COMPUTING THE BHATTACHARYYA ERROR BOUND IN CLASSIFIERS USING DIRECT BIAS CORRECTION BOOTSTRAP METHODS

B. John Oommen[1] and Qun Wang
School of Computer Science
Carleton University
Ottawa, Canada, K1S 5B6

ABSTRACT

The Bhattacharyya Bound is a measurement of the error rate of a classifier. If the distributions of the classes are independent Normal distributions, and their parameters are known, the Bhattacharyya Bound can be calculated explicitly. On the other hand, if the parameters of the distributions are unknown this bound has to be estimated. Both the theory and simulation results indicate that the estimator of the Bhattacharyya Bound given by traditional methods is seriously biased especially when the training sample size is small. By applying the bootstrap technique to the problem of estimating the Bhattacharyya Bound, we introduce several bootstrap schemes for this purpose. The results of the simulations prove that the bootstrap technique works very successfully, and dramatically reduces the bias of the estimate.

I. INTRODUCTION

I.1 Bhattacharrya Bound

In a statistical pattern recognition problem, a random vector $\underline{\mathbf{X}}$, called a pattern, is composed of a pair

$$\underline{\mathbf{X}} = (\underline{\mathbf{V}}, \omega), \qquad \text{(I.1)}$$

where $\underline{\mathbf{V}}$ is a d-dimensional vector called the feature vector, and ω is a variable representing a class. Generally, a feature vector $\underline{\mathbf{V}}$ can take a continuous value in a d-dimensional space and a class variable ω can take one of a finite set of values. In this paper, we consider only the case when there are two class cases, i.e. $\omega \in \{1, 2\}$. The *a priori* probability of the two classes, $\mathbf{P}_1$ and $\mathbf{P}_2$ are assumed known and equal.

The conditional density of a feature vector $\underline{\mathbf{V}}$ given ω is denoted as $\mathbf{p}_\omega(\underline{\mathbf{v}})$, $\omega = 1, 2$. It is well known that the posterior probability of ω given $\underline{\mathbf{V}}$ is

$$\mathbf{q}_\omega(\underline{\mathbf{v}}) = \frac{\mathbf{P}_w \mathbf{p}_w(\underline{\mathbf{v}})}{\mathbf{p}(\underline{\mathbf{x}})}, \quad \omega = 1, 2 \qquad \text{(I.2)}$$

where $\mathbf{p}(\underline{\mathbf{x}})$ is the mixed density function and is a constant independent of ω.

The purpose of pattern recognition is to determine whether a given feature vector $\underline{\mathbf{V}}$ belongs to class 1 or 2. In other words, the aim is to predict the value of ω from a given feature vector $\underline{\mathbf{V}}$. A decision rule based on probabilities is to maximize the posterior probability with the given feature vector $\underline{\mathbf{V}}$. Thus, according to (I.2), the classification rule, called the Bayesian Decision Rule, is:

$$\omega = 1 \text{ if } \mathbf{q}_1(\underline{\mathbf{V}}) > \mathbf{q}_2(\underline{\mathbf{V}}), \text{ or } \frac{\mathbf{p}_1(\underline{\mathbf{V}})}{\mathbf{p}_2(\underline{\mathbf{V}})} > \frac{\mathbf{P}_2}{\mathbf{P}_1};$$

$$\omega = 2 \text{ if } \mathbf{q}_1(\underline{\mathbf{V}}) < \mathbf{q}_2(\underline{\mathbf{V}}), \text{ or } \frac{\mathbf{p}_1(\underline{\mathbf{V}})}{\mathbf{p}_2(\underline{\mathbf{V}})} < \frac{\mathbf{P}_2}{\mathbf{P}_1}. \qquad \text{(I.3)}$$

The conditional error caused by rule (I.3) is :

$$\mathbf{r}(\underline{\mathbf{V}}) = \min\{\mathbf{q}_1(\underline{\mathbf{V}}), \mathbf{q}_2(\underline{\mathbf{V}})\}. \qquad \text{(I.4)}$$

The Bayesian error is the average of $\mathbf{r}(\underline{\mathbf{V}})$,

$$\varepsilon = \mathbf{E}\{\mathbf{r}(\underline{\mathbf{V}})\} = \int \mathbf{r}(\underline{\mathbf{V}})\mathbf{p}(\underline{\mathbf{V}})d\underline{\mathbf{V}}$$

$$= \int \min\{\mathbf{P}_1\mathbf{p}_1(\underline{\mathbf{V}}), \mathbf{P}_2\mathbf{p}_2(\underline{\mathbf{V}})\}d\underline{\mathbf{V}}. \qquad \text{(I.5)}$$

Equation (I.5) is the crucial measurement for the performance of the decision rule. Unfortunately, it is generally very hard to directly calculate a Bayesian error even if the conditional probabilities $\mathbf{p}_1(\underline{\mathbf{V}})$ and $\mathbf{p}_2(\underline{\mathbf{V}})$ are known. Therefore, many researchers have devised upper bounds for (I.5) which are easier to calculate and estimate. Two such bounds, the Chernoff and Bhattacharyya bounds, are the most well-known ones and will be

[1] Senior Member, IEEE.

discussed in the second section. In this paper we concentrate on the Bhattacharyya bound. Thus, the third section will consider the traditional approach of estimating the Bhattacharyya bound and the theoretical properties of the estimate. We will see that traditional methods to estimate the Bhattacharyya bound perform poorly especially when the sample size is small (for example, eight). This is clear from simulation results given in detail in [OW00] and [Wa00]. The concepts and general algorithms of the bootstrap are discussed in Section IV. Although the problem of estimating the Chernoff bound is not mentioned here; it is easy to see that the approaches introduced here for Bhattacharyya bound estimations can be directly applied to the Chernoff bound.

I.2 Experimental Data Set

The data set used for all the experiments in this paper is a set of randomly generated samples, which consists of training samples for seven classes. Each class has a 2-dimensional normal distribution with covariance matrix $\Sigma = I$, and a training sample size of eight. The mean vectors of the eight classes are listed in TABLE I.1.

TABLE I.1 Expectations of the classes

CLASS	A	B	C	D
Mean	(0.0, 0.0)	(0.5, 0.0)	(1.1, 0.0)	(1.1, 0.7)
CLASS	E	F	G	
Mean	(1.1, 1.5)	(2.0, 1.5)	(3.0, 1.5)	

Only two classes are involved in each experiment: one is the class **A**, and the other is selected from the rest. Hence, there are, in total, six experimental pairs of classes, (**A**, **B**), (**A**, **C**), (**A**, **D**), (**A**, **E**), (**A**, **F**), and (**A**, **G**), where each successive pair tests classes which are increasingly distant from the other. With each class pair, an experiment repeatedly does 200 trials of simulation for any given algorithm. The results of the experiments given in this paper are the average statistics from the 200 trials. As only the case of two classes is discussed in this paper, for simplicity, we use the notation that $\underline{\mathbf{X}}_{i,\,\omega} = (\underline{\mathbf{V}}, \omega)$. So $\underline{\mathbf{X}}_{i,1}$ represents sample data from class 1, and $\underline{\mathbf{X}}_{i,2}$ represents a sample data from class 2.

II. CHERNOFF-BHATTACHARRYA BOUNDS

II.1 Chernoff Bound

For any real numbers, $\mathbf{a, b} \geq 0$, it is well known that the following inequality holds:

$$\min\{\mathbf{a}, \mathbf{b}\} \leq \mathbf{a}^{s}\,\mathbf{b}^{1-s}, \quad 0 \leq s \leq 1. \qquad \text{(II.1)}$$

Applying inequality (II.1) to (I.5), we have

$$\min\{\mathbf{P}_1\mathbf{p}_1(\underline{\mathbf{V}}), \mathbf{P}_2\mathbf{p}_2(\underline{\mathbf{V}})\}d\underline{\mathbf{V}}$$

$$\leq \mathbf{P}_1^{s}\;\mathbf{P}_2^{1-s}\quad \mathbf{p}_1^{s}(\underline{\mathbf{V}}), \mathbf{p}_2^{1-s}(\underline{\mathbf{V}})d\underline{\mathbf{V}}.$$

The right side of the above inequality

$$\varepsilon_u = \mathbf{P}_1^{s}\;\mathbf{P}_2^{1-s}\quad \mathbf{p}_1^{s}(\underline{\mathbf{V}}), \mathbf{p}_2^{1-s}(\underline{\mathbf{V}})d\underline{\mathbf{V}}. \qquad \text{(II.2)}$$

is called the Chernoff bound. The optimum **s** is the value that minimizes the value of ε_u. We consider the case when the conditional density functions are normal distributions

$$\mathbf{p}_j(\underline{\mathbf{V}}) = \mathbf{N}_j(\underline{\mathbf{M}}_j, \Sigma_j), \qquad j = 1, 2\ .$$

The integration part of (II.1) can be expressed as

$$\mathbf{p}_1^{s}(\underline{\mathbf{V}}), \mathbf{p}_2^{1-s}(\underline{\mathbf{V}})d\underline{\mathbf{V}} = \mathbf{e}^{-\mu(s)}, \text{ where,} \qquad \text{(II.3)}$$

$$\mu(\mathbf{s}) = \frac{\mathbf{s}(1-\mathbf{s})}{2}(\underline{\mathbf{M}}_1 - \underline{\mathbf{M}}_2)^{T}\,[\mathbf{s}\,\Sigma_1 + (1-\mathbf{s})\,\Sigma_2]^{-1}$$

$$(\underline{\mathbf{M}}_1 - \underline{\mathbf{M}}_2) + \frac{1}{2}\ln\frac{|\,\mathbf{s}\,\Sigma_1 + (1-\mathbf{s})\,\Sigma_2\,|}{|\,\Sigma_1\,|^{s}\,|\,\Sigma_2\,|^{1-s}}. \qquad \text{(II.4)}$$

The quantity $\mu(\mathbf{s})$ is called the Chernoff distance. It is obvious that the optimum value of **s** will maximize the value of $\mu(\mathbf{s})$, which can be obtained by studying the variation of $\mu(\mathbf{s})$ for various values of **s** for the given $\underline{\mathbf{M}}_j$ and Σ_j (j = 1, 2).

II.2 Bhattacharyya Bound

The Bhattacharyya bound results when the Chernoff bound is simplified by selecting the value $\mathbf{s} = 1/2$. In such a case, (II.2) becomes

$$\varepsilon_u = \sqrt{\mathbf{P}_1\mathbf{P}_2}\quad \sqrt{\mathbf{p}_1(\underline{\mathbf{V}})\mathbf{p}_2(\underline{\mathbf{V}})}\,d\underline{\mathbf{V}}$$

$$= \sqrt{\mathbf{P}_1\mathbf{P}_2}\;\mathbf{e}^{-\mu(1/2)}. \qquad \text{(II.5)}$$

The upper bound given by (II.5) is called the Bhattacharyya bound. In our case, it is assumed that $\mathbf{P}_1 = \mathbf{P}_2 = ½$ therefore, the Bhattacharyya bound will take a form of:

$$\varepsilon_u = \frac{1}{2}\quad \sqrt{\mathbf{p}_1(\underline{\mathbf{V}})\mathbf{p}_2(\underline{\mathbf{V}})}\,d\underline{\mathbf{V}} = \frac{1}{2}\mathbf{e}^{-\mu(1/2)}. \qquad \text{(II.6)}$$

Correspondingly, the Bhattacharyya distance is:

$$\mu(1/2) = \frac{1}{8}(\underline{\mathbf{M}}_1 - \underline{\mathbf{M}}_2)^{T}\left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1}$$

$$(\underline{\mathbf{M}}_1 - \underline{\mathbf{M}}_2) + \frac{1}{2}\ln\frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}}. \qquad \text{(II.7)}$$

III. ESTIMATING BHATTACHARYYA BOUNDS

Let $\mathbf{X} = \{\underline{\mathbf{X}}_{1,1}, \underline{\mathbf{X}}_{2,1}, \ldots, \underline{\mathbf{X}}_{m,1}, \underline{\mathbf{X}}_{1,2}, \ldots, \underline{\mathbf{X}}_{n,2}\}$ be the training sample: the first *m* sample points $\underline{\mathbf{X}}_{1,1}, \underline{\mathbf{X}}_{2,1}, \ldots, \underline{\mathbf{X}}_{m,1}$ belongs to class 1 with a normal conditional distribution

$$\underline{\mathbf{X}}_{1,1}, \ldots, \underline{\mathbf{X}}_{m,1} \sim \mathbf{N}_1(\underline{\mathbf{M}}_1, \Sigma_1), \qquad \text{(III.1)}$$

and the other *n* sample data $\underline{\mathbf{X}}_{1,2}, \ldots, \underline{\mathbf{X}}_{n,2}$ belongs to class 2 with a normal conditional distribution

$$\underline{\mathbf{X}}_{1,2}, \ldots, \underline{\mathbf{X}}_{n,2} \sim \mathbf{N}_2(\underline{\mathbf{M}}_2, \Sigma_2). \qquad \text{(III.2)}$$

The maximum likelihood estimates of the means $\underline{\mathbf{M}}_j$ and covariances Σ_j (j = 1, 2) [Fu92] are given respectively by

$$\hat{\underline{\mathbf{M}}}_1,\ \hat{\Sigma}_1,\ \hat{\underline{\mathbf{M}}}_2 \text{ and } \hat{\Sigma}_2. \qquad \text{(III.3)}$$

It is well known that the estimates given by (III.3) have good properties, and are the optimized estimates of the parameters $\underline{\mathbf{M}}_j$ and Σ_j (j = 1, 2). Based on (III.3), an estimate of the Bhattacharyya bound can be calculated by the following steps:

1. Replace the parameters in (II.7) with their estimates given by (III.3) to get an estimate of Bhattacharyya distance, say $\hat{\mu}(1/2)$.
2. Replace $\mu(1/2)$ in (II.6) with $\hat{\mu}(1/2)$ to get an estimate of the Bhattacharyya bound.

We refer to the estimating method described above as the *Direct Estimating approach* or the *General approach*. The unanswered question is one of determining how good the estimate of the General approach will be? The answer to this question is by no means trivial, because the Bhattacharyya distance and bound are fairly complex functions of the parameters $\underline{\mathbf{M}}_j$ and Σ_j.

To clarify issues, we present a brief discussion on the estimate properties of a function of the parameters. Let $\underline{\theta} = (\theta_1, \theta_2, \ldots, \theta_t)^T$ be a parameter vector, whose estimate are $\hat{\underline{\theta}} = (\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_t)^T$, and $\mathbf{f} = \mathbf{f}(\underline{\theta})$ be a function of $\underline{\theta}$. Using the General approach, an estimator of $\hat{\mathbf{f}} = \mathbf{f}(\hat{\underline{\theta}})$ can be obtained. Assuming that $\underline{\theta}$ and $\hat{\underline{\theta}}$ are close enough, a Taylor series can be used by expanding $\hat{\mathbf{f}} = \mathbf{f}(\hat{\underline{\theta}})$ up to the second order terms,

$$\hat{\mathbf{f}} = \mathbf{f}(\hat{\underline{\theta}}) \cong \mathbf{f}(\underline{\theta}) + \frac{\partial \mathbf{f}}{\partial \underline{\theta}}^T \Delta\underline{\theta} + \frac{1}{2}\mathrm{tr}\left(\frac{\partial^2 \mathbf{f}}{\partial \underline{\theta}^2}\Delta\underline{\theta}\Delta\theta^T\right), \qquad \text{(III.4)}$$

where $\Delta\underline{\theta} = \hat{\underline{\theta}} - \underline{\theta}$. If $\hat{\underline{\theta}}$ is an unbiased estimate of $\underline{\theta}$, $\mathbf{E}(\Delta\underline{\theta}) = 0$. Thus,

$$\mathbf{E}(\hat{\mathbf{f}}) \cong \mathbf{f}(\underline{\theta}) + \frac{1}{2}\mathrm{tr}\left(\frac{\partial^2 \mathbf{f}}{\partial \underline{\theta}^2}\mathbf{E}\{\Delta\underline{\theta}\Delta\theta^T\}\right). \qquad \text{(III.5)}$$

It is clear to see that $\hat{\mathbf{f}}$ is generally a biased estimator of **f**. In our case

$$\underline{\theta} = (\underline{\mathbf{M}}_1^T, \underline{\mathbf{M}}_2^T, \mathrm{svec}(\Sigma_1)^T, \mathrm{svec}(\Sigma_2)^T)^T,$$

where svec(**A**) represents a vector consisting of all different components of a symmetric matrix $\underline{\mathbf{A}} = (\mathbf{a}_{i,j})_{n\times n}$, $\mathrm{svec}(\underline{\mathbf{A}}) = (\mathbf{a}_{11}, \ldots, \mathbf{a}_{n1}, \mathbf{a}_{22}, \ldots, \mathbf{a}_{n2}, \ldots, \mathbf{a}_{nn})$, and its estimator

$$\hat{\underline{\theta}} = (\underline{\mathbf{M}}_1^T, \underline{\mathbf{M}}_2^T, \mathrm{svec}(\Sigma_1)^T, \mathrm{svec}(\Sigma_2)^T)^T$$

given in (III.3) is unbiased. For the Bhattacharyya distance, the second term on the right side in (III.5) is extremely complicated (please refer to Fukunaga's book [Fu92] for more details). It is thus apparent that the estimate of the Bhattacharyya distance deduced from the General approach is biased, as is the Bhattacharyya bound.

IV. THE BOOTSTRAP TECHNIQUE

IV.1 Basic Bootstrap

The bootstrap technique was first introduced by Efron in the late 1970's [Ef79]. The basic strategy of bootstrap is based on resampling and simulation.

Let $\mathbf{X} = \{\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, \ldots, \underline{\mathbf{X}}_n\}$ be an i.i.d. *d*-dimensional sample from an unknown distribution $\mathbf{F}$. We consider an arbitrary functional of $\mathbf{F}$, $\theta = \theta(\mathbf{F})$, which for example, could be an expectation, a quantile, a variance, etc. The quantity θ is estimated by a functional of the empirical distribution $\hat{\mathbf{F}}$, $\hat{\theta} = \theta(\hat{\mathbf{F}})$, where

$$\hat{\mathbf{F}} : \text{mass } \frac{1}{n} \text{ at } \underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \ldots, \underline{\mathbf{x}}_n,$$

where *n* is the sample size, and $\{\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \ldots, \underline{\mathbf{x}}_n\}$ are the observed values of the sample $\{\underline{\mathbf{X}}_1, \ldots, \underline{\mathbf{X}}_n\}$.

The bias of $\hat{\theta}$ is well defined as

$$\mathbf{Bias} = \mathbf{E}_{\mathbf{F}}[\hat{\theta} - \theta] = \mathbf{E}_{\mathbf{F}}[\theta(\hat{\mathbf{F}}) - \theta(\mathbf{F})]. \quad \text{(IV.1)}$$

where "$\mathbf{E}_{\mathbf{F}}$" indicates the expectation under distribution $\mathbf{F}$. Though it is possible to calculate $\hat{\theta} = \theta(\hat{\mathbf{F}})$ having observed $\underline{\mathbf{X}}_1 = \underline{\mathbf{x}}_1, \underline{\mathbf{X}}_2 = \underline{\mathbf{x}}_2, \ldots, \underline{\mathbf{X}}_n = \underline{\mathbf{x}}_n$, it is not possible to derive the bias directly because of the fact that both θ and $\mathbf{F}$ are unknown.

From one perspective, the quantity $\hat{\theta}$ can be regarded as a simulation of θ, since we use the empirical distribution $\hat{\mathbf{F}}$ to mirror the characteristic of the unknown distribution $\mathbf{F}$ via the sampling process. Extending the idea to the bias estimation, we can use the same strategy to solve the problem. As the empirical distribution $\hat{\mathbf{F}}$ is known, it is easy to randomly generate a sample from $\hat{\mathbf{F}}$. Assuming the size of the sample generated is *n*, the sample is

$$\mathbf{X}^* = \{\underline{\mathbf{X}}_1^*, \underline{\mathbf{X}}_2^*, \ldots, \underline{\mathbf{X}}_n^*\}. \quad \text{(IV.2)}$$

Thus we have an empirical distribution $\hat{\mathbf{F}}^*$ of the empirical distribution $\hat{\mathbf{F}}$, where,

$$\hat{\mathbf{F}}^*: \text{mass } \frac{1}{n} \text{ at } \underline{\mathbf{x}}_1^*, \underline{\mathbf{x}}_2^*, \ldots, \underline{\mathbf{x}}_n^*,$$

and $\{\underline{\mathbf{x}}_1^*, \underline{\mathbf{x}}_2^*, \ldots, \underline{\mathbf{x}}_n^*\}$ are the observed values of the $\{\underline{\mathbf{X}}_1^*, \underline{\mathbf{X}}_2^*, \ldots, \underline{\mathbf{X}}_n^*\}$, and a corresponding $\hat{\theta}^* = \theta(\hat{\mathbf{F}}^*)$ is the estimate of $\hat{\theta}$. Thus an estimator of the bias is

$$\hat{\mathbf{Bias}} = \mathbf{E}^*[\hat{\theta}^* - \hat{\theta}] = \mathbf{E}^*[\theta(\hat{\mathbf{F}}^*) - \theta(\hat{\mathbf{F}})], \quad \text{(IV.3)}$$

where $\mathbf{E}^*$ is the conditional expectation of $\hat{\mathbf{F}}$ given $\{\underline{\mathbf{X}}_1^* = \underline{\mathbf{x}}_1^*, \underline{\mathbf{X}}_2^* = \underline{\mathbf{x}}_2^*, \ldots, \underline{\mathbf{X}}_n^* = \underline{\mathbf{x}}_n^*\}$. The procedure to estimate the bias described above is called the *Bootstrap* technique. A typical bootstrap procedure will take several steps :

1. Generate an an empirical distribution $\hat{\mathbf{F}}^*$ of the empirical distribution $\hat{\mathbf{F}}$;
2. Get a corresponding $\hat{\theta}^* = \theta(\hat{\mathbf{F}}^*)$ as the estimate of $\hat{\theta}$;
3. Repeat steps 1 and 2 *B* times, so as to have a set of estimates $\{\hat{\theta}_1^*, \hat{\theta}_2^*, \ldots, \hat{\theta}_B^*\}$ of $\hat{\theta}$;
4. Calculate the estimate of the bias as

$$\hat{\mathbf{Bias}} = \frac{1}{B}\sum_b \hat{\theta}_b^* - \hat{\theta}, \quad \text{(IV.4)}$$

In the interest of brevity, the details are omitted here - they can be found in [Wa00] and in the unabridged paper [OW00]. Efron suggested choosing $B = 200$ as the number of times the bootstrap resampling is repeated[E83], which is quite adequate for most of purposes.

If $\hat{\theta}$ is an estimate of θ and a bootstrap estimate of the bias (IV.4) is provided, a bootstrap estimate of θ can be obtained by correcting the bias of the original estimate as:

$$\hat{\theta}_{\text{BOOT}} = \hat{\theta} - \hat{\mathbf{Bias}} = 2\hat{\theta} - \hat{\theta}^* \quad \text{(IV.5)}$$

This strategy will be used later to estimate the Bhattacharyya bound.

IV.2 Bayesian/Random Weighting Method

As described above, the key issue of the bootstrap technique is to obtain an empirical distribution $\hat{\mathbf{F}}^*$ of the empirical distribution $\hat{\mathbf{F}}$. It is possible to generalize the sampling scheme for retrieving a bootstrap sample in the following way.

Let $\underline{\mathbf{P}}^* = (\mathbf{P}_1^*, \mathbf{P}_2^*, \ldots, \mathbf{P}_n^*)$ be any probability vector on the *n*-dimensional simplex

$$\boldsymbol{\Delta}_n = \{\underline{\mathbf{P}}^*: \mathbf{P}_i^* \geq 0, \textstyle\sum_i \mathbf{P}_i^* = 1\}, \quad \text{(IV.6)}$$

called a *resampling vector* [Ef82]. For a sample $\mathbf{X} = \{\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, \ldots, \underline{\mathbf{X}}_n\}$ a re-weighted empirical probability distribution $\hat{\mathbf{F}}^*$ is defined with a resampling vector $\underline{\mathbf{P}}^*$ as

$$\hat{\mathbf{F}}^*: \text{mass } \mathbf{P}_i^* \text{ on } \underline{\mathbf{x}}_i, \; i = 1, 2, \ldots, n, \quad \text{(IV.7)}$$

where $\{\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \ldots, \underline{\mathbf{x}}_n\}$ are the observed values of the sample $\{\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, \ldots, \underline{\mathbf{X}}_n\}$. Thus, a resampled value of $\hat{\theta}$, say $\hat{\theta}^*$, will be

$$\hat{\theta}^* = \theta(\hat{\mathbf{F}}^*(\underline{\mathbf{P}}^*)) = \theta(\underline{\mathbf{P}}^*). \quad \text{(IV.8)}$$

As before, after repeatedly generating the resampling vector $\underline{\mathbf{P}}^*$ *B* times, a sample of $\hat{\theta}$ will be obtained, say $\{\hat{\theta}_b^* : b = 1, 2, \ldots, B\}$. The basic bootstrap is, from this point of view, a specific case of (IV.7) - (IV.8) in which the resampling vector $\underline{\mathbf{P}}^*$ takes the form

$$\underline{\mathbf{P}}_i^* = n_i^* / n, \quad \text{(IV.9)}$$

where n_i^* is the number of times $\underline{\mathbf{X}}_i$ appears in a bootstrap sample. This means that $\underline{\mathbf{P}}^*$ follows a multinominal distribution,

$$\underline{\mathbf{P}}^* \sim \frac{1}{n}\text{Mult}(n, \underline{\mathbf{P}}_0), \quad \text{(IV.10)}$$

where $\underline{\mathbf{P}}_0 = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$ is a n-dimensional vector. In other words, it is possible to execute a resampling procedure by choosing another $\underline{\mathbf{P}}^*$ in the n-dimensional simplex (IV.6). These arguments lead to the Bayesian Bootstrap and Random Weighting Method introduced by Rubin [ST95] and Zhen [Zh87] respectively. We request the reader to please refer to [OW00), [ST95] [Wa00], and [Zh87] for the details of the algorithms of the Bayesian and Random Weighting Methods.

V. DIRECT BIAS CORRECTION BOOTSTRAPS

The direct bias correction schemes for the estimate of the Bhattacharyya bound directly use the bootstrap estimate of the Bhattacharyya bound to correct the bias, i.e. to apply (IV.5) given in section IV. The algorithms with the Basic Bootstrap resampling scheme is described below:

Algorithm V.1 Basic Bootstrap
Input:
(i) n : the size of the training sample ;
(ii) $\underline{x}[1, 1], \ldots, \underline{x}[n, 1]$: training samples - Class 1;
(iii) $\underline{x}[1, 2], \ldots, \underline{x}[n, 2]$: training samples Class 2;
(iv) B : repeated times of the bootstrap resampling.
Output: The Bhattacharyya bound estimate.
Method
BEGIN
ε_0 = CalcBound($\underline{x}$[1,1],…,$\underline{x}$[n,1], $\underline{x}$[1,2],…, $\underline{x}$[n,2])
$\varepsilon = 0$
For (i = 1 to B)
  For (j = 1 to n)
    m = random integer in [1, n]
    $\underline{y}$[j, 1] = $\underline{x}$[m, 1]
  End-For
  For (j = 1 to n)
    m = random integer in [1, n]
    $\underline{y}$[j, 2] = $\underline{x}$[m, 2]
  End-For
  $\varepsilon = \varepsilon$ + CalcBound($\underline{y}$[1,1],…, $\underline{y}$[n, 1] , $\underline{y}$[1, 2],…, $\underline{y}$[n, 2])
End-For
$\varepsilon = 2 \times \varepsilon_0 - \varepsilon / B$
Return ε
END Basic Bootstrap

Procedure CalcBound
Input:
(i) n : the size of the training sample ;
(ii) $\underline{z}[1, 1], \ldots, \underline{z}[n, 1]$: sample data - Class 1;
(iii) $\underline{z}[1, 2], \ldots, \underline{z}[n, 2]$: sample data Class 2;
Output: A Bhattacharyya bound estimate.
Method
BEGIN
$\mu(1/2)$ = CalcDistance($\underline{z}$[1, 1], …, $\underline{z}$[n, 1], $\underline{z}$[1, 2], …, $\underline{z}$[n, 2])
$\varepsilon = 0.5 * \text{EXP}(-\mu(1/2))$
Return ε
END Procedure CalcBound

Procedure CalcDistance
Input: As in Procedure **CalcBound.**
Output: A Bhattacharyya bound estimate.
Method
BEGIN

$$\bar{\underline{\mathbf{z}}}[1] = \frac{1}{n}\sum_{j=1}^{n} \underline{z}[j, 1]$$

$$\Sigma_1 = \frac{1}{n-1}\sum_{j=1}^{n} (\underline{z}[j, 1] - \bar{\underline{\mathbf{z}}}[1])(\underline{z}[j, 1] - \bar{\underline{\mathbf{z}}}[1])^T$$

$$\bar{\underline{\mathbf{z}}}[2] = \frac{1}{n}\sum_{j=1}^{n} \underline{z}[j, 2]$$

$$\Sigma_2 = \frac{1}{n-1}\sum_{j=1}^{n} (\underline{z}[j, 2] - \bar{\underline{\mathbf{z}}}[2])(\underline{z}[j, 2] - \bar{\underline{\mathbf{z}}}[2])^T$$

$$\mu(1/2) = \frac{1}{8}(\bar{\underline{\mathbf{z}}}[1] - \bar{\underline{\mathbf{z}}}[2])^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\bar{\underline{\mathbf{z}}}[1] - \bar{\underline{\mathbf{z}}}[2]) + \frac{1}{2}\ln \frac{|\frac{\Sigma_1 + \Sigma_2}{2}|}{\sqrt{|\Sigma_1||\Sigma_2|}}$$

Return $\mu(1/2)$
END Procedure CalcDistance

The algorithms are straightforward, but more detailed explanations of the various steps are found in [OW00] and [Wa00]. But, by changing the

technique of generating the bootstrap samples, the other two estimating algorithms for the Bhattacharyya bound, the Bayesian bootstrap and the random weighting algorithms can be obtained. They are straightforward and are omitted in the interest of brevity and can be found in [Wa00]. However, the new procedure **CalcNewDistance** is included as it will be utilized later.

Algorithm Procedure CalcNewDistance
Input: As in Procedure **CalcBound**, and a *flag* to indicate the resampling schemes.
Output: A Bhattacharyya bound estimate.
Method
BEGIN
Get_ Bootstrap_Weights (*n*, flag)

$$\underline{\bar{z}}\,[1] = \sum_{j=1}^{n} \mathbf{w}[j]\underline{\mathbf{z}}\,[j, 1]$$

$$\Sigma_1 = \sum_{j=1}^{n} \mathbf{w}[j](\underline{\mathbf{z}}\,[j, 1] - \underline{\bar{y}}\,[1])(\underline{\mathbf{z}}[j, 1] - \underline{\bar{y}}\,[1])^T$$

Get_ Bootstrap_Weights (*n*, flag)

$$\underline{\bar{z}}\,[2] = \sum_{j=1}^{n} \mathbf{w}[j]\underline{\mathbf{z}}[j, 2]$$

$$\Sigma_2 = \sum_{j=1}^{n} \mathbf{w}[j](\underline{\mathbf{z}}\,[j, 2] - \underline{\bar{z}}\,[2])(\underline{\mathbf{z}}\,[j, 2] - \underline{\bar{z}}\,[2])^T$$

$$\mu(1/2) = \frac{1}{8}(\underline{\bar{z}}\,[1] - \underline{\bar{z}}\,[2])^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\underline{\bar{z}}\,[1] - \underline{\bar{z}}\,[2]) + \frac{1}{2}\ln \frac{\left|\frac{\Sigma_1 + \Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}}$$

Return $\mu(1/2)$
END Procedure CalcNewDistance

Here, the Get_ Bootstrap_Weights procedure is used to set the resampling weights w[1], w[2], …, w[n]. The two parameters passed to it determine the dimension of the resampling vector and the scheme used to generate the resampling vector.

V.1 Simulation Results

The simulation experiments were done with the above three algorithms and the statistics for them are listed in TABLE V.1 – V.3. The results are given in terms of percentages.

TABLE V.1 Basic Bootstrap (%)

CLASS PAIR	(A,B)	(A,C)	(A,D)	(A,E)	(A,F)	(A,G)
Theoretical Value	46.97	43.58	42.48	39.63	36.58	32.88
Mean	48.34	42.16	40.53	31.84	18.52	12.64
STD	5.94	8.29	8.84	10.21	14.41	8.46
Maximum	58.68	58.84	58.44	55.33	49.52	37.17
80%	53.55	49.24	48.09	40.64	33.08	19.96
Median	49.90	42.76	41.23	32.58	19.41	11.52
20%	39.76	32.02	27.84	16.56	-0.32	2.48
Minimum	30.44	8.33	12.08	8.92	-11.00	0.02

TABLE V.2 Bayesian Bootstrap (%)

CLASS PAIR	(A,B)	(A,C)	(A,D)	(A,E)	(A,F)	(A,G)
Theoretical Value	46.97	43.58	42.48	39.63	36.58	32.88
Means	44.91	39.30	37.69	29.68	22.38	11.87
STD	5.53	7.72	8.19	9.44	9.39	7.87
Maximum	55.08	54.29	54.22	49.74	45.16	35.48
80%	50.03	45.81	44.99	37.71	30.52	18.52
Median	45.90	39.98	38.46	30.62	23.36	11.13
20%	36.64	29.56	26.53	15.42	9.07	2.30
Minimum	28.29	7.88	11.72	8.37	1.82	0.01

TABLE V.3 Random Weighting Method (%)

CLASS PAIR	(A,B)	(A,C)	(A,D)	(A,E)	(A,F)	(A,G)
Theoretical Value	46.97	43.58	42.48	39.63	36.58	32.88
Means	42.33	36.66	35.14	27.35	20.45	10.71
STD	5.32	7.39	7.88	8.92	8.77	7.20
Maximum	51.68	51.24	51.35	46.67	42.30	32.54
80%	47.11	43.09	41.97	34.85	27.62	17.18
Median	43.21	37.50	36.08	28.27	21.16	9.94
20%	34.19	27.11	24.34	13.99	8.13	2.07
Minimum	25.70	8.29	10.04	7.62	1.54	0.01

As can been seen from the tables, the bootstrap techniques does work in this case. On average, the means of the 200 estimates for all the six class pairs improves to some degree. As can be seen, the percentage of the estimates with values over the theoretical value also increased. We note, however, that there are a few disadvantages to the three algorithms. The standard deviations of the estimates are larger than that of the General approach. In the experiment of the basic bootstrap algorithm with the class pair (A, F), about 20% of the estimates are even below zero, which is quite unacceptable. Of course, a restriction to the algorithm could be added to discard negative estimate values. A simple way of doing this is to just reject an estimate when a negative estimate value is reported, and to re-

draw the bootstrap samples and calculate the estimate again. This procedure would be repeated until a nonnegative estimate is produced. Comparatively, the Bayesian bootstrap and random weighting method algorithms have smaller standard deviations and no negative estimate values. More detailed experimental results can be found in [OW00, Wa00] and are omitted here.

REFERENCES

[CMN85] Chernick, M. R., Murthy, V. K. and Nealy, C.D. (1985), "Application of bootstrap and other resampling techniques: evaluation of classifier performance", *Pattern Recognition Letters*, 3, pp.167 - 178.

[CMN86] Chernick, M. R., Murthy, V. K. and Nealy, C. D. (1986), "Correction to 'application of bootstrap and other resampling techniques: evaluation of classifier performance'", *Pattern Recognition Letters*, 4, pp.133 - 142.

[DH92] Davison, A. C. and Hall, P. (1992), "On the bias and variability of bootstrap and cross-validation estimates of error rate in discrimination problems", *Biometrika*, 79, pp.279 - 284.

[DH97] Davison, A. C. and Hinkley, D. V. (1997), *Bootstrap methods and Their Application*, Cambridge University Press, Cambridge.

[Ef79] Efron, B. (1979), "Bootstrap Method: Another Look at the Jackknife", *Annals of Statistics*, 7, pp.1 - 26.

[Ef82] Efron, B. (1982), "*The Jackknife, the Bootstrap and Other Resampling Plans*, *CBMS-NSF, Regional Conference Series in Applied Mathematics*, Society for Industrial and Applied Mathematics.

[Ef83] Efron, B. (1983), "Estimating the error rate of a prediction rule: improvment on cross-validation", *Journal of the American Statistical Association*, 78, pp.316 - 331.

[Ef86] Efron, B. (1986), "How biased is the apparent error rate of a prediction rule?", *Journal of the American Statistical Association*, 81, pp.461 - 470.

[ET93] Efron, B. and Tibshirani, R. J. (1993), *An introduction to the Bootstrap*, Chapman & Hall, Inc.

[Fu90] Fukunaga, K. (1990), *Introduction to Statistical Pattern Recognition, Second Edition*, Academic Press, Inc.

[OW00] Oommen, B.J. and Wang, Q. "Bootstrap Estimation of a Classifier's Bhattacharyya Error Bound". In Preparation.

[ST95] Shao, J. and Tu, D., (1995), *The Jackknife and Bootstrap*, Springer-Verlag, New York.

[Wa00] Wang, Q. (2000), *Bootstrap Techniques for Statistical Pattern Recognition*. Master Thesis, School of Computer Science, Carleton University, Ottawa, Canada.

[Zh87] Zhen, Z. (1987), "Random weighting methods", *Acta Math. Appl. Sinica*, 10, pp.247 - 253.