p_j). \]
(20)
Obviously, a sufficient condition for (19) to hold is, therefore,
\[ d_{k+1} \leq d_k e_{jj}/(1 + d_j e_{jj}). \]  
(22)
All the major conclusions arrived at in this section can be formally stated as follows:

**Theorem 4.1**

Let \( \{\theta_h\} \) and \( \{\tilde{\theta}_j\} \) be sequences of estimates defined by equations (5) and (11) respectively. Let \( D(\cdot) \) be as defined in equation (4).

If

(P1) \( f: R^q \to R^q \) is such that
\[ f_i(\theta) \geq f_i(0), i = 1, 2, \ldots, q \]

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

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

(P4) \( p_n = (1 - a_n^2, 1)/(1 - a_n^2) \),
where \( \tilde{a}_n = 1 - a_n \)

(P5) \( \tilde{a}_n^2 > \left[(1 - e_k - e_\gamma a_n^2)\right] \) 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
\[ D(\theta_h) < D(\tilde{\theta}_j). \]

**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 \( \lambda_n \)

In this section we shall show that it is possible to obtain certain approximations to the zone-controlling parameter \( \lambda_n \), 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

\[
\mathbf{\mu}_k = \sum_{j=1}^{m} \delta_{kj} \mathbf{\mu}_j
\]

and dispersion matrix

\[
\mathbf{\Sigma}_k = \sum_{j=1}^{m} \delta_{kj} \mathbf{\Sigma}_j,
\]

where

\[
\delta_{kj} = \begin{cases} 1 & \text{if } k = j, \\ 0 & \text{otherwise}. \end{cases}
\]

Let \( \hat{\mathbf{\mu}}_k(n) \) be as in Ref. (6), that is, let \( \hat{\mathbf{\mu}}_k(n) = \sum_{j=1}^{n} \hat{\mu}_{kj}(n) \mathbf{\mu}_j \), where

\[
\hat{\mu}_{kj}(n) = \hat{P}(A_k(n) | \mathbf{X}_j, \hat{w} = k, w = j) \hat{\sigma}_{kj}.
\]

(L2) \( \hat{\mathbf{\mu}}_k(n) \) can be approximated by \( \hat{\mathbf{\mu}}_k \).

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 \( X(k) \), i.e. the feature vector having the "given" label \( k \) (as opposed to the true label), is \( N \)-variate normal with mean vector

\[
\mathbf{\mu}_k = \mathbf{\mu}_k
\]

and dispersion matrix

\[
\mathbf{\Sigma}_k = \mathbf{\Sigma}_k,
\]

(L2)' \( \hat{\mathbf{\mu}}_k \) is equal to \( \hat{\mathbf{\mu}}_k(n) \) (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 \( \hat{\mathbf{\mu}}_k \) and \( \hat{\mathbf{\mu}}_k(n) \) 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

\[
\sigma_n = 1 /
\]

and

\[
\mathbf{B}_n^{-1} = (1/n) \sum_{j=1}^{n} (X_j - \hat{\mathbf{\mu}}_n)(X_j - \hat{\mathbf{\mu}}_n).
\]

Then for \( n > N \), a large-sample approximation to \( \lambda_n \) is given by

\[
\lambda_n^2 = \frac{n(n + 1)(n - 1)}{n(n - 1)} [\mu_p(1 - u_p),
\]

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

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

and

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

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

\[
\hat{\mathbf{\mu}}_n \rightarrow \mathbf{\mu}_n^0
\]

and

\[
\hat{\mathbf{\mu}}_n - \hat{\mathbf{\mu}}_k(n) \rightarrow 0,
\]

implying that

\[
(\hat{\mathbf{\mu}} - \hat{\mathbf{\mu}}) - (\hat{\mathbf{\mu}}_n - \hat{\mathbf{\mu}}_k(n)) \rightarrow 0.
\]

Also, as \( \sigma_n = 1/n \), it follows that \( \hat{\mathbf{\mu}}_n \) is nothing but the arithmetic mean of the \( n \) observations \( X_1, X_2, \ldots, X_n \), so that from well-known results of statistical sampling theory, it follows, on account of (L1), that

\[
\hat{\mathbf{\mu}}_n \sim N(\mathbf{\mu}_n, 1/n) \mathbf{\Sigma}_n.
\]

Relations (23) and (24) together imply that

\[
\sqrt{n} \hat{\mathbf{\mu}}_n - \hat{\mathbf{\mu}}_k(n) \sim N(0, \mathbf{\Sigma}_n),
\]

where the notation \( \rightarrow \) denotes convergence in distribution or Law; \( \sim \) denotes "is distributed as" and \( N(\mathbf{\mu}, \ldots) \) is the \( N \)-variate normal variable.

By (A5), \( X_{n+1}^k \) is independent of \( X_1^k, X_2^k, \ldots, X_n^k \) and hence of \( \sqrt{n} \hat{\mathbf{\mu}}_n - \hat{\mathbf{\mu}}_k(n) \), so that

\[
[1 + 1/n]^{-1}(X_{n+1} - (\hat{\mathbf{\mu}}_n - \hat{\mathbf{\mu}}_k(n))) \rightarrow N(\mathbf{\mu}_n, \mathbf{\Sigma}_n).
\]

Also, for \( i < n + 1 \),

\[
[1 - 1/n]^{-1}(X_i - (\hat{\mathbf{\mu}}_n - \hat{\mathbf{\mu}}_k(n))) \rightarrow N(\mathbf{\mu}_n, \mathbf{\Sigma}_n),
\]

since, by (23), \( \text{cov}(X_i, \hat{\mathbf{\mu}}_n) - \text{cov}(X_i, \hat{\mathbf{\mu}}_k(n)) \rightarrow 0 \).

Also it can easily be observed that

\[
E[(X_i - \hat{\mathbf{\mu}}_k) - (\hat{\mathbf{\mu}}_n - \hat{\mathbf{\mu}}_k(n))] = (1 - 1/n) \mathbf{\Sigma}_k \text{ if } i = j,
\]

and

\[
E[(X_i) - (\hat{\mathbf{\mu}}_n - \hat{\mathbf{\mu}}_k(n))] = - (1/n) \mathbf{\Sigma}_k \text{ if } i \neq j.
\]
Applying (L.2) to (26) and (27) gives

\[ \sqrt{[n/(n + 1)](X_{i+1} - \mu_n)^T \Sigma_n^{-1}(X_i - \mu_n)} \sim N(0, \Sigma_n) \]  

(29)

and, for \( i < n + 1 \),

\[ \sqrt{[n/(n + 1)](X_i - \mu_n)^T \Sigma_n^{-1}(X_i - \mu_n)} \sim W_n(\Sigma_n | n - 1). \]

The relation (28) implies that

\[ \left( \frac{n}{n(n-1)} \right) \sum_{i=1}^{n} E[(X_i - \mu_n)(X_i - \mu_n)^T] \sim W_n(\Sigma_n | n - 1). \]

(30)

The relation (29) and (31) together imply that

\[ \left( \frac{n}{n(n-1)} \right) \sum_{i=1}^{n} (X_i - \mu_n)(X_i - \mu_n)^T \sim W_n(\Sigma_n | n - 1). \]

(31)

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

\[ \left( \frac{k - N + 1}{N} \right) \frac{1}{N} \sum \frac{1}{n(k - 1)} \sim \text{Hotelling}(N, N - k + 1); \]

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

\[ I_n(m, n) = \int_0^x x^{m-1}(1 - x)^{n-1} dx \]

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

(2) These approximations of \( \lambda_n \) 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 \( \lambda_n \), as compared with the methods used earlier: it 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 \( \lambda_n \) 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 \( \lambda_n \), the GGA is applied to an artificially generated data set for a two-feature three-class pattern recognition problem, the values of \( \lambda_n \) 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 \( N(\mu, \Sigma) \) where

\[ \mu = (\mu_x, \mu_y)^T \quad \text{and} \quad \Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix} \]

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

**Lemma 5.1.**

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

\[ \mu = (\mu_x, \mu_y)^T \]

and dispersion matrix

\[ \Sigma = \begin{bmatrix} \sigma_x^2 & \rho \sigma_x \sigma_y \\ \rho \sigma_x \sigma_y & \sigma_y^2 \end{bmatrix}, \]

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

\[ E(y|x) = \mu_y + \frac{\sigma_y}{\sigma_x}(x - \mu_x) \]

and

\[ \text{var}(y|x) = \sigma_y^2 \left(1 - \rho^2 \right) \]

Thus, we have

\[ x = \mu_x + \tau_x \sigma_x \gamma_x \]

and

\[ y = \mu_y + \tau_y \sigma_y \gamma_y. \]

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 \((a_{ij})\)-matrix:

\[ (a_{ij}) = \begin{bmatrix} 17 & 1 & 2 \\ 1 & 16 & 3 \\ 2 & 3 & 15 \end{bmatrix} \]
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 $\lambda_n$ 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

$$MSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\mu_i - \mu^*)^2}$$

where $\mu^*$ 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,

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 $E(x'x)$ rather than $E(x - \mu)(x - \mu)'$. 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 $\lambda_n$ 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 $\lambda_n$ for a certain class of recursive parameter learning algorithms (GGA) 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 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 $X_i$ from the current estimate $\bar{\mu}_{n+1}$ of the mean vector and compare it with some threshold value $\lambda_n$. 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 $\lambda_n$. Making certain assumptions about the normality of the unconditional distribution of the feature vector, we have obtained estimates for $\lambda_n$ as

$$\lambda_n = \sqrt{n(n + 1)/(n - 1)} \mu_p(1 - u_p),$$

where $c = N/(n - N)$,

$N$ is the dimension of the feature vector,

$n$ is the iteration number,

$\mu_p$ is the lower p-percentage point of the beta distribution with degrees of freedom $N/2$ and $(n - N)/2$, so that
p = \left[ B(N/2, (n - N)/2) \right]^{-1} \frac{\Gamma(n/2)}{\Gamma(N/2) \Gamma(n - N)/2} \frac{u^{(n-2)/2} (1-u)^{N/2 - 1}}{u^{1/2} (1-v)^{(n-N)/2 - 1}} \, du

B(N/2, (n - N)/2) = \frac{\Gamma(N/2)}{\Gamma(n/2) \Gamma(n - N)/2} \int_0^1 u^{(n-2)/2} (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 \( p \) is a simple matter. These approximations to the \( a_k \) 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

<table>
<thead>
<tr>
<th>Sample no.</th>
<th>Training sample</th>
<th>True class</th>
<th>Distance</th>
<th>Update?</th>
<th>Euclidean distances from &quot;true&quot; sample values of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GGA-estimates of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>non-GGA estimates of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mean vector</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4.30</td>
<td>8.51</td>
<td>3</td>
<td></td>
<td>2.041</td>
</tr>
<tr>
<td>2</td>
<td>6.62</td>
<td>10.81</td>
<td>3</td>
<td></td>
<td>0.779</td>
</tr>
<tr>
<td>3</td>
<td>5.50</td>
<td>10.34</td>
<td>2</td>
<td>3.03</td>
<td>0.587</td>
</tr>
<tr>
<td>4</td>
<td>6.57</td>
<td>12.69</td>
<td>3</td>
<td>2.22</td>
<td>0.606</td>
</tr>
<tr>
<td>5</td>
<td>4.95</td>
<td>15.46</td>
<td>3</td>
<td>3.02</td>
<td>0.606</td>
</tr>
<tr>
<td>6</td>
<td>3.59</td>
<td>11.43</td>
<td>3</td>
<td>3.28</td>
<td>0.254</td>
</tr>
<tr>
<td>7</td>
<td>4.56</td>
<td>11.88</td>
<td>3</td>
<td>1.74</td>
<td>0.408</td>
</tr>
<tr>
<td>8</td>
<td>4.72</td>
<td>8.25</td>
<td>3</td>
<td>8.21</td>
<td>0.408</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population values</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean vector</td>
<td>10.000</td>
<td>15.000</td>
<td>5.000</td>
</tr>
<tr>
<td>Covariance matrix</td>
<td>103.000</td>
<td>152.000</td>
<td>29.000</td>
</tr>
<tr>
<td>(uncorrected)</td>
<td>152.000</td>
<td>253.000</td>
<td>25.000</td>
</tr>
<tr>
<td></td>
<td>29.000</td>
<td>25.000</td>
<td>49.000</td>
</tr>
<tr>
<td></td>
<td>100.00</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>True sample estimates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean vector</td>
<td>10.747</td>
<td>14.512</td>
<td>4.516</td>
</tr>
<tr>
<td>Covariance matrix</td>
<td>117.200</td>
<td>156.605</td>
<td>23.536</td>
</tr>
<tr>
<td>(uncorrected)</td>
<td>156.605</td>
<td>212.656</td>
<td>18.100</td>
</tr>
<tr>
<td>Initial estimates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean vector</td>
<td>9.406</td>
<td>14.720</td>
<td>5.558</td>
</tr>
<tr>
<td>Covariance matrix</td>
<td>88.471</td>
<td>138.456</td>
<td>30.891</td>
</tr>
<tr>
<td>(uncorrected)</td>
<td>138.456</td>
<td>216.682</td>
<td>25.233</td>
</tr>
<tr>
<td>Final GGA estimates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean vector</td>
<td>11.272</td>
<td>14.787</td>
<td>4.986</td>
</tr>
<tr>
<td>Covariance matrix</td>
<td>128.468</td>
<td>167.198</td>
<td>26.249</td>
</tr>
<tr>
<td>(uncorrected)</td>
<td>167.198</td>
<td>219.549</td>
<td>21.061</td>
</tr>
<tr>
<td>Final non-GGA estimates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean vector</td>
<td>10.112</td>
<td>13.643</td>
<td>4.691</td>
</tr>
<tr>
<td>Covariance matrix</td>
<td>106.176</td>
<td>141.853</td>
<td>25.991</td>
</tr>
<tr>
<td>(uncorrected)</td>
<td>141.853</td>
<td>192.547</td>
<td>25.748</td>
</tr>
</tbody>
</table>

The matrix $E(XX')$.}

REFERENCES


**APPENDIX**

**Proof of Lemma 4.1**

Let us 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

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

and

$$g(b_i, c_i) > g(b_{i+1}, c_{i+1}) \quad \text{whatever} \quad b \quad \text{may be},$$

so that

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

whence

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

**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

$$\prod_{k=2}^\infty (1 - r_k) = \frac{1 - b_1}{1 - b_0 c_2} \frac{1 - b_2}{1 - b_1 c_3} \cdots \frac{1 - b_n}{1 - b_{n-1} c_n}$$

$$= (1 - b_2)/(1 - b_0 c_2) \quad \text{as} \quad c_k = b_{k+1}/b_k \quad \to (1 - b_2) \quad \text{as} \quad n \to \infty,$$

as

$$1 \geq 1 - b_0 c_n \geq 1 - b_n \to 1 \quad \text{as} \quad n \to \infty,$$

implying that

$$1 - b_0 c_n \to 1 \quad \text{as} \quad n \to \infty.$$

**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 Physics, 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-doctoral 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, College Park, and to lecture at U.S. universities and laboratories. He was also a Guest Lecturer (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 Production 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).