LINEAR AND ORDER STATISTICS COMBINERS FOR PATTERN CLASSIFICATION

Kagan Tumer
NASA Ames Research Center
MS 269-4, Moffett Field, CA, 94035-1000
kagan@ptolemy.arc.nasa.gov

Joydeep Ghosh
Department of Electrical and Computer Engineering,
University of Texas, Austin, TX 78712-1084
ghosh@pine.ece.utexas.edu

Abstract

Several researchers have experimentally shown that substantial improvements can be obtained in difficult pattern recognition problems by combining or integrating the outputs of multiple classifiers. This chapter provides an analytical framework to *quantify* the improvements in classification results due to combining. The results apply to both linear combiners and order statistics combiners. We first show that to a first order approximation, the error rate obtained over and above the Bayes error rate, is directly proportional to the variance of the actual decision boundaries around the Bayes optimum boundary. Combining classifiers in output space reduces this variance, and hence reduces the "added" error. If N unbiased classifiers are combined by simple averaging, the added error rate can be reduced by a factor of N if the individual errors in approximating the decision boundaries are uncorrelated. Expressions are then derived for linear combiners which are biased or correlated, and the effect of output correlations on ensemble performance is quantified. For order statistics based non-linear combiners, we derive expressions that indicate how much the median, the maximum and in general the ith order statistic can improve classifier performance. The analysis presented here facilitates the understanding of the relationships among error rates, classifier boundary distributions, and combining in output space. Experimental results on several public domain data sets are provided to illustrate the benefits of combining and to support the analytical results.

1 Introduction

Training a parametric classifier involves the use of a *training* set of data with known labeling to estimate or "learn" the parameters of the chosen model. A *test* set, consisting of patterns not previously seen 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 [38, 77]. Given infinite training data, consistent classifiers approximate the Bayesian decision boundaries to arbitrary precision, therefore providing similar generalizations [24]. However, often only a limited portion of the pattern space is available or observable [16, 22]. Given a finite and noisy data set, different classifiers typically provide different generalizations by realizing different decision boundaries [26]. For example, when classification is performed using a multilayered, feed-forward artificial neural network, different weight initializations,

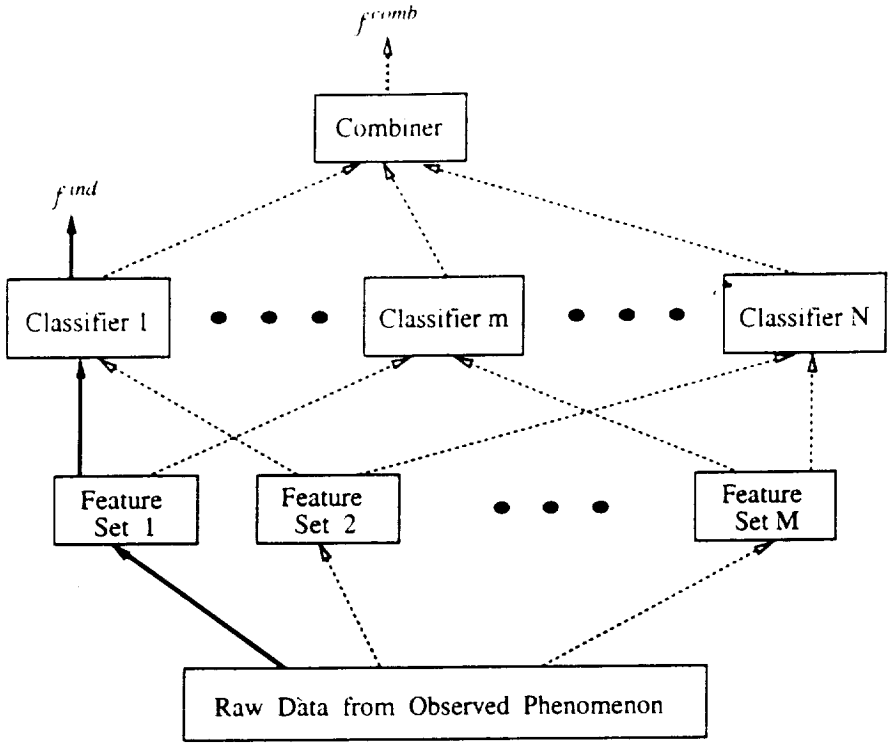

Figure 1: Combining Strategy. The solid lines leading to f^{ind} represent the decision of a specific classifier, while the dashed lines lead to f^{comb}, the output of the combiner.

- relates the location of the decision boundary to the classifier error.

The rest of this article is organized as follows. Section 2 introduces the overall framework for estimating error rates and the effects of combining. In Section 3 we analyze linear combiners, and derive expressions for the error rates for both biased and unbiased classifiers. In Section 4, we examine order statistics combiners, and analyze the resulting classifier boundaries and error regions. In Section 5 we study linear combiners that make correlated errors, derive their error reduction rates, and discuss how to use this information to build better combiners. In Section 6, we present experimental results based on real world problems, and we conclude with a discussion of the implications of the work presented in this article.

2 Class Boundary Analysis and Error Regions

Consider a single classifier whose outputs are expected to approximate the corresponding *a posteriori* class probabilities if it is reasonably well trained. The decision boundaries obtained by such a classifier are thus expected to be close to Bayesian decision boundaries. Moreover, these boundaries will tend to occur in regions where the number of training samples belonging to the two most locally dominant classes (say, classes i and j) are comparable.

We will focus our analysis on network performance around the decision boundaries. Consider the boundary between classes i and j for a single-dimensional input (the extension to multi-dimensional inputs is discussed in [73]). First, let us express the output response of the ith unit of a *one-of-L*

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

$$p_i(x^*) + b\,p'_i(x^*) + \epsilon_i(x_b) = p_j(x^*) + b\,p'_j(x^*) + \epsilon_j(x_b). \tag{4}$$

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:

$$b = \frac{\epsilon_i(x_b) - \epsilon_j(x_b)}{s}, \tag{5}$$

where:

$$s = p'_j(x^*) - p'_i(x^*). \tag{6}$$

Let the error $\epsilon_i(x_b)$ be broken into a bias and noise term ($\epsilon_i(x_b) = \beta_i + \eta_i(x_b)$). Note that the term "bias" and "noise" are only analogies, since the error is due to the classifier as well as the data. For the time being, the bias is assumed to be zero (i.e. $\beta_k = 0\ \forall k$). The case with nonzero bias will be discussed at the end of this section. Let $\sigma^2_{\eta_k}$ denote the variances of $\eta_k(x)$, which are taken to be i.i.d. variables[3]. Then, the variance of the zero-mean variable b is given by (using Equation 5):

$$\sigma^2_b = \frac{2\,\sigma^2_{\eta_k}}{s^2}. \tag{7}$$

Figure 2 shows the *a posteriori* probabilities obtained by a non-ideal classifier, and the associated added error region. 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.

The added error region, denoted by $A(b)$, is given by:

$$A(b) = \int_{x^*}^{x^*+b} (p_j(x) - p_i(x))\,dx. \tag{8}$$

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. More explicitly, the expected added error is:

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

One can compute $A(b)$ directly by using the approximation in Equation 3 and solving Equation 8. The accuracy of this approximation depends on the proximity of the boundary to the ideal boundary. However, since in general, the boundary density decreases rapidly with increasing distance from the

[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.

3 Linear Combining

3.1 Linear Combining of Unbiased Classifiers

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

$$f_i^{ave}(x) = \frac{1}{N} \sum_{m=1}^{N} f_i^m(x), \tag{16}$$

or:

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

where:

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

and

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

If the classifiers are unbiased, $\bar{\beta}_i = 0$. Moreover, if the errors of different classifiers are i.i.d., the variance of $\bar{\eta}_i$ is given by:

$$\sigma_{\bar{\eta}_i}^2 = \frac{1}{N^2} \sum_{m=1}^{N} \sigma_{\eta_i^m}^2 = \frac{1}{N} \sigma_{\eta_i}^2. \tag{17}$$

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

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

and:

$$b^{ave} = \frac{\bar{\eta}_i(x_{b^{ave}}) - \bar{\eta}_j(x_{b^{ave}})}{s}. \tag{18}$$

The variance of b^{ave}, $\sigma_{b^{ave}}^2$, can be computed in a manner similar to σ_b^2, resulting in:

$$\sigma_{b^{ave}}^2 = \frac{\sigma_{\bar{\eta}_i}^2 + \sigma_{\bar{\eta}_j}^2}{s^2},$$

which, using Equation 17, leads to:

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

or:

$$\sigma_{b^{ave}}^2 = \frac{\sigma_b^2}{N}. \tag{19}$$

leading to:

$$E_{add}^{ave}(\beta) \leq \frac{1}{N} E_{add}(\beta). \tag{24}$$

Equation 24 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. If both the variance and the bias contribute to the error, and their contributions are of similar magnitude, the actual reduction is given by $min(c^2, N)$. If the bias can be kept low (e.g. by purposefully using a larger network than required), then once again N becomes the reduction factor. 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, thus providing an interesting solution to the bias/variance problem [24].

4 Order Statistics

4.1 Introduction

Approaches to pooling classifiers can be separated into two main categories: simple combiners, e.g., averaging, and computationally expensive combiners, e.g., stacking. The simple combining methods are best suited for problems where the individual classifiers perform the same task, and have comparable success. However, such combiners are susceptible to outliers and to unevenly performing classifiers. In the second category, "meta-learners," i.e., either sets of combining rules, or full fledged classifiers acting on the outputs of the individual classifiers, are constructed. This type of combining is more general, but suffers from all the problems associated with the extra learning (e.g., overparameterizing, lengthy training time).

Both these methods are in fact ill-suited for problems where *most* (but not all) classifiers perform within a well-specified range. In such cases the simplicity of averaging the classifier outputs is appealing, but the prospect of one poor classifier corrupting the combiner makes this a risky choice. Although, weighted averaging of classifier outputs appears to provide some flexibility, obtaining the optimal weights can be computationally expensive. Furthermore, the weights are generally assigned on a per classifier, rather than per sample or per class basis. If a classifier is accurate only in certain areas of the inputs space, this scheme fails to take advantage of the variable accuracy of the classifier in question. Using a meta learner that would have weights for each classifier on each pattern, would solve this problem, but at a considerable cost. The robust combiners presented in this section aim at bridging the gap between simplicity and generality by allowing the flexible selection of classifiers without the associated cost of training meta classifiers.

4.2 Background

In this section we will briefly discuss some basic concepts and properties of order statistics. Let X be a random variable with a probability density function $f_X(\cdot)$, and cumulative distribution function $F_X(\cdot)$. Let $(X_1, X_2, \cdots, X_N)$ be a random sample drawn from this distribution. Now, let us arrange them in non-decreasing order, providing:

$$X_{1:N} \leq X_{2:N} \leq \cdots \leq X_{N:N}.$$

on the other hand considers the most "typical" representation of each class. For highly noisy data, this combiner is more desirable than either the *min* or *max* combiners since the decision is not compromised as much by a single large error.

The analysis of the properties of these combiners does not depend on the order statistic chosen. Therefore we will denote all three by $f_i^{os}(x)$ and derive the error regions. The network output provided by $f_i^{os}(x)$ is given by:

$$f_i^{os}(x) = p_i(x) + \epsilon_i^{os}(x) \,. \tag{29}$$

Let us first investigate the zero-bias case ($\beta_k = 0 \; \forall k$). We get $\epsilon_k^{os}(x) = \eta_k^{os}(x) \; \forall k$, since the variations in the kth output of the classifiers are solely due to noise. Proceeding as before, the boundary b^{os} is shown to be:

$$b^{os} = \frac{\eta_i^{os}(x_b) - \eta_j^{os}(x_b)}{s}. \tag{30}$$

Since η_k's are i.i.d. and η_k^{os} is the same order statistic for each class, the moments will be identical for each class. Moreover, taking the order statistic will shift the mean of both η_i^{os} and η_j^{os} by the same amount, leaving the mean of the difference unaffected. Therefore, b^{os} will have zero mean, and variance:

$$\sigma_{b^{os}}^2 = \frac{2\,\sigma_{\eta_k^{os}}^2}{s^2} = \frac{2\,\alpha\sigma_{\eta_k}^2}{s^2} = \alpha\sigma_b^2, \tag{31}$$

where α is a reduction factor that depends on the order statistic and on the distribution of b. For most distributions, α can be found in tabulated form [3]. For example, Table 1 provides α values for all three *os* combiners, up to 15 classifiers, for a Gaussian distribution [3, 58].

Returning to the error calculation, we have: $M_1^{os} = 0$, and $M_2^{os} = \sigma_{b^{os}}^2$, providing:

$$E_{add}^{os} = \frac{sM_2^{os}}{2} = \frac{s\sigma_{b^{os}}^2}{2} = \frac{s\alpha\sigma_b^2}{2} = \alpha\; E_{add}. \tag{32}$$

Equation 32 shows that the reduction in the error region is directly related to the reduction in the variance of the boundary offset b. Since the means and variances of order statistics for a variety of distributions are widely available in tabular form, the reductions can be readily quantified.

4.4 Combining Biased Classifiers through OS

In this section, we analyze the error regions in the presence of bias. Let us study b^{os} in detail when multiple classifiers are combined using order statistics. First note that the bias and noise cannot be separated, since in general $(a+b)^{os} \neq a^{os} + b^{os}$. We will therefore need to specify the mean and variance of the result of each operation[6]. Equation 30 becomes:

$$b^{os} = \frac{(\beta_i + \eta_i(x_b))^{os} - (\beta_j + \eta_j(x_b))^{os}}{s}. \tag{33}$$

Now, β_k has mean $\bar{\beta}_k$, given by $\frac{1}{N}\sum_{m=1}^{N}\beta_k^m$, where m denotes the different classifiers. Since the noise is zero-mean, $\beta_k + \eta_k(x_b)$ has first moment $\bar{\beta}_k$ and variance $\sigma_{\eta_k}^2 + \sigma_{\beta_k}^2$, where $\sigma_{\beta_k}^2 = \frac{1}{N-1}\sum_{m=1}^{N}(\beta_k^m - \bar{\beta}_k)^2$.

[6]Since the exact distribution parameters of b^{os} are not known, we use the sample mean and the sample variance.

we get

$$E_{add}^{os}(\beta) = \alpha E_{add}(\beta) + \frac{s}{2}(\alpha\sigma_\beta^2 + \bar{\beta}^2 - \alpha\beta^2) \quad (39)$$

Analyzing the error reduction in the general case requires knowledge about the bias introduced by each classifier. However, it is possible to analyze the extreme cases. If each classifier has the same bias for example, σ_β^2 is reduced to zero and $\bar{\beta} = \beta$. In this case the error reduction can be expressed as:

$$E_{add}^{os}(\beta) = \frac{s}{2}(\alpha\sigma_b^2 + \beta^2),$$

where only the error contribution due to the variance of b is reduced. In this case it is important to reduce classifier bias before combining (e.g. by using an overparametrized model). If on the other hand, the biases produce a zero mean variable, i.e. they cancel each other out, we obtain $\bar{\beta} = 0$. In this case, the added error becomes:

$$E_{add}^{os}(\beta) = \alpha E_{add}(\beta) + \frac{s\,\alpha}{2}(\sigma_\beta^2 - \beta^2)$$

and the error reduction will be significant as long as $\sigma_\beta^2 \leq \beta^2$.

5 Correlated Classifier Combining

5.1 Introduction

The discussion so far focused on finding the types of combiners that improve performance. Yet, it is important to note that if the classifiers to be combined repeatedly provide the same (either erroneous or correct) classification decisions, there is little to be gained from combining, regardless of the chosen scheme. Therefore, the selection and training of the classifiers that will be combined is as critical an issue as the selection of the combining method. Indeed, classifier/data selection is directly tied to the amount of correlation among the various classifiers, which in turn affects the amount of error reduction that can be achieved.

The tie between error correlation and classifier performance was directly or indirectly observed by many researchers. For regression problems, Perrone and Cooper show that their combining results are weakened if the networks are not independent [49]. Ali and Pazzani discuss the relationship between error correlations and error reductions in the context of decision trees [2]. Meir discusses the effect of independence on combiner performance [41], and Jacobs reports that $N' \leq N$ independent classifiers are worth as much as N dependent classifiers [34]. The influence of the amount of training on ensemble performance is studied in [64]. For classification problems, the effect of the correlation among the classifier errors on combiner performance was quantified by the authors [70].

5.2 Combining Unbiased Correlated Classifiers

In this section we derive the explicit relationship between the correlation among classifier errors and the error reduction due to combining. Let us focus on the linear combination of unbiased classifiers. Without the independence assumption, the variance of $\tilde{\eta}_i$ is given by:

$$\sigma_{\tilde{\eta}_i}^2 = \frac{1}{N^2}\sum_{l=1}^{N}\sum_{m=1}^{N} cov(\eta_i^m(x), \eta_i^l(x))$$

This expression only considers the error that occur between classes i and j. In order to extend this expression to include all the boundaries, we introduce an overall correlation term δ. Then, the added error is computed in terms of δ. The correlation among classifiers is calculated using the following expression:

$$\delta = \sum_{i=1}^{L} P_i \, \delta_i \tag{42}$$

where P_i is the prior probability of class i. The correlation contribution of each class to the overall correlation, is proportional to the prior probability of that class.

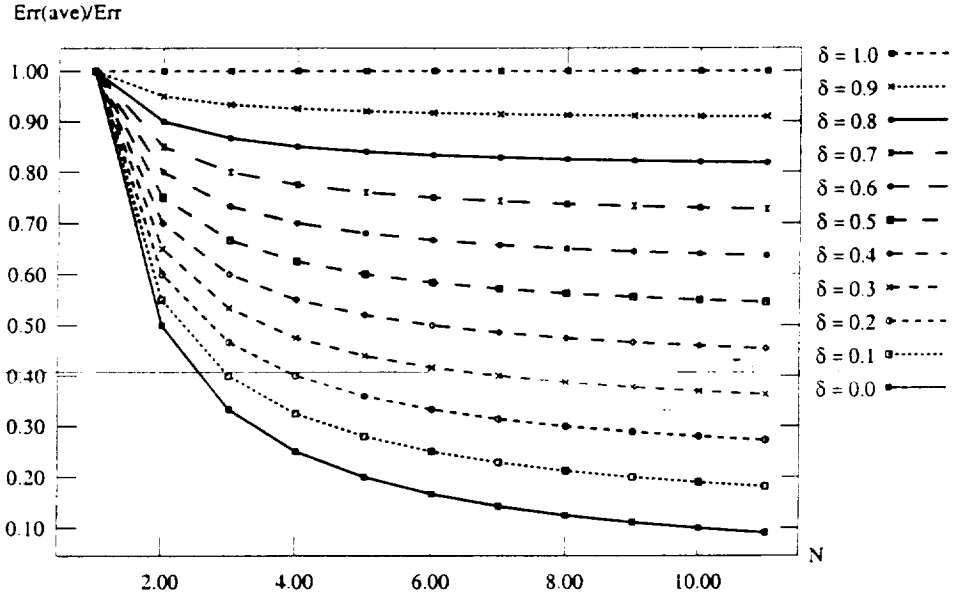

Figure 3: Error reduction ($\frac{E^{ave}_{add}}{E_{add}}$) for different classifier error correlations.

Let us now return to the error region analysis. With this formulation the first and second moments of b^{ave} yield: $M_1^{ave} = 0$, and $M_2^{ave} = \sigma^2_{b^{ave}}$. The derivation is identical to that of Section 3.1 and the only change is in the relation between σ_b^2 and $\sigma^2_{b^{ave}}$. We then get:

$$\begin{aligned} E^{ave}_{add} &= \frac{s M_2^{ave}}{2} = \frac{s}{2}\,\sigma^2_{b^{ave}} \\ &= \frac{s}{2}\,\sigma_b^2 \left(\frac{1 + \delta(N-1)}{N} \right) \\ &= E_{add} \left(\frac{1 + \delta(N-1)}{N} \right). \end{aligned} \tag{43}$$

The effect of the correlation between the errors of each classifier is readily apparent from Equation 43. If the errors are independent, then the second part of the reduction term vanishes and the combined error is reduced by N. If on the other hand, the error of each classifier has correlation 1, then the error of the combiner is equal to the initial errors and there is no improvement due to combining. Figure 3 shows how the variance reduction is affected by N and δ (using Equation 43).

Equation 49 shows the error reduction for correlated, biased classifiers. As long as the biases of individual classifiers are reduced by a larger amount than the correlated variances, the reduction will be similar to those in Section 5.2. However, if the biases are not reduced, the improvement gains will not be as significant. These results are conceptually identical to those obtained in Section 3, but vary in how the bias reduction z relates to N. In effect, the requirements on reducing z are lower than they were previously, since in the presence of bias, the error reduction is less than $\frac{1}{N}$. The practical implication of this observation is that, even in the presence of bias, the correlation dependent variance reduction term (given in Equation 43) will often be the limiting factor, and dictate the error reductions.

5.4 Discussion

In this section we established the importance of the correlation among the errors of individual classifiers in a combiner system. One can exploit this relationship explicitly by reducing the correlation among classifiers that will be combined. Several methods have been proposed for this purpose and many researchers are actively exploring this area [60].

Cross-validation, a statistical method aimed at estimating the "true' error [21, 65, 75], can also be used to control the amount of correlation among classifiers. By only training individual classifiers on overlapping subsets of the data, the correlation can be reduced. The various boosting algorithms exploit the relationship between corrlation and error rate by training subsequent classifiers on training patterns that have been "selected" by earlier classifiers [15, 13, 19, 59] thus reducing the correlation among them. Krogh and Vedelsky discuss how cross-validation can be used to improve ensemble performance [36]. Bootstrapping, or generating diferent training sets for each classifier by resampling the original set [17, 18, 35, 75], provides another method for correlation reduction [47]. Breiman also addresses this issue, and discusses methods aimed at reducing the correlation among estimators [9, 10]. Twomey and Smith discuss combining and resampling in the context of a 1-d regression problem [74]. The use of principal component regression to handle multi-collinearity while combining outputs of multiple regressors, was suggested in [42]. Another approach to reducing the correlation of classifiers can be found in input decimation, or in purposefully withholding some parts of each pattern from a given classifier [70]. Modifying the training of individual classifiers in order to obtain less correlated classifiers was also explored [56], and the selection of individual classifier through a genetic algorithm is suggested in [46].

In theory, reducing the correlation among classifiers that are combined increases the ensemble classification rates. In practice however, since each classifier uses a subset of the training data, individual classifier performance can deteriorate, thus offsetting any potential gains at the ensemble level [70]. It is therefore crucial to reduce the correlations without increasing the individual classifiers' error rates.

6 Experimental Combining Results

In order to provide in depth analysis and to demonstrate the result on public domain data sets, we have divided this section into two parts. First we will provide detailed experimental results on one difficult data set, outlining all the relevant design steps/parameters. Then we will summarize results on several public domain data sets taken from the UCI depository/Proben1 benchmarks [50].

Table 3: Combining Results for FS1.

Classifier(s)		Ave		Med		Max		Min	
	N	Error	σ	Error	σ	Error	σ	Error	σ
MLP	3	7.19	0.29	7.25	0.21	7.38	0.37	7.19	0.37
	5	7.13	0.27	7.30	0.29	7.32	0.41	7.20	0.37
	7	7.11	0.23	7.27	0.29	7.27	0.37	7.35	0.30
RBF	3	6.15	0.30	6.42	0.29	6.22	0.34	6.30	0.40
	5	6.05	0.20	6.23	0.18	6.12	0.34	6.06	0.39
	7	5.97	0.22	6.25	0.20	6.03	0.35	5.92	0.31
BOTH	3	6.11	0.34	6.02	0.33	6.48	0.43	6.89	0.29
	5	6.11	0.31	5.76	0.29	6.59	0.40	6.89	0.24
	7	6.08	0.32	5.67	0.27	6.68	0.41	6.90	0.26

Table 4: Combining Results for FS2.

Classifier(s)		Ave		Med		Max		Min	
	N	Error	σ	Error	σ	Error	σ	Error	σ
MLP	3	9.32	0.35	9.47	0.47	9.64	0.47	9.39	0.34
	5	9.20	0.30	9.22	0.30	9.73	0.44	9.27	0.30
	7	9.07	0.36	9.11	0.29	9.80	0.48	9.25	0.36
RBF	3	10.55	0.45	10.49	0.42	10.59	0.57	10.74	0.34
	5	10.43	0.30	10.51	0.34	10.55	0.40	10.65	0.37
	7	10.44	0.32	10.46	0.31	10.58	0.43	10.66	0.39
BOTH	3	8.46	0.57	9.20	0.49	8.65	0.47	9.56	0.53
	5	8.17	0.41	8.97	0.54	8.71	0.36	9.50	0.45
	7	8.14	0.28	8.85	0.45	8.79	0.40	9.40	0.39

- different classifiers trained with a single feature set (fifth and sixth rows);
- single classifier trained on two different feature sets (seventh and eighth rows).

There is a striking similarity between these correlation results and the improvements obtained through combining. When different runs of a single classifier are combined using only one feature set, the combining improvements are very modest. These are also the cases where the classifier correlation coefficients are the highest. Mixing different classifiers reduces the correlation, and in most cases, improves the combining results. The most drastic improvements are obtained when two qualitatively different feature sets are used, which are also the cases with the lowest classifier correlations.

6.2 Proben1 Benchmarks

In this section, examples from the Proben1 benchmark set[9] are used to study the benefits of combining [50]. Table 7 shows the test set error rate for both the MLP and the RBF classifiers on six different data sets taken from the Proben1 benchmarks[10].

[9] Available from: *ftp://ftp.ira.uka.de/pub/papers/techreports/1994/1994-21.ps.Z*.

[10] These Proben1 results correspond to the "pivot" and "no-shortcut" architectures (A and B respectively), reported in [50]. The large error in the Proben1 no-shortcut architecture for the SOYBEAN1 problem is not explained.

Table 8: Combining Results for CANCER1.

Classifier(s)		Ave		Med		Max		Min	
	N	Error	σ	Error	σ	Error	σ	Error	σ
MLP	3	0.60	0.13	0.63	0.17	0.66	0.21	0.66	0.21
	5	0.60	0.13	0.58	0.00	0.63	0.17	0.63	0.17
	7	0.60	0.13	0.58	0.00	0.60	0.13	0.60	0.13
RBF	3	1.29	0.48	1.12	0.53	1.90	0.52	0.95	0.42
	5	1.26	0.47	1.12	0.47	1.81	0.58	0.98	0.37
	7	1.32	0.41	1.18	0.43	1.81	0.53	0.89	0.34
BOTH	3	0.86	0.39	0.63	0.18	1.03	0.53	0.95	0.42
	5	0.72	0.25	0.72	0.25	1.38	0.43	0.83	0.29
	7	0.86	0.39	0.58	0.00	1.49	0.39	0.83	0.34

Table 9: Combining Results for CARD1.

Classifier(s)		Ave		Med		Max		Min	
	N	Error	σ	Error	σ	Error	σ	Error	σ
MLP	3	13.37	0.45	13.61	0.56	13.43	0.44	13.40	0.47
	5	13.23	0.36	13.40	0.39	13.37	0.45	13.31	0.40
	7	13.20	0.26	13.29	0.33	13.26	0.35	13.20	0.32
RBF	3	13.40	0.70	13.58	0.76	14.01	0.66	13.08	1.05
	5	13.11	0.60	13.29	0.67	13.95	0.66	12.88	0.98
	7	13.02	0.33	12.99	0.33	13.75	0.76	12.82	0.67
BOTH	3	13.75	0.69	13.69	0.70	13.49	0.62	13.66	0.70
	5	13.78	0.55	13.66	0.67	13.66	0.65	13.75	0.64
	7	13.84	0.51	13.52	0.58	13.66	0.60	13.72	0.70

The CARD1 data set consists of credit approval decisions [51, 52]. 51 inputs are used to determine whether or not to approve the credit card application of a customer. There are 690 examples in this set, and 345 are used for training. The MLP has one hidden layer with 20 units, and the RBF network has 20 kernels.

The DIABETES1 data set is based on personal data of the Pima Indians obtained from the National Institute of Diabetes and Digestive and Kidney Diseases [63]. The binary output determines whether or not the subjects show signs of diabetes according to the World Health Organization. The input consists of 8 attributes, and there are 768 examples in this set, half of which are used for training. MLPs with one hidden layer with 10 units, and RBF networks with 10 kernels are selected for this data set.

The GENE1 is based on intron/exon boundary detection, or the detection of splice junctions in DNA sequences [45, 66]. 120 inputs are used to determine whether a DNA section is a donor, an acceptor or neither. There are 3175 examples, of which 1588 are used for training. The MLP architecture consists of a single hidden layer network with 20 hidden units. The RBF network has 10 kernels.

The GLASS1 data set is based on the chemical analysis of glass splinters. The 9 inputs are used to classify 6 different types of glass. There are 214 examples in this set, and 107 of them are used for training. MLPs with a single hidden layer of 15 units, and RBF networks with 20 kernels are

Table 12: Combining Results for GLASS1

Classifier(s)		Ave		Med		Max		Min	
	N	Error	σ	Error	σ	Error	σ	Error	σ
MLP	3	32.07	0.00	32.07	0.00	32.07	0.00	32.07	0.00
	5	32.07	0.00	32.07	0.00	32.07	0.00	32.07	0.00
	7	32.07	0.00	32.07	0.00	32.07	0.00	32.07	0.00
RBF	3	29.81	2.28	30.76	2.74	30.28	2.02	29.43	2.89
	5	29.25	1.84	30.19	1.69	30.85	2.00	28.30	2.46
	7	29.06	1.51	30.00	1.88	31.89	1.78	27.55	1.83
BOTH	3	30.66	2.52	29.06	2.02	33.87	1.74	29.91	2.25
	5	32.36	1.82	28.30	1.46	33.68	1.82	29.72	1.78
	7	32.45	0.96	27.93	1.75	34.15	1.68	29.91	1.61

Table 13: Combining Results for SOYBEAN1.

Classifier(s)		Ave		Med		Max		Min	
	N	Error	σ	Error	σ	Error	σ	Error	σ
MLP	3	7.06	0.00	7.09	0.13	7.06	0.00	7.85	1.42
	5	7.06	0.00	7.06	0.00	7.06	0.00	8.38	1.63
	7	7.06	0.00	7.06	0.00	7.06	0.00	8.38	1.68
RBF	3	7.74	0.47	7.65	0.42	7.85	0.47	7.77	0.44
	5	7.62	0.23	7.68	0.30	7.77	0.30	7.65	0.42
	7	7.68	0.23	7.82	0.33	7.68	0.29	7.59	0.45
BOTH	3	7.18	0.23	7.12	0.17	7.56	0.28	7.85	1.27
	5	7.18	0.23	7.12	0.17	7.50	0.25	8.06	1.22
	7	7.18	0.24	7.18	0.23	7.50	0.25	8.09	1.05

in most cases. If the combined bias is not lowered, the combiner will not outperform the better classifier. Second, as discussed in section 5.2, the correlation plays a major role in the final reduction factor. There are no guarantees that using different types of classifiers will reduce the correlation factors. Therefore, the combining of different types of classifiers, especially when their respective performances are significantly different (the error rate for the RBF network on the CANCER1 data set is over twice the error rate for MLPs) has to be treated with caution.

Determining which combiner (e.g. *ave* or *med*), or which classifier selection (e.g. multiple MLPs or MLPs and RBFs) will perform best in a given situation is not generally an easy task. However, some information can be extracted from the experimental results. The linear combiner, for example, appears more compatible with the MLP classifiers than with the RBF networks. When combining two types of network, the *med* combiner often performs better than other combiners. One reason for this is that the outputs that will be combined come from different sources, and selecting the largest or smallest value can favor one type of network over another. These results emphasize the need for closely coupling the problem at hand with a classifier/combiner. There does not seem to be a single type of network or combiner that can be labeled "best" under all circumstances.

overtraining, but not undertraining (except in cases where the undertraining is very mild). This corroborates well with the theoretical framework which shows combining to be more effective at variance reduction than bias reduction.

The classification rates obtained by the order statistics combiners in section 6 are in general, comparable to those obtained by averaging. The advantage of OS approaches should be more evident in situations where there is substantial variability in the performance of individual classifiers, and the thus robust properties of OS combining can be brought to bear upon. Such variability in individual performance may be due to, for example, the classifiers being geographically distributed and working only on locally available data of highly varying quality. Current work by the authors indicate that this is indeed the case, but the issue needs to be examined in greater detail.

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.

Acknowledgements: This research was supported in part by AFOSR contract F49620-93-1-0307, NSF grant ECS 9307632, and ARO contracts DAAH 04-94-G0417 and 04-95-10494.

References

[1] K. Al-Ghoneim and B. V. K. Vijaya Kumar. Learning ranks with neural networks (Invited paper). In *Applications and Science of Artificial Neural Networks, Proceedings of the SPIE*, volume 2492, pages 446–464, April 1995.

[2] K. M. Ali and M. J. Pazzani. On the link between error correlation and error reduction in decision tree ensembles. Technical Report 95-38, Department of Information and Computer Science, University of California, Irvine, 1995.

[3] B.C. Arnold, N. Balakrishnan, and H.N. Nagaraja. *A First Course in Order Statistics*. Wiley, New York, 1992.

[4] J.A. Barnett. Computational methods for a mathematical theory of evidence. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 868–875, August 1981.

[5] R. Battiti and A. M. Colla. Democracy in neural nets: Voting schemes for classification. *Neural Networks*, 7(4):691–709, 1994.

[6] W. G. Baxt. Improving the accuracy of an artificial neural network using multiple differently trained networks. *Neural Computation*, 4:772–780, 1992.

[7] J.A. Benediktsson, J.R. Sveinsson, O.K. Ersoy, and P.H. Swain. Parallel consensual neural networks with optimally weighted outputs. In *Proceedings of the World Congress on Neural Networks*, pages III:129–137. INNS Press, 1994.

[8] V. Biou, J.F. Gibrat, J.M. Levin, B. Robson, and J. Garnier. Secondary structure prediction: combination of three different methods. *Protein Engineering*, 2:185–91, 1988.

[26] J. Ghosh and K. Turner. Structural adaptation and generalization in supervised feedforward networks. *Journal of Artificial Neural Networks*, 1(4):431–458, 1994.

[27] J. Ghosh, K. Turner, S. Beck, and L. Deuser. Integration of neural classifiers for passive sonar signals. In C.T. Leondes, editor, *Control and Dynamic Systems—Advances in Theory and Applications*, volume 77, pages 301–338. Academic Press, 1996.

[28] C. W. J. Granger. Combining forecasts—twenty years later. *Journal of Forecasting*, 8(3):167–173, 1989.

[29] J.B. Hampshire and A.H. Waibel. The Meta-Pi network: Building distributed representations for robust multisource pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(7):751–769, 1992.

[30] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1000, 1990.

[31] S. Hashem and B. Schmeiser. Approximating a function and its derivatives using MSE-optimal linear combinations of trained feedforward neural networks. In *Proceedings of the Joint Conference on Neural Networks*, volume 87, pages I:617–620, New Jersey, 1993.

[32] 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.

[33] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–76, 1994.

[34] Robert Jacobs. Method for combining experts' probability assessments. *Neural Computation*, 7(5):867–888, 1995.

[35] A. Jain, R. Dubes, and C. Chen. Bootstrap techniques for error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:628–633, 1987.

[36] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation and active learning. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems-7*, pages 231–238. M.I.T. Press, 1995.

[37] J. Lee, J.-N. Hwang, D.T. Davis, and A.C. Nelson. Integration of neural networks and decision tree classifiers for automated cytology screening. In *Proceedings of the International Joint Conference on Neural Networks, Seattle*, pages I:257–262, July 1991.

[38] 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.

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

[40] O. L. Mangasarian, R. Setiono, and W. H. Wolberg. Pattern recognition via linear programming: Theory and application to medical diagnosis. In Thomas F. Coleman and Yuying Li, editors, *Large-Scale Numerical Optimization*, pages 22–30. SIAM Publications, 1990.

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

[56] B. Rosen. Ensemble learning using decorrelated neural networks. *Connection Science, Special Issue on Combining Artificial Neural Networks: Ensemble Approaches*, 8(3 & 4):373–384, 1996.

[57] D. W. Ruck, S. K. Rogers, M. E. Kabrisky, M. E. Oxley, and B. W. Suter. The multilayer Perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1(4):296–298, 1990.

[58] A. E. Sarhan and B. G. Greenberg. Estimation of location and scale parameters by order statistics from singly and doubly censored samples. *Annals of Mathematical Statistics Science*, 27:427–451, 1956.

[59] R. Schapire, Y. Freund, P. Bartlett, and Lee W.S. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann, 1997.

[60] A. J. J. Sharkey. (editor). *Connection Science: Special Issue on Combining Artificial Neural Networks: Ensemble Approaches*, 8(3 & 4), 1996.

[61] S. Shlien. Multiple binary decision tree classifiers. *Pattern Recognition*, 23(7):757–63, 1990.

[62] 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.

[63] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications and Medical Care*, pages 261–265. IEEE Computer Society Press, 1988.

[64] P. Sollich and A. Krogh. Learning with ensembles: How overfitting can be useful. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems-8*, pages 190–196. M.I.T. Press, 1996.

[65] M. Stone. Cross-validatory choice and assessment of statistical prediction. *Journal of the Royal Statistical Society*, 36:111–147, 1974.

[66] G. G. Towell and J. W. Shavlik. Interpretation of artificial neural networks: Mapping knowledge-based neural networks into rules. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems-4*, pages 977–984. Morgan Kaufmann, 1992.

[67] K. Tumer and J. Ghosh. Limits to performance gains in combined neural classifiers. In *Proceedings of the Artificial Neural Networks in Engineering '95*, pages 419–424, St. Louis, 1995.

[68] K. Tumer and J. Ghosh. Order statistics combiners for neural classifiers. In *Proceedings of the World Congress on Neural Networks*, pages I:31–34, Washington D.C., 1995. INNS Press.

[69] K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341–348, February 1996.