ANALYSIS OF DECISION BOUNDARIES IN LINEARLY COMBINED NEURAL CLASSIFIERS

Kagan Tumer and Joydeep Ghosh
Department of Electrical and Computer Engineering,
University of Texas, Austin, TX 78712-1084
E-mail: kagan@pine.ece.utexas.edu
ghosh@pine.ece.utexas.edu

Abstract: Combining or integrating the outputs of several pattern classifiers has led to improved performance in a multitude of applications. This paper provides an analytical framework to quantify the improvements in classification results due to combining. We show that combining networks linearly in output space reduces the variance of the actual decision region boundaries around the optimum boundary. This result is valid under the assumption that the *a posteriori* probability distributions for each class are locally monotonic around the Bayes optimum boundary. In the absence of classifier bias, the error is shown to be proportional to the boundary variance, resulting in a simple expression for error rate improvements. In the presence of bias, the error reduction, expressed in terms of a bias reduction factor, is shown to be less than or equal to the reduction obtained in the absence of bias. The analysis presented here facilitates the understanding of the relationships among error rates, classifier boundary distributions, and combining in output space.

Keywords: combining, decision boundary, neural networks, pattern classification, hybrid networks, variance reduction.

1 Introduction

Training a parametric classifier involves the use of a *training* set of data with known classification to estimate or "learn" the parameters of the chosen model. A *test* set, consisting of patterns previously unseen by the classifier, is then used to determine the classification performance. This ability to meaningfully respond to novel patterns, or generalize, is an important aspect of a classifier system and in essence, the true gauge of performance [1, 2]. Given infinite training data, consistent classifiers approximate the Bayesian decision boundaries to arbitrary precision, therefore providing similar generalizations [3]. However, often only a limited portion of the pattern space is available or observable [4, 5]. Given a finite and noisy data set, different classifiers typically provide different generalizations (or different decision boundaries) [6]. For example, when classification is performed using a multilayered, feed-forward artificial neural network, different weight initializations, or different architectures (number of hidden units, hidden layers, node activation functions etc.) result in differences in performance. It is therefore necessary to train a multitude of networks when approaching a classification problem to ensure that a good model/parameter set is found. However, selecting such a classifier is not necessarily the ideal choice, since potentially valuable information may be wasted by discarding the results of less-successful classifiers [7].

In order to avoid the potential loss of information through selecting only one classifier, the outputs of all the available classifiers can be pooled before a decision is made. This approach is particularly useful for difficult problems, such as those that involve a large amount of noise, limited number of training data, or unusually high dimensional patterns. The overall architecture of a combiner is shown in Figure 1. The output of an individual classifier using a single feature set is given by f^{ind}. Multiple classifiers, possibly trained on different feature sets, provide the combined output f^{comb}.

There are several methods of combining that have proved effective in improving the classifier performance. Simple averaging of the outputs of individual classifiers has been suggested by different researchers as an alternative to selecting the best network [8, 9, 10]. Methods that select the class with the highest activation value, use the geometric mean or entropy based criteria, or perform a majority vote have been analyzed [11, 12, 13]. Methods based on confidence factors obtained through the theory evidence have also been studied [14]. Weighted averaging has been proposed, along with different methods of computing the proper classifier weights [8, 9]. A survey of leading combining techniques, along with experimental results is given in [11, 12].

Combining techniques such as majority voting can generally be applied to any type of classifier, while others rely on specific outputs, or specific interpretations of the output. For example, the confidence factors method relies on the interpretation of the outputs as the belief that the patterns belong to a given class [15, 10]. Averaging, on the other

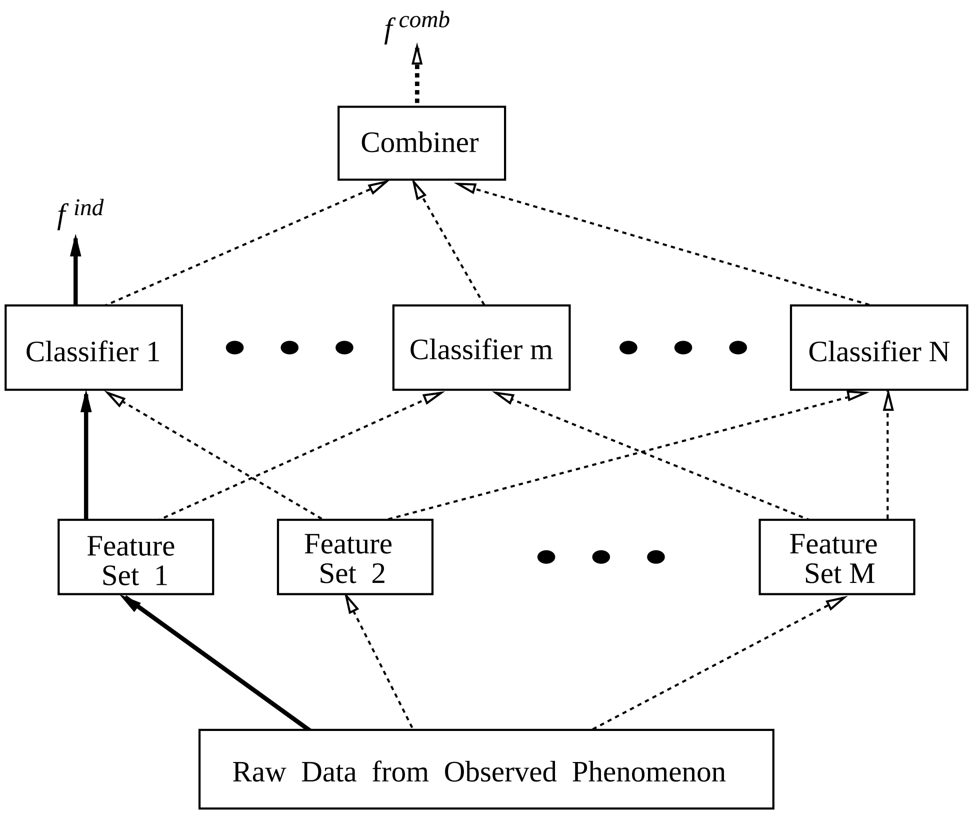

Figure 1: Combining Strategy. The solid lines leading to f^{ind} represent the decision of a specific classifier, while the dashed lines lead to f^{comb}, the output of the combiner.

hand, uses the result that the outputs of parametric classifiers that are trained to minimize a cross-entropy or mean square error (MSE) function, given "one-of-n" desired outputs, approximate the *a posteriori* probability densities of each class [16]. In particular, the MSE is shown to be equivalent to

$$MSE = K_1 + \sum_i \int_x D_i(x)\,(p(C_i|x) - f_i(x))^2\,dx$$

where K_1 and $D_i(x)$ depend on the class distribution only, $f_i(x)$ is the output of the node representing class i given an output x, $p(C_i|x)$ denotes the posterior probability and the summation is over all classes. Thus minimizing the (expected) MSE corresponds to a weighted least squares fit of the network outputs to the posterior probabilities [16, 17].

For regression (or function approximation) problems, recent work analyzing the effect of linear combining is available [18, 19]. However, despite the increasing body of experimental results showing classification improvements due to combining, there has been no analytical study that can quantify the achievable gains. In this paper we analytically study the effect of combining in output space with a focus on the relationship between decision boundary distributions and error rates. Our objective is to provide an analysis encapsulating the most commonly used combining strategy, namely, averaging in output space. The analysis focuses on boundary distributions, and how the parameters of that distribution influence the error rates. Ultimately, our goal is to both quantify and predict the error reductions due to combining.

2 Class Boundary Analysis in Absence of Bias

As mentioned above, the outputs of certain classifiers are expected to approximate the corresponding *a posteriori* class probabilities if they are reasonably well trained. Thus the decision boundaries obtained by such classifiers are expected to be close to Bayesian decision boundaries. Moreover, these boundaries will occur in regions where the number of training samples belonging to the two most locally dominant classes are comparable.

We will focus our analysis to network performance around the decision boundaries. Consider the boundary between class i and j. First, let us express the output response of the i^{th} unit of a *one-of-n* classifier network to a given input x as[1]:

$$f_i(x) = p_i(x) + \epsilon_i(x)\,, \tag{1}$$

where $p_i(x)$ is the *a posteriori* probability distribution of the i^{th} class given input x, and $\epsilon_i(x)$ is the error associated with the i^{th} output[2]. The following analysis is for scalar x, for simplicity. However, the analysis can be readily extended for multi-dimensional inputs.

For the Bayes optimum decision, a vector x is assigned to class i if $p_i(x) > p_k(x)$, $\forall k \neq i$, so the Bayes optimal boundary is the loci of all points x^*: $p_i(x^*) = p_j(x^*)$ for a two-class problem. Since our classifier provides $f_i(\cdot)$ instead of $p_i(\cdot)$, the decision boundary obtained, x_b, may vary from the ideal boundary (see Figure 2). Let b denote the amount by which the boundary of the classifier differs from the ideal boundary ($b = x_b - x^*$). We have:

$$f_i(x^* + b) = f_j(x^* + b),$$

by definition of the boundary. This implies:

$$p_i(x^* + b) + \epsilon_i(x_b) \; = \; p_j(x^* + b) + \epsilon_j(x_b)\,. \tag{2}$$

Now, let us assume that the posteriors are locally monotonic functions around the decision boundaries. This hypothesis is well founded since typically the boundaries are

[1] If two or more classifiers need to be distinguished, a superscript is added to $f_i(x)$ and $\epsilon_i(x)$ to indicate the classifier number.

[2] Here, $p_i(x)$ is used for simplicity to denote $p(C_i|x)$.

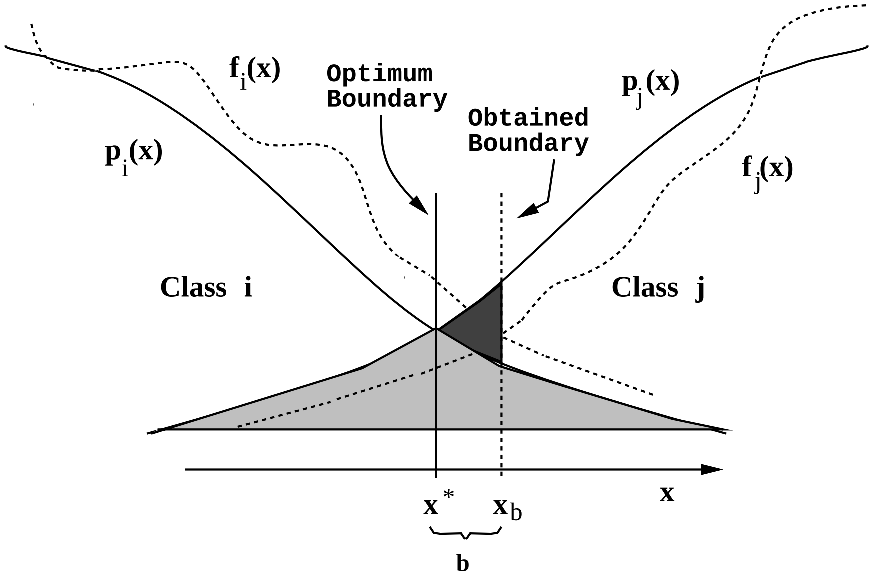

Figure 2: Error regions associated with approximating the *a posteriori* probablities.

located in transition regions where the posteriors are not in a local extrema. Then a linear approximation of $p_k(x)$ around x^* provides:

$$p_k(x^* + b) \simeq p_k(x^*) \;+\; b\, p_k'(x^*) \;\; \forall k \;,$$

where $p_k'(\cdot)$ denotes the derivative of $p_k(\cdot)$. With this substitution, Equation 2 becomes:

$$p_i(x^*) \;+\; b\, p_i'(x^*) \;+\; \epsilon_i(x_b) \;=\; p_j(x^*) \;+\; b\, p_j'(x^*) \;+\; \epsilon_j(x_b). \tag{3}$$

Now, since $p_i(x^*) = p_j(x^*)$, we get:

$$b\,(p_j'(x^*) \;-\; p_i'(x^*)) \;=\; \epsilon_i(x_b) - \epsilon_j(x_b).$$

Finally we obtain:

$$b \;=\; \frac{\epsilon_i(x_b) - \epsilon_j(x_b)}{s}, \tag{4}$$

where:

$$s \;=\; p_j'(x^*) \;-\; p_i'(x^*). \tag{5}$$

Equation 4 can be used to obtain the distribution of b. Let the error $\epsilon_i(x_b)$ be broken into a bias and a zero-mean noise term ($\epsilon_i(x_b) = \beta_i + \eta_i(x_b)$). For the time being, the bias is assumed to be zero (i.e. $\beta_k = 0 \; \forall k$), and the error is entirely due to noise. The case with non-zero bias will be discussed in the next section. Let the noise $\eta_k(x)$ be independent $\forall k$, and have Gaussian distributions with zero-mean and $\sigma^2_{\eta_k}$ variance[3]. Then, b is a Gaussian random variable with zero-mean and variance σ_b^2 where:

$$\sigma_b^2 \;=\; \frac{\sigma^2_{\eta_i} + \sigma^2_{\eta_j}}{s^2}.$$

Figure 2 shows the *a posteriori* probabilities obtained by a non-ideal classifier, and the added error region associated with it. The lightly shaded area provides the Bayesian error region. The darkly shaded area is the added error region associated with selecting a decision boundary that is offset by b, since patterns corresponding to the darkly shaded region are erroneously assigned to class i by the classifier, although ideally they should be assigned to class j.

Let us now divert our attention to the effects of combining multiple classifiers. In what follows, the combiner denoted by *ave* performs an arithmetic average in output space. If N classifiers are available, by using the *ave* combiner, we obtain an approximation to $p_i(x)$ given by:

$$f_i^{ave}(x) = \frac{1}{N} \sum_{m=1}^{N} f_i^m(x),$$

which can be written as:

$$f_i^{ave}(x) = p_i(x) + \bar{\eta}_i(x),$$

where:

$$\bar{\eta}_i(x) = \frac{1}{N} \sum_{m=1}^{N} \eta_i^m(x).$$

If the errors of different classifiers are independent, the variance of $\bar{\eta}_i$ is given by:

$$\sigma^2_{\bar{\eta}_i} = \frac{1}{N^2} \sum_{m=1}^{N} \sigma^2_{\eta_i^m} \; .$$

The boundary x^{ave} then has an offset b^{ave}, where:

$$f_i^{ave}(x^* + b^{ave}) = f_j^{ave}(x^* + b^{ave}),$$

and the variance $\sigma^2_{b^{ave}}$ can be computed in a manner similar to σ_b^2, resulting in:

$$\sigma^2_{b^{ave}} \;=\; \frac{\sigma^2_{\bar{\eta}_i} + \sigma^2_{\bar{\eta}_j}}{s^2}.$$

[3] Each output of each network does approximate a smooth function, and therefore the noise for two nearby patterns on the same class (i.e. $\eta_k(x)$ and $\eta_k(x + \Delta x)$) is correlated. The independence assumption applies to inter-class noise (i.e. $\eta_i(x)$ and $\eta_j(x)$), not intra-class noise.

In particular, if $\sigma^2_{\eta_i^m} \;=\; \sigma^2_{\eta_i^l}, \forall m, l$, we get:

$$\sigma^2_{\bar{\eta}_i} = \frac{1}{N}\sigma^2_{\eta_i}\,, \tag{6}$$

which leads to:

$$\sigma^2_{b^{ave}} \;=\; \frac{\sigma^2_{\eta_i} + \sigma^2_{\eta_j}}{N\,s^2},$$

or:

$$\sigma^2_{b^{ave}} \;=\; \frac{\sigma^2_b}{N}. \tag{7}$$

This reduction in variance can be readily translated into a reduction in error rates, since a narrower boundary distribution means the likelihood that a boundary will be near the ideal one is increased. In effect, using the evidence of more than one classifier reduces the variance of the class boundary from the ideal one, thereby providing a "tighter" error-prone area. In order to establish the exact improvements in the classification rate, the expected added error region will be computed, and the relationship between classifier boundary variance and error rates will be explored further in Section 4.1.

3 Class Boundary Analysis in Presence of Bias

In general, the estimate of the posterior probabilities obtained by a network will be biased, i.e. $\beta_k \neq 0$. As discussed, in the previous section, the error is expressed as the sum of bias and noise, resulting in:

$$f_i(x) = p_i(x) \;+\; \beta_i \;+\; \eta_i(x).$$

Here, β_i is the bias introduced by the classifier[4], and $\eta_i(x)$ is the zero-mean noise term of Section 2.

Proceeding in a manner similar to that of Section 2, one readily obtains:

$$b \;=\; \frac{\eta_i(x_b) - \eta_j(x_b)}{s} \;+ \beta, \tag{8}$$

where s is as in Equation 5 and:

$$\beta \;=\; \frac{\beta_i - \beta_j}{s}.$$

[4]The bias is expected to be different for distinct classes. If the bias term is a simple additive constant, independent of the class (that is $\beta_i = \beta_j$), then in the difference $f_i(x) - f_j(x)$, the biases cancel out, reducing the decision boundary to the one of the previous section.

Again taking the noise to be independent between classes and Gaussian with zero-mean and variance $\sigma^2_{\eta_i}$, we conclude that b is a Gaussian random variable with mean β and variance σ^2_b, which is given by Equation 6.

Combining multiple classifiers through *ave* provides:

$$f_i^{ave}(x) = p_i(x) \ + \ \bar{\beta}_i \ + \ \bar{\eta}_i(x) \, ,$$

where:

$$\bar{\beta}_i = \frac{1}{N} \sum_{m=1}^{N} \beta_i^m \ ,$$

and

$$\bar{\eta}_i(x) = \frac{1}{N} \sum_{m=1}^{N} \eta_i^m(x).$$

The variance of $\bar{\eta}_i$(x) is given in Section 2. The boundary x^{ave} has an offset b^{ave} given by:

$$b^{ave} \ = \ \frac{\bar{\eta}_i(x) - \bar{\eta}_j(x)}{s} + \bar{\beta}.$$

The distribution of b^{ave} can be obtained from those of $\bar{\eta}_i(x)$ and $\bar{\eta}_j(x)$, and yields a Gaussian distribution with mean $\bar{\beta}$ and variance $\sigma^2_{b^{ave}}$.

The effect of combining is less clear in this case, since the average bias ($\bar{\beta}$) is not necessarily less than each of the individual biases. The effect of the bias on the error regions will be studied in detail in Section 4.2. However, from an inspection of the distribution of b^{ave} certain observations can be made. If the bias is extremely small, and the error is mainly due to the variance, combining can be an effective tool. If however errors are mainly due to high bias, this type of combining becomes effective only if the biases are not highly correlated.

These limiting cases for the error (mostly bias or mostly variance) also show a new approach to tackling the well known bias/variance problem [3]. By keeping the bias very small for each classifier, achieved by using larger networks than necessary, combining reduces the errors, mainly due to variance, significantly. These results highlight the basic strengths of combining, which not only provides improved error rates, but is also a method of controlling the bias and variance components of the error separately. The selection of network size and training regime can then directly reflect this result.

4 Added Error Region Analysis

4.1 Added Errors in the Absence of Bias

In the previous section, we showed that combining is an effective way of reducing the variance of the decision boundaries. The question of how this result translates into improved classification results is discussed in this section.

The added error region associated with a classifier, denoted by $A(b)$, is given by:

$$A(b) \;=\; \int_{x^*}^{x^*+b} (p_j(x) - p_i(x))\, dx,$$

which is the darkly shaded region in Figure 2. Based on this area, the expected added error, E_{add}, is given by:

$$E_{add} = \int_{-\infty}^{\infty} A(b) f_b(b) db, \tag{9}$$

where f_b is the density function for b, as discussed in Section 2. The expected error becomes:

$$E_{add} = \int_{-\infty}^{\infty} \int_{x^*}^{x^*+b} (p_j(x) - p_i(x))\;\, f_b(b)\, dx db.$$

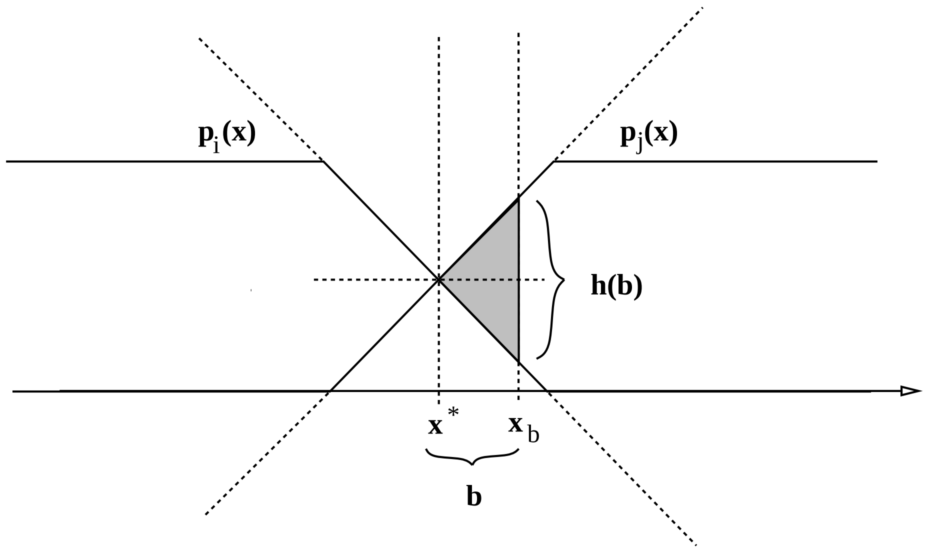

Figure 3: Linear added error region analysis.

Now, recall that the boundary is located in the region where the posteriors are locally monotonic. This allows the linear approximation discussed in Section 2, making the region $A(b)$ triangular in shape (see Figure 3). The accuracy of this approximation depends on the proximity of the boundary to the ideal boundary. However, since the boundary density decreases exponentially with increasing distance from the ideal boundary, we can expect the triangular region to reasonably represent the added error area for most likely (i.e. small) values of b. The base, $h(b)$, of the triangular region is simply $b\, p_j'(x) \;-\; b\, p_i'(x)$. Thus we obtain:

$$\begin{aligned} A(b) \;&=\; \frac{1}{2}\, b \left(b\, p_j'(x^*) \;-\; b\, p_i'(x^*) \right) \\ &=\; \frac{1}{2}\, b^2\, s, \end{aligned} \tag{10}$$

where s is given by Equation 5. Furthermore, due to the symmetry of the problem, the integration of Equation 9 can be performed only for $b \geq 0$, and multiplied by two.

By using the value given by Equation 10 for $A(b)$, and performing the integration for $b \geq 0$, the expected error becomes:

$$E_{add} = 2 \left(\int_0^\infty \frac{s}{2} \, b^2 \, \frac{1}{\sqrt{2\pi\sigma_b^2}} \, e^{-\frac{b^2}{2\sigma_b^2}} db \right), \tag{11}$$

leading to:

$$E_{add} = \frac{s}{\sqrt{2\pi\sigma_b^2}} \int_0^\infty b^2 \, e^{-\frac{b^2}{2\sigma_b^2}} db.$$

Integrating by parts we obtain:

$$E_{add} = \frac{s\sigma_b}{\sqrt{2\pi}} \left(\left[-b \, e^{-\frac{b^2}{2\sigma_b^2}} \right]_0^\infty + \int_0^\infty e^{-\frac{b^2}{2\sigma_b^2}} db \right).$$

The first term gives a value of 0 at both limits, and the second term gives $\frac{\sqrt{2\pi\sigma_b^2}}{2}$. The expected error then yields:

$$E_{add} = \frac{s \, \sigma_b^2}{2}. \tag{12}$$

The importance of Equation 12 is that it provides a direct relationship between the expected added error region and the variance of b, the amount by which the selected boundary differs from the ideal boundary. Any reduction in the variance of b is directly translated into a reduction in expected error rates. Let the error region associated with classifier *ave* be denoted by E_{add}^{ave}:

$$E_{add}^{ave} = \frac{s \, \sigma_{b^{ave}}^2}{2} = \frac{s \, \sigma_b^2}{2 \, N} = \frac{E_{add}}{N}. \tag{13}$$

Equation 13 quantifies the improvements due to combining N classifiers. Under the assumptions of Section 2, combining in output space reduces added error regions by a factor of N. Of course, the total error, which is the sum of Bayes error and the added error, will be reduced by a smaller amount, since Bayesian error will be non-zero for problems with overlapping classes.

The value provided by Equation 12 will generally tend to be a conservative bound on the added error. The total height $h(b)$ is bounded by 1, since it is the difference of two *a posteriori* probability distributions. A more accurate method for computing A(b) is to use the following height equation:

$$h(b) = \begin{cases} b \, s & \text{if } 0 \leq b \leq \frac{1}{s} \\ 1 & \text{otherwise} \end{cases}$$

Using this height in computing the added error area leads Equation 9 to:

$$E'_{add} = 2\left(\int_0^{\frac{1}{s}} A(b) f_b(b) db \,+\, \int_{\frac{1}{s}}^{\infty} A(b) f_b(b) db\right),$$

which leads to:

$$E'_{add} = \frac{2}{\sqrt{2\pi\sigma_b^2}} \left(\int_0^{\frac{1}{s}} \frac{b^2\, s}{2} e^{-\frac{b^2}{2\sigma_b^2}} db \,+\, \int_{\frac{1}{s}}^{\infty} \left[A(\frac{1}{s}) \,+\, (b-\frac{1}{s})\right] e^{-\frac{b^2}{2\sigma_b^2}} db\right),$$

or:

$$\begin{aligned} E'_{add} &= \frac{s}{\sqrt{2\pi\sigma_b^2}} \int_0^{\frac{1}{s}} b^2 e^{-\frac{b^2}{2\sigma_b^2}} db \\ &+ \frac{2\,(A(\frac{1}{s})-\frac{1}{s})}{\sqrt{2\pi\sigma_b^2}} \int_{\frac{1}{s}}^{\infty} e^{-\frac{b^2}{2\sigma_b^2}} db \\ &+ \frac{2}{\sqrt{2\pi\sigma_b^2}} \int_{\frac{1}{s}}^{\infty} b\, e^{-\frac{b2}{2\sigma_b^2}} db. \end{aligned} \tag{14}$$

Equation 14 provides a more accurate added error term than Equation 12. However, due to its considerable complexity, it is generally not preferable to compute the added error in this form. If Equation 14 needs to be explicitly computed, the following procedure can be followed: The first term can be computed using integration by parts, the second term can be expressed in terms of Gaussian distribution functions ($F(\cdot)$), and the third term can be integrated, leading to:

$$E'_{add} \;=\; \frac{-3\sigma_b}{\sqrt{2\pi}}\, e^{-\frac{1}{2s^2\sigma_b^2}} \,+\, s\sigma_b^2 \left(F(\frac{1}{s}) - \frac{1}{2}\right) \,-\, \frac{1}{s}\left(1 - F(\frac{1}{s})\right).$$

However, it is important to note that in a majority of cases, Equation 12 will be sufficient. Only when the boundary provided by classifier m falls in the region where the *a posteriori* probabilities are at their limiting values is a more accurate expression needed. A classifier that repeatedly puts the boundary in such a region is of little use in general. Therefore, it is reasonable to expect that most classifiers will provide boundaries that fall in the region where the linear approximation is adequate.

4.2 Added Errors in the Presence of Bias

In this section we compute the expected error in the presence of a bias β. The actual error area as computed in the previous section is not affected by the bias. However the distribution of b is affected and the expected value takes a different form. Equation 9 leads to:

$$E_{add}(\beta) = \int_{-\infty}^{\infty} \frac{s}{2}\, b^2\, \frac{1}{\sqrt{2\pi\sigma_b^2}}\, e^{-\frac{(b-\beta)^2}{2\sigma_b^2}}\, db, \tag{15}$$

which is similar to Equation 11, but for the shift in the center of the distribution of b. A change of variable is needed to enable us to compute this expression. Let $y = b - \beta$ (y has the same variance as b but has zero mean). With this change of variable we obtain:

$$E_{add}(\beta) = \frac{s}{2\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} (y+\beta)^2 \, e^{-\frac{y^2}{2\sigma_b^2}} \, dy,$$

which leads to:

$$\begin{aligned} E_{add}(\beta) &= \frac{s}{2\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} y^2 \, e^{-\frac{y^2}{2\sigma_b^2}} \, dy \\ &+ \frac{s\,\beta^2}{2\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} e^{-\frac{y^2}{2\sigma_b^2}} \, dy \\ &+ \frac{s\,\beta}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} y \, e^{-\frac{y^2}{2\sigma_b^2}} \, dy. \end{aligned}$$

Simplifying each component in terms of first and second moments provides:

$$E_{add}(\beta) = \frac{s}{2}E[y^2] + \frac{s}{2}\,\beta^2 + s\,\beta E[y]$$

or:

$$E_{add}(\beta) = \frac{s}{2}(\sigma_b^2 + \beta^2) \tag{16}$$

Equation 16 reduces to Equation 12 if the bias is set to zero.

The effect of combining is not as obvious as it previously was. Let us study the error associated with $E_{add}^{ave}(\beta^{ave})$:

$$E_{add}^{ave}(\beta^{ave}) = \frac{s}{2}(\sigma_{b^{ave}}^2 + (\beta^{ave})^2)$$

which is:

$$E_{add}^{ave}(\beta^{ave}) = \frac{s}{2}\left(\frac{\sigma_b^2}{N} + \frac{\beta^2}{z^2}\right) \tag{17}$$

where $\beta^{ave} = \frac{\beta}{z}$, and $z \geq 1$. Now let us limit the study to the case where $z \leq \sqrt{N}$. Then[5]:

$$E_{add}^{ave}(\beta^{ave}) \leq \frac{s}{2}\left(\frac{\sigma_b^2 + \beta^2}{z^2}\right)$$

leading to:

$$E_{add}^{ave}(\beta^{ave}) \leq \frac{1}{z^2}\,E_{add}(\beta). \tag{18}$$

[5] If $z \geq \sqrt{N}$, then the reduction of the variance becomes the limiting factor, and the reductions established in the previous section hold.

Equation 18 quantifies the error reduction in the presence of network bias. The improvements are more modest than those of the previous section, since both the bias and the variance of the noise need to be reduced. The actual reduction is given by $min(z^2, N)$, demonstrating that the smaller reduction is the limiting factor. This result underlines why methods aimed at reducing only the variance or only the bias generally do not lead to significant improvements in overall classification performance.

5 Discussion

Combining classifiers in output space has led to improved performance in many applications [12, 13, 20]. This paper concentrates on explaining the reasons for expecting such improvements and to quantify the gains achieved. Under the assumption that the *a posteriori* probability distributions for each class are locally monotonic functions about the decision boundaries, we showed that combining networks in output space reduces the variance in boundary locations. Furthermore, the error regions are directly computed and given in terms of the boundary distribution parameters. In the absence of network bias, the reduction in the error is directly proportional to the reduction in the variance. Moreover, if the network errors are zero-mean i.i.d. Gaussian, then the reduction in variance boundary location is by a factor of N, the number of classifiers that are combined. In the presence of network bias, the reductions are less than or equal to N, depending on the correlation among the network biases.

Although our analysis focused on only two classes, it is readily applicable to a multi-class problem. Since the largest network output determines class membership, only a handful of classes are likely at any given point in input space. Therefore, even in a multi-class problem, one only needs to consider the two classes with the highest activation values in a given localized region.

The distribution of the boundary is shown to be Gaussian through its relationship with the noise terms. If the noise proves to have a distribution other than Gaussian, the analysis can be modified to accommodate the new distribution. The expected error given in Equation 9 is in general form, and any density function can be used from there on to reflect changes in the distribution function. For problems with higher dimensionality, the analysis becomes significantly more complicated, but retains the same conceptual structure. The added error area of Figure 3 becomes a volume for 2-dimensional signals, and in general is an $(n+1)$-dimensional hypervolume for n-dimensional problems. In the most general case we have:

$$E(m) = \int_{\mathbb{R}^n} \int_A h(x_1, x_2, \cdots, x_n) dx_1 dx_2 \cdots dx_n \; f_B(B) \, dB.$$

where $h(\cdot)$ delineates a multidimensional difference between *a posteriori* probabilities defined over an n-dimensional region A, and $f_B(B)$ is the multidimensional density function of the

n-dimensional boundary B defined over $\mathbb{R}^n$.

Another important feature of combining that arose from this study relates to the classic bias/variance dilemma. Combining provides a method for decoupling the two components of the error to a degree, allowing a reduction in the overall error. Bias in the individual networks can be reduced by using larger networks than required, and the increased variance due to the larger networks can be reduced during the combining stage. Studying the effects of this coupling between different errors and distinguishing situations that lead to the highest error reduction rates are the driving motivations behind this work. That goal is attained by clarifying the relationship between output space combining and classification performance. The analysis presented here provides an understanding of the interactions between the error rates and classifier boundary distributions, and ultimately between error rates and output space combining.

Several practical issues that relate to this analysis can now be addressed. First, let us note that since in general each individual network will have some amount of bias, the actual improvements will be less radical than those obtained in Section 4.1. It is therefore important to determine how the biases of individual networks can be kept uncorrelated (or have only minimal correlation). One method is to use networks with architectures based on different principles. For example, using multi-layered perceptrons and radial basis function networks provides both global and local information processing, ensuring that the biases are not highly correlated. Another method is to train similar networks on different features extracted from the same underlying data. Although the same network type is used, the biases will less correlated, since they are a function of the training data as well as the network. Experimental results obtained by us on an oceanic data set with four distinct classes support the above conclusions [21].

One final note that needs to be considered is the behavior of combiners for a large number of classifiers (N). Clearly, the errors cannot be arbitrarily reduced by increasing N indefinitely. This observation however, does not contradict the results presented in this analysis. For large N, the assumption that the errors were i.i.d. breaks down, reducing the improvements due to each extra classifier. The number of classifiers that yield the best results depends on a number of factors, including the number of feature sets extracted from the data, their dimensionality, and the selection of the network architectures.

The focus of this paper is on combining in output space through averaging. Although the simplicity of averaging provides a pleasing framework, it is not the only method that yields encouraging results. As mentioned previously, there are many other possibilities in combining networks that require closer investigation. The use of order statistics, for example, promises to provide improvements that can be analytically studied, and we are currently pursuing that line of research.

Acknowledgements: This research was supported in part by AFOSR contract F49620-93-1-0307, ONR contract N00014-92-C-0232, and NSF grant ECS 9307632.

References

[1] E. Levin, N. Tishby, and S. A. Solla. A statistical approach to learning and generalization in layered neural networks. *Proc. IEEE*, 78(10):1568–74, Oct 1990.

[2] D. H. Wolpert. A mathematical theory of generalization. *Complex Systems*, 4:151–200, 1990.

[3] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[4] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[5] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[6] J. Ghosh and K. Tumer. Structural adaptation and generalization in supervised feed-forward networks. *Journal of Artificial Neural Networks*, 1(4):431–458, 1994.

[7] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[8] W.P. Lincoln and J. Skrzypek. Synergy of clustering multiple back propagation networks. In *Advances in Neural Information Processing Systems -2*, pages 650–657, 1990.

[9] M.P. Perrone and L. N. Cooper. When networks disagree: Ensemble methods for hybrid neural networks. In R. J. Mammone, editor, *Neural Networks for Speech and Image Processing*. Chapmann-Hall, 1993.

[10] K. Tumer and J. Ghosh. A framework for estimating performance improvements in hybrid pattern classifiers. In *Proceedings of the World Congress on Neural Networks*, pages III:220–225. INNS Press, 1994.

[11] J. Ghosh, S. Beck, and C.C. Chu. Evidence combination techniques for robust classification of short-duration oceanic signals. In *SPIE Conf. on Adaptive and Learning Systems, SPIE Proc. Vol. 1706*, pages 266–276, Orlando, Fl., April 1992.

[12] J. Ghosh, K. Tumer, S. Beck, and L. Deuser. Integration of neural classifiers for passive sonar signals. In C.T. Leondes, editor, *DSP Theory and Applications*. Academic Press, 1995.

[13] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 22(3):418–435, May 1992.

[14] G. Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

[15] D. Heckerman. Probabilistic interpretation for MYCIN's uncertainty factors. In L.N Kanal and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 167–196. North-Holland, 1986.

[16] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation*, 3(4):461–483, 1991.

[17] P.A. Shoemaker, M.J. Carlin, R.L. Shimabukuro, and C.E. Priebe. Least squares learning and approximation of posterior probabilities on classification problems by neural network models. In *Proc. 2nd Workshop on Neural Networks, WNN-AIND91,Auburn*, pages 187–196, February 1991.

[18] S. Hashem. *Optimal Linear Combinations of Neural Networks.* PhD thesis, Purdue University, December 1993.

[19] M. P. Perrone. *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization.* PhD thesis, Brown University, May 1993.

[20] X. Zhang, J.P. Mesirov, and D.L. Waltz. Hybrid system for protein secondary structure prediction. *J. Molecular Biology*, 225:1049–63, 1992.

[21] K. Tumer and J. Ghosh. Boundary variance reduction for improved classification through hybrid networks (Invited paper). In *Applications and Science of Artificial Neural Networks, Proceedings of the SPIE*, volume 2492, pages 573–584, April 1995.