

# Robust Combining of Disparate Classifiers through Order Statistics\*

**Kagan Tumer**

Computational Sciences Div.  
NASA Ames Research Center  
MS 269-4, Moffett Field, CA, 94035  
kagan@ptolemy.arc.nasa.gov

**Joydeep Ghosh**

Dept of Elec. and Comp. Engr.  
The University of Texas  
Austin, TX 78712-1084  
ghosh@ece.utexas.edu

November 1, 2001

## Abstract

Integrating the outputs of multiple classifiers via combiners or meta-learners has led to substantial improvements in several difficult pattern recognition problems. In this article we investigate a family of combiners based on order statistics, for robust handling of situations where there are large discrepancies in performance of individual classifiers. Based on a mathematical modeling of how the decision boundaries are affected by order statistic combiners, we derive expressions for the reductions in error expected when simple output combination methods based on the the median, the maximum and in general, the  $i^{th}$  order statistic, are used. Furthermore, we analyze the trim and spread combiners, both based on linear combinations of the ordered classifier outputs, and show that in the presence of uneven classifier performance, they often provide substantial gains over both linear and simple order statistics combiners. Experimental results on both real world data and standard public domain data sets corroborate these findings.

*Keywords:* Ensembles, order statistics, trimmed means, classification error, robust statistics.

## 1 Introduction

Since different types of classifiers have different “inductive bias”, one does not expect the generalization performance of two classifiers to be identical [14, 16] for difficult pattern recognition problems, even when they are both trained on the same data set. If only the “best” classifier is selected based on an *estimation* of the true generalization performance using a finite test set valuable information contained in the results of the discarded classifiers may be lost. Such potential loss of information can be avoided if the outputs of all available classifiers are used in the final classification decision. This concept has received a great deal of attention recently, and many methods for combining

---

\*To appear in *Pattern Analysis and Applications* special issue on “Fusion of Multiple Classifiers.”

classifier outputs have been proposed [15, 17, 19, 24, 29]. Furthermore, promoting diversity among classifiers prior to combining forms the basis of many strategies, including bagging, arcing, boosting and correlation control [6, 31].

Approaches to pooling classifiers can be separated into two main categories: (i) simple combiners, e.g., voting [3], Bayesian based weighted product rule [22], or averaging [24, 30], and, (ii) meta-learners, such as arbitration [7] or stacking [34]. 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 more susceptible to outliers and to unevenly performing classifiers. In the second category, either sets of combining rules, or full fledged classifiers acting on the outputs of the individual classifiers, are constructed [1, 20, 34]. This type of combining is more general, but is vulnerable to all the problems associated with the added learning (e.g., overfitting, lengthy training time).

An implicit assumption in most combining schemes is that each classifier sees the same training data or resampled versions of the same data. If the individual classifiers are then appropriately chosen and trained properly, their performances will be (relatively) comparable in any region of the problem space. So gains from combining are derived from the diversity among classifiers rather than by compensating for weak members of the pool (i.e., variance reduction) [11]. However, there are situations where individual classifiers may not have access to the same data. Such conditions arise in certain data mining, sensor fusion and electrical logging (oil services) problems where there are large variabilities in the data which is acquired locally and needs to be processed in (near) real time at geographically separated places [9]. These conditions create a pool of classifiers that may have significant variations in their overall performance. Moreover, they may lead to conditions where individual classifiers have similar *average* performance, but substantially different performance over different parts of the input space.

In such cases, combining is still desirable, but neither simple combiners nor meta-learners are particularly well-suited for the type of problems that arise. For example, the simplicity of averaging the classifier outputs is appealing, but the prospect of one poor classifier corrupting the ensemble makes this a risky choice. Weighted averaging of classifier outputs appears to provide some flexibility [18, 23]. Unfortunately, the weights are still assigned on a per classifier basis rather than a per sample or per class basis. If a classifier is accurate only in certain areas of the input space, this scheme fails to take advantage of the variable accuracy of the classifier in question. Using a meta learner that provides different weights for different patterns can potentially solve this problem, but at a considerable cost. In particular, the off-line training of a meta-learner using substantial amount of data outputted by geographically distributed classifiers, may not be feasible. In addition to providing robustness, the order statistic combiners presented in this work also aim at bridging the gap between simplicity and generality by allowing the flexible selection of classifiers without the associated cost of training meta-classifiers.

Section 2 summarizes the relationship between classifier errors and decision boundaries and provides the necessary background for mathematically analyzing order statis-

tic combiners [30, 32]. Section 3 introduces simple order statistic combiners. Based on these concepts, in Section 4 we investigate two powerful combiners, *trim* and *spread*, and derive the amount of error reduction associated with each. In Section 5 we present the performance of order statistic combiners on a real world sonar problem [15], and several data sets from the Proben1/UCI benchmarks [4, 25]. Section 6 discusses the implications of using linear combinations of order statistics as a strategy for pooling the outputs of individual classifiers.

## 2 Background

In this section we first summarize the approach in [30, 32]<sup>1</sup>, where a framework to quantify the effect of inaccuracies in estimating *a posteriori* class probabilities on the classification error was introduced. This background is needed to characterize and understand the impact of order statistics combiners, as described in Sections 3 and 4. It also introduces the necessary notations and definitions. Then we briefly review the basic concepts and properties of order statistics.

### 2.1 Relationship Between *a Posteriori* Probability Estimates and Classification Error for a Single Classifier

A wide variety of classification models not only provide the suggested class label for a given input, but an estimation of the posterior class probabilities for that input as well. For example, it is well known that, given *one-of-L* desired outputs and sufficient training, the outputs of a multilayered perceptron network or a radial basis function network based classifier trained to minimize a mean square error criteria, approximate the *a posteriori* probabilities of the corresponding classes [26]. This result is based on the universal approximation capabilities of the underlying function approximators.

For such classification models, one can represent the *i*th output (corresponding to class *i*) of the classifier as:

$$f_i(x) = p_i(x) + \epsilon_i(x), \quad (1)$$

where  $p_i(x)$  is the true posterior for *i*th class on input  $x$ , i.e.,  $p_i(x) = P(C_i|x)$ , and  $\epsilon_i(x)$  is the error of the classifier in estimating that posterior.

Figure 1 illustrates the true (solid lines) and approximated (dashed lines) posterior probabilities for classes *i* and *j*, given a one dimensional input,  $x$ . From Bayes decision theory, the ideal class boundary is  $x^*$ , where  $p_i(x) = p_j(x)$ . The realized boundary is  $x_b$ , where  $f_i(x) = f_j(x)$ . The Bayes error rate is related to the lightly shaded region, while the *extra classification error*, (called model error or  $E_{model}$ ), incurred because of our imperfect classifier, is determined by the darkly shaded region, whose size is governed by  $b$ , the offset between  $x^*$  and  $x_b$ . To quantify this relationship, we first break down the posterior probability approximation error in Eq. 1 into two parts:

$$\epsilon_i(x) = \beta_i + \eta_i(x). \quad (2)$$

---

<sup>1</sup>These and other related papers can be downloaded from URL <http://www.lans.ece.utexas.edu>.

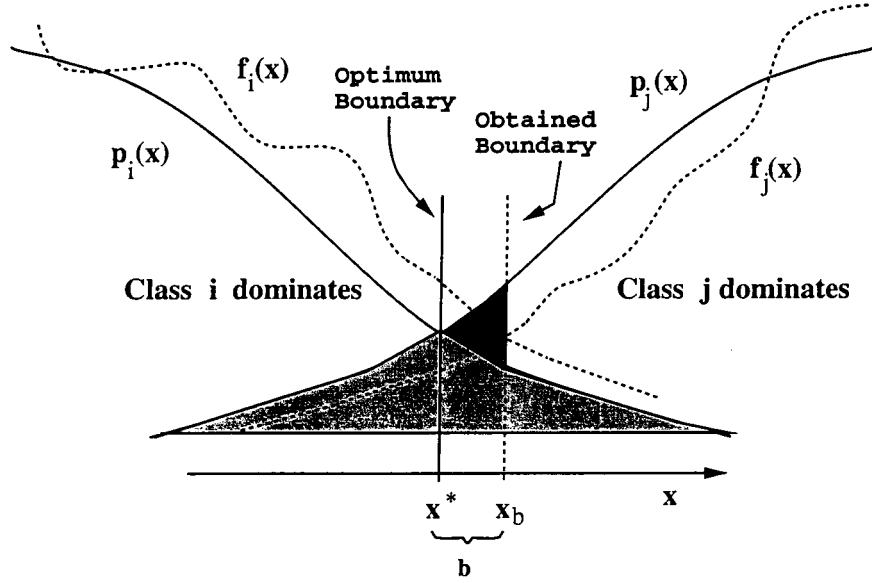


Figure 1: Error regions associated with approximating the *a posteriori* probabilities [30].

The first component is the bias or offset in estimating the aposterior probability of class  $i$ , and does not vary with the input. The second component gives the variability from the systematic offset error, for different input samples, and has zero mean and variance  $\sigma_{\eta_i}^2$ . These two components of the error are similar to the bias and variance decomposition for a quadratic loss function given in [14], although they are at the individual input level. We will therefore refer to classifiers as “biased” and “unbiased” implying  $\beta_k \neq 0$  for some  $k$ , and  $\beta_k = 0, \forall k$ , respectively.

Now, the boundary offset ( $b = x_b - x^*$ ) is a random variable with probability density function  $p_b(x)$ , whose mean and variance are given by:

$$\beta = \frac{\beta_i - \beta_j}{s}, \quad (3)$$

and

$$\sigma_b^2 = \frac{\sigma_{\eta_i}^2 + \sigma_{\eta_j}^2}{s^2}, \quad (4)$$

as derived in [30, 32]. Also, let the first and second moments of the boundary offset  $b$  be represented by

$$\mathcal{M}_1 = \int_{-\infty}^{\infty} x p_b(x) dx \quad \text{and} \quad \mathcal{M}_2 = \int_{-\infty}^{\infty} x^2 p_b(x) dx .$$

By integrating over the densely shared area of Fig. 1 weighted by  $p_b(x)$ , one can show that the extra model error can be simply expressed as:

$$E_{model} = \frac{s \mathcal{M}_2}{2} = \frac{s}{2} (\sigma_b^2 + \beta^2). \quad (5)$$

Note that if the classifier is not biased, the second term will drop out. To emphasize the distinction between biased and unbiased classifiers, the model error will be given as a function of  $\beta$  for biased classifiers in Section 3.

The above framework sets the stage for evaluating the effects of combining multiple classifiers. In [30] we studied what happens to the model error,  $E_{model}^{ave}$  of an *averaging* combiner, where for each class  $i$ , the approximated posterior probabilities  $f_i^m(x)$ ,  $1 \leq m \leq N$  of  $N$  individual classifiers are averaged, and then the class with the highest value of this average is chosen as the winner. In this paper, we perform a somewhat more involved derivation of what happens to the model error  $E_{model}^{os}$ , when the combination is based on order statistics rather than simple averaging.

## 2.2 Background on Order Statistics

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

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

The  $i$ th order statistic denoted by  $X_{i:N}$ , is the  $i$ th value in this progression. The cumulative distribution function for the smallest and largest order statistic can be obtained by noting that:

$$F_{X_{N:N}}(x) = P(X_{N:N} \leq x) = \prod_{i=1}^N P(X_{i:N} \leq x) = [F_X(x)]^N$$

and:

$$\begin{aligned} F_{X_{1:N}}(x) &= P(X_{1:N} \leq x) = 1 - P(X_{1:N} \geq x) = 1 - \prod_{i=1}^N P(X_{i:N} \geq x) \\ &= 1 - \prod_{i=1}^N (1 - P(X_{i:N} \leq x)) = 1 - [1 - F_X(x)]^N \end{aligned}$$

The corresponding probability density functions can be obtained from these equations. In general, for the  $i$ th order statistic, the cumulative distribution function gives the probability that exactly  $i$  of the chosen  $X$ 's are less than or equal to  $x$ . The probability density function of  $X_{i:N}$  is then given by [10]:

$$p_{X_{i:N}}(x) = \frac{N!}{(i-1)!(N-i)!} [F_X(x)]^{i-1} [1 - F_X(x)]^{N-i} p_X(x). \quad (6)$$

This general form however, cannot always be computed in closed form. Therefore, obtaining the expected value of a function of  $x$  using Equation 6 is not always possible. However, the first two moments of the density function are widely available for a variety of distributions [2]. These moments can be used to compute the expected values of certain specific functions, e.g., polynomials of order less than two.

## 3 Order Statistics Ensembles

Now, let us turn our attention to designing an ensemble of  $N$  classifiers using order statistics (OS) concepts for combining the outputs. For a given input  $x$ , let the outputs of each of the  $N$  classifiers for each class  $i$  be ordered in the following manner:

$$f_i^{1:N}(x) \leq f_i^{2:N}(x) \leq \dots \leq f_i^{N:N}(x).$$

Then one constructs the  $k$ th order statistic combiner, by selecting the  $k$ th ranked output for each class ( $f_i^{k:N}(x)$ ), as representing its posterior.

In particular, *max*, *med* and *min* combiners are defined as follows:

$$f_i^{max}(x) = f_i^{N:N}(x), \quad (7)$$

$$f_i^{med}(x) = \begin{cases} \frac{f_i^{\frac{N}{2}:N}(x) + f_i^{\frac{N}{2}+1:N}(x)}{2} & \text{if } N \text{ is even} \\ f_i^{\frac{N+1}{2}:N}(x) & \text{if } N \text{ is odd,} \end{cases} \quad (8)$$

$$f_i^{min}(x) = f_i^{1:N}(x). \quad (9)$$

These three combiners are relevant because they represent important qualitative interpretations of the output space. Selecting the maximum combiner is equivalent to selecting the class with the highest posterior. Indeed, since the network outputs approximate the class *a posteriori* distributions, selecting the maximum reduces to selecting the classifier that is the most “certain” of its decision. The drawback of this method however is that it can be compromised by a single classifier that repeatedly provides high values. The selection of the minimum combiner follows a similar logic, but focuses on classes that are unlikely to be correct, rather than on the correct class. Thus, this combiner eliminates less likely classes by basing the decision on the lowest value for a given class. This combiner suffers from the same ills as the *max* combiner. However, it is less dependent on a single error, since it performs a min-max operation, rather than a max-max<sup>2</sup>. The median classifier 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 that follows does not depend on the particular order statistic chosen. Therefore, we will denote all OS combiners by  $f_k^{os}(x)$  and derive the model error,  $E_{model}^{os}$ . The network output provided by  $f_k^{os}(x)$  is given by:

$$f_k^{os}(x) = p_k(x) + \epsilon_k^{os}(x), \quad (10)$$

We shall first analyze the case when there is no bias, and then consider the more involved situation when at least one classifier provides a biased estimate of the *a posteriori* class probabilities.

### 3.1 Combining Unbiased Classifiers through Order Statistics

For the zero-bias case ( $\beta_k = 0, \forall k$ ), we get  $\epsilon_k^{os}(x) = \eta_k^{os}(x)$ . Proceeding as in Section 2, the boundary  $b^{os}$  is shown to be:

$$b^{os} = \frac{\eta_i^{os}(x_b) - \eta_j^{os}(x_b)}{s}. \quad (11)$$

For *i.i.d.*  $\eta_k$ 's, the first two moments will be identical for each class. Moreover, taking the order statistic will shift the mean of both  $\eta_i^{os}$  and  $\eta_j^{os}$  by the same amount, leaving

<sup>2</sup>Recall that the pattern is ultimately assigned to the class with the highest combined output.

the mean of the difference unaffected. Therefore,  $b^{os}$  will have zero mean, and variance:

$$\sigma_{b^{os}}^2 = \frac{2 \sigma_{\eta_k^{os}}^2}{s^2} = \frac{2 \alpha \sigma_{\eta_k}^2}{s^2} = \alpha \sigma_b^2, \quad (12)$$

where  $\alpha$  is a reduction factor that depends on the order statistic and on the distribution of  $b$ . For most distributions,  $\alpha$  can be found in tabulated form [2]. For example, Table 1 provides  $\alpha$  values for all order statistic combiners, up to 10 classifiers, for a Gaussian distribution [2, 27]. (Because this distribution is symmetric, the  $\alpha$  values of  $l$  and  $k$  where  $l + k = N + 1$  are identical, and listed in parenthesis).

Returning to the error calculation, we have:  $\mathcal{M}_1^{os} = 0$ , and  $\mathcal{M}_2^{os} = \sigma_{b^{os}}^2$ , providing:

$$E_{model}^{os} = \frac{s \mathcal{M}_2^{os}}{2} = \frac{s \sigma_{b^{os}}^2}{2} = \frac{s \alpha \sigma_b^2}{2} = \alpha E_{model}. \quad (13)$$

Table 1: Reduction factors  $\alpha$  for the Gaussian Distribution, based on [27].

N	k	$\alpha$	N	k	$\alpha$	N	k	$\alpha$
1	1	1.00	6	2 (5)	.280		1 (9)	.357
2	1 (2)	.682		3 (4)	.246		2 (8)	.226
3	1 (3)	.560		1 (7)	.392	9	3 (7)	.186
	2	.449	7	2 (6)	.257		4 (6)	.171
4	1 (4)	.492		3 (5)	.220		5	.166
	2 (3)	.360		4	.210		1 (10)	.344
	1 (5)	.448		1 (8)	.373		2 (9)	.215
5	2 (4)	.312	8	2 (7)	.239	10	3 (8)	.175
	3	.287		3 (6)	.201		4 (7)	.158
6	1 (6)	.416		4 (5)	.187		5 (6)	.151

Equation 13 shows that the reduction in the error due to using the OS combiner instead of the  $m$ th classifier 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.

### 3.2 Combining Biased Classifiers through Order Statistics

In this section, we analyze the error regions for biased classifiers. Let us return our attention to  $b^{os}$ . First, note that the error terms can no longer be studied separately, 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<sup>3</sup>. Equation 11 becomes:

$$b^{os} = \frac{(\beta_i + \eta_i(x_b))^{os} - (\beta_j + \eta_j(x_b))^{os}}{s}. \quad (14)$$

Let  $\bar{\beta}_k = \frac{1}{N} \sum_{m=1}^N \beta_k$  be the mean of classifier biases. Since  $\eta_k$ 's have zero-mean,  $\beta_k + \eta_k(x_b)$  has first moment  $\bar{\beta}_k$  and variance  $\sigma_{\eta_k}^2 + \sigma_{\beta_k}^2$ , with  $\sigma_{\beta_k}^2 = E[(\beta_k)^2] - \bar{\beta}_k^2$ , where  $E[\cdot]$  denotes the expected value operator.

<sup>3</sup>Since the exact distribution parameters of  $b^{os}$  are not known, we use the sample mean and the sample variance.

Taking a specific order statistic of Equation 14 will modify both moments. The first moment is given by  $\bar{\beta}_k + \mu^{os}$ , where  $\mu^{os}$  is a shift which depends on the order statistic chosen, but not on the class. Then, the first moment of  $b^{os}$  is given by:

$$\frac{(\bar{\beta}_i + \mu^{os}) - (\bar{\beta}_j + \mu^{os})}{s} = \frac{\bar{\beta}_i - \bar{\beta}_j}{s} = \bar{\beta}. \quad (15)$$

Note that the bias term represents an ‘‘average bias’’ since the contributions due to the order statistic are removed. Therefore, reductions in bias cannot be obtained from a table similar to Table 1.

Now, let us turn our attention to the variance. Since  $\beta_k + \eta_k(x_b)$  has variance  $\sigma_{\eta_k}^2 + \sigma_{\beta_k}^2$ , it follows that  $(\beta_k + \eta_k(x_b))^{os}$  has variance  $\sigma_{\eta_k}^{2os} = \alpha(\sigma_{\eta_k}^2 + \sigma_{\beta_k}^2)$ , where  $\alpha$  is the factor discussed in Section 3.1. Therefore, the variance of  $b^{os}$  is given by:

$$\begin{aligned} \sigma_{b^{os}}^2 &= \frac{\sigma_{\eta_i}^{2os} + \sigma_{\eta_j}^{2os}}{s^2} = \frac{2\alpha\sigma_{\eta_i}^2}{s^2} + \frac{\alpha(\sigma_{\beta_i}^2 + \sigma_{\beta_j}^2)}{s^2} \\ &= \alpha(\sigma_b^2 + \sigma_\beta^2), \end{aligned} \quad (16)$$

where  $\sigma_\beta^2 = \frac{\sigma_{\beta_i}^2 + \sigma_{\beta_j}^2}{s^2}$  is the variance introduced by the systematic errors of different classifiers.

We have now obtained the first and second moments of  $b^{os}$ , and can compute the model error. Namely, we have  $\mathcal{M}_1^{os} = \bar{\beta}$  and  $\sigma_{b^{os}}^2 = \mathcal{M}_2^{os} - (\mathcal{M}_1^{os})^2$ , leading to:

$$E_{model}^{os}(\beta) = \frac{s}{2} \mathcal{M}_2^{os} = \frac{s}{2} (\sigma_{b^{os}}^2 + \bar{\beta}^2) \quad (17)$$

$$= \frac{s}{2} (\alpha(\sigma_b^2 + \sigma_\beta^2) + \bar{\beta}^2). \quad (18)$$

The reduction in the error is more difficult to assess in this case. By writing the error as:

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

we get:

$$E_{model}^{os}(\beta) = \alpha E_{model}(\beta) + \frac{s}{2} (\alpha\sigma_\beta^2 + \bar{\beta}^2 - \alpha(\beta)^2). \quad (19)$$

Analyzing the error reduction in the general case requires knowledge about the bias introduced by each classifier. Unlike regression problems where the bias and variance contributions to the error are additive and well-understood, in classification problems their interaction is more complex [13]. Indeed it has been observed that ensemble methods do more than simply reduce the variance [28].

Based on these observations and Equation 19, let us analyze extreme cases. For example, if each classifier has the same bias,  $\sigma_\beta^2$  is reduced to zero and  $\bar{\beta} = \beta$ . In this case the error reduction can be expressed as:

$$E_{model}^{os}(\beta) = \frac{s}{2} (\alpha\sigma_b^2 + (\beta)^2) = \alpha E_{model}(\beta) + \frac{s(1-\alpha)}{2} (\beta)^2,$$

where  $\alpha$  balances the two contributions to the error. A small value for  $\alpha$  will reduce the first component of the error (mainly variance), while leaving the second term untouched.



The net effect will be very similar to results obtained for regression problems. 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, we obtain  $\bar{\beta} = 0$ . In this case, the model error becomes:

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

and the error reduction will be significant if the second term is small or negative. In fact, if the variation among the biases is small relative to their magnitude, the error will be reduced more than in the unbiased cases. If however, the variation is large compared to the magnitude, the error reduction will be minimal. Furthermore, if  $\alpha$  is large and the biases are small and highly varied, it is possible for this combiner to do worse than the individual classifiers, which is a danger not present for regression problems. This observation very closely parallels results reported in [13].

## 4 Linear Combining of Ordered Classifier Outputs

In the previous section, we derived error reductions when the class posteriors are directly estimated through the ordered classifier outputs. Since simple averaging has also been shown to provide benefits, in this section, we investigate the combinations of averaging and order statistics for pooling classifier outputs.

### 4.1 Spread Combiner

The first linear combination of ordered classifier outputs we study focuses on extrema. As discussed in Section 3.1, the maximum and minimum of a set of classifier outputs carry specific meanings. Indeed, the maximum can be viewed as the class for which there is the most evidence. Similarly, the minimum deletes classes with little evidence. In order to avoid a single classifier from having too large of an impact on the eventual output, these two values can be averaged to yield the *spread* combiner. This combiner strikes a balance between the positive and negative evidence, leading to a more robust combiner than either of them.

#### 4.1.1 Spread Combiner for Unbiased Classifiers:

For a classifier without bias, the spread combiner is formally defined as:

$$f_i^{spr}(x) = \frac{1}{2} (f_i^{1:N}(x) + f_i^{N:N}(x)) = p_i(x) + \eta_i^{spr}(x), \quad (20)$$

where:

$$\eta_i^{spr}(x) = \frac{1}{2} (\eta_i^{1:N}(x) + \eta_i^{N:N}(x)).$$

The variance of  $\eta_i^{spr}(x)$  is given by:

$$\sigma_{\eta_i^{spr}}^2 = \frac{1}{4} \sigma_{\eta_i^{1:N}(x)}^2 + \frac{1}{4} \sigma_{\eta_i^{N:N}(x)}^2 + \frac{1}{2} cov(\eta_i^{1:N}(x), \eta_i^{N:N}(x)). \quad (21)$$

where  $cov(\cdot, \cdot)$  represents the covariance between two variables (even when the  $\eta_i$ 's are independent, ordering introduces correlations). Note that because of the ordering, the variances in the first two terms of Equation 21 can be expressed in terms of the individual classifier variances. Furthermore, the covariance between two order statistics can also be determined in tabulated form for given distributions. Table 2 provides these values for a Gaussian distribution based on [27]. This expression can be further simplified for symmetric distributions where  $\sigma_{\eta^{1:N}}^2 = \sigma_{\eta^{N:N}}^2$  (e.g., Gaussian noise model) and leads to:

$$\sigma_{\eta_i^{spr}}^2 = \frac{1}{2} (\alpha_{1:N} + B_{1,N:N}) \sigma_{\eta_i(x)}^2, \quad (22)$$

where  $\alpha_{m:N}$  is the variance of the  $m$ th ordered sample and  $B_{m,l:N}$  is the covariance between the  $m$ th and  $l$ th ordered samples, given that the initial samples had unit variance [27]. Because this is a symmetric distribution, the  $\beta$  values are also symmetric (e.g.,  $\beta_{1,2:5} = \beta_{4,5:5}$ ).

Table 2: Some Reduction Factors  $B$  for the Gaussian Distribution, based on [27].

N	$k, l$	$B$	N	$k, l$	$B$	N	$k, l$	$B$	N	$k, l$	$B$
2	1,2	.318	6	2,3	.189	8	1,4	.095	9	1,6	.059
3	1,2	.276		2,4	.140		1,5	.075		1,7	.049
	1,3	.165		2,5	.106		1,6	.060		1,8	.040
4	1,2	.246	3,4	.183	1,7	.048	1,9	.031			
	1,3	.158	7	1,2	.196	1,8	.037	2,3	.154		
	1,4	.105		1,3	.132	2,3	.163	2,4	.117		
2,3	.236	1,4		.099	2,4	.123	2,5	.093			
5	1,2	.224	1,5	.077	2,5	.098	2,6	.077			
	1,3	.148	1,6	.060	2,6	.079	2,7	.063			
	1,4	.106	1,7	.045	2,7	.063	2,8	.052			
	1,5	.074	2,3	.175	3,4	.152	3,4	.142			
	2,3	.208	2,4	.131	3,5	.121	3,5	.114			
6	2,4	.150	2,5	.102	3,6	.098	3,6	.093			
	1,2	.209	2,6	.080	4,5	.149	3,7	.077			
	1,3	.139	3,4	.166	1,2	.178	4,5	.137			
	1,4	.102	3,5	.130	1,3	.121	4,6	.113			
6	1,5	.077	8	1,2	.186	9	1,4	.091			
	1,6	.056		1,3	.126	1,5	.073				

Then, using Equation 4, the variance of the boundary offset  $b^{spr}$  can be calculated:

$$\begin{aligned} \sigma_{b^{spr}}^2 &= \frac{\sigma_{\eta_i^{spr}}^2 + \sigma_{\eta_j^{spr}}^2}{s^2} \\ &= \frac{1}{2} (\alpha_{1:N} + B_{1,N:N}) \sigma_b^2. \end{aligned} \quad (23)$$

Finally, through Equation 5, we can obtain the reduction in the model error due to the

spread combiner:

$$\frac{E_{model}^{spr}}{E_{model}} = \frac{\alpha_{1:N} + B_{1,N:N}}{2}. \quad (24)$$

Based on Equation 24 and Tables 1 and 2, Table 3 displays the error reductions provided by the spread combiner for a Gaussian noise model (for comparison purposes, the error reduction for the *min* and *max* combiners is also provided. Note that for the Gaussian distribution, the error reduction of *min* is equal to that of *max*).

Table 3: Error Reduction Factors for the Spread, *min* and *max* Combiners with Gaussian Noise Model.

N	<i>spread</i>	<i>min</i> or <i>max</i>
2	.500	.682
3	.362	.560
4	.299	.492
5	.261	.448
6	.236	.416
7	.219	.392
8	.205	.373
9	.194	.357
10	.186	.344

#### 4.1.2 Spread Combiner for Biased Classifiers:

Now, if the classifier biases are non-zero, the spread combiner's output is given by:

$$f_i^{spr}(x) = \frac{1}{2} (f_i^{1:N}(x) + f_i^{N:N}(x)) = p_i(x) + (\eta_i(x) + \beta_i)^{spr}. \quad (25)$$

In that case, the boundary offset is given by:

$$b^{spr} = \frac{(\beta_i + \eta_i(x_b))^{spr} - (\beta_j + \eta_j(x_b))^{spr}}{s}, \quad (26)$$

which after expanding each term and regrouping can be expressed as:

$$b^{spr} = \frac{(\beta_i + \eta_i(x_b))^{1:N} - (\beta_j + \eta_j(x_b))^{1:N}}{2s} + \frac{(\beta_i + \eta_i(x_b))^{N:N} - (\beta_j + \eta_j(x_b))^{N:N}}{2s}. \quad (27)$$

The first moment of  $b^{spr}$  can be obtained by analyzing each term of Equation 27. In fact, the offset introduced by the first and  $n$ th order statistic for classes  $i$  and  $j$  will cancel each other out, leaving only the average bias between the min and max components of the error (as in Equation 15), given by  $\beta^{spr} = \frac{\beta_i^{1:N} - \beta_j^{1:N} + \beta_i^{N:N} - \beta_j^{N:N}}{s}$ .

The variance of  $b^{spr}$  needs to be derived from Equation 27. Proceeding as in Equation 16, the variance of the spread combiner can be expressed as:

$$\sigma_{b^{spr}}^2 = \left( \frac{1}{4}\alpha_{1:N} + \frac{1}{4}\alpha_{N:N} + \frac{1}{2}B_{1,N:N} \right) (\sigma_b^2 + \sigma_\beta^2). \quad (28)$$

For a symmetric distribution (where  $\alpha_{1:N} = \alpha_{N:N}$ ), we obtain the following error:

$$\begin{aligned} E_{model}^{spr}(\beta) &= \frac{s}{2} \mathcal{M}_2 = \frac{s}{2} (\sigma_{b^{spr}}^2 + \mathcal{M}_1^2) \\ &= \frac{s}{2} \left( \left( \frac{1}{2}\alpha_{1:N} + \frac{1}{2}B_{1,N:N} \right) (\sigma_b^2 + \sigma_\beta^2) + (\beta^{spr})^2 \right) \\ &= \frac{1}{2} (\alpha_{1:N} + B_{1,N:N}) E_{model}(\beta) + \\ &\quad \frac{s}{4} (\alpha_{1:N} + B_{1,N:N}) (\sigma_b^2 - (\beta)^2) + \frac{s}{2} (\beta^{spr})^2, \end{aligned} \quad (29)$$

which is very similar to Equation 19, where the value of  $\alpha$  for a single order statistic is now replaced by  $\frac{\alpha_{1:N} + B_{1,N:N}}{2}$ , since the mean of the first and  $n$ th order statistic is used in the posterior estimate.

## 4.2 Trimmed Means

Instead of actively using the extreme values as was the case with the spread combiner, one can base the posterior estimate around the median values. However, instead of selecting one classifier output as was done for  $f^{med}$ , one can use multiple classifiers whose outputs are “typical.” In this scheme, only a certain fraction of all available classifiers are used *for a given* pattern. The main advantage of this method over weighted averaging is that the set of classifiers which contribute to the combiner vary from pattern to pattern. Furthermore, they do not need to be determined externally, but are a function of the current pattern and the classifier responses to that pattern.

### 4.2.1 Trimmed Mean Combiner for Unbiased Classifiers:

Let us formally define the trimmed mean combiner ( $\beta_k = 0, \forall k$ ) as follows:

$$f_i^{trim}(x) = \frac{1}{N_2 - N_1 + 1} \sum_{m=N_1}^{N_2} f_i^{m:N}(x) = p_i(x) + \eta_i^{trim}(x), \quad (30)$$

where:

$$\eta_i^{trim}(x) = \frac{1}{N_2 - N_1 + 1} \sum_{m=N_1}^{N_2} \eta_i^{m:N}(x).$$

The variance of  $\eta_i^{trim}(x)$  is given by:

$$\begin{aligned} \sigma_{\eta_i^{trim}}^2 &= \frac{1}{(N_2 - N_1 + 1)^2} \sum_{l=N_1}^{N_2} \sum_{m=N_1}^{N_2} cov(\eta_i^{m:N}(x), \eta_i^{l:N}(x)) \\ &= \frac{1}{(N_2 - N_1 + 1)^2} \left( \sum_{m=N_1}^{N_2} \sigma_{\eta_i^{m:N}(x)}^2 + \sum_{m=N_1}^{N_2} \sum_{l>m}^{N_2} 2 cov(\eta_i^{m:N}(x), \eta_i^{l:N}(x)) \right). \end{aligned} \quad (31)$$

Again, using the factors in Tables 1 and 2, Equation 31 can be further simplified. Note that because the Gaussian distribution is symmetric, the covariance between the  $k$ th and  $l$ th ordered samples is the same as that between the  $N + 1 - k$ th and  $N + 1 - l$ th ordered samples. Therefore, Equation 31 leads to:

$$\begin{aligned} \sigma_{\eta_i}^{2\text{trim}} &= \frac{1}{(N_2 - N_1 + 1)^2} \sum_{m=N_1}^{N_2} \alpha_{m:N} \sigma_{\eta_i(x)}^2 \\ &+ \frac{2}{(N_2 - N_1 + 1)^2} \sum_{m=N_1}^{N_2} \sum_{l>m} B_{m,l:N} \sigma_{\eta_i(x)}^2, \end{aligned} \quad (32)$$

where  $\alpha_{m:N}$  is the variance of the  $m$ th ordered sample and  $B_{m,l:N}$  is the covariance between the  $m$ th and  $l$ th ordered samples, given that the initial samples had unit variance [27]. Using the theory highlighted in Section 2, and Equation 32, we obtain the following model error reduction:

$$\frac{E_{\text{model}}^{\text{trim}}}{E_{\text{model}}} = \frac{1}{(N_2 - N_1 + 1)^2} \left( \sum_{m=N_1}^{N_2} \alpha_{m:N} + 2 \sum_{m=N_1}^{N_2} \sum_{l>m} B_{m,l:N} \right). \quad (33)$$

Based on Equation 33 and Tables 1 and 2, we have generated a sample *trim* combiner reduction table. Because there are many possibilities for  $N_1$  and  $N_2$ , a table that exhaustively provides all reduction values is not practical. In this sample table we have selected  $N_1 = 2$  and  $N_2 = N - 1$ , that is, averaging after the lowest and highest values have been removed. For comparison purposes the reduction factors of the averaging combiner for  $N$  and  $N - 2$  classifiers are also provided (for i.i.d. classifiers the reduction factors are  $1/N$  as derived in [30, 32]; similar results were obtained for regression problems [24]). As these numbers demonstrate, although  $N - 2$  classifiers are used in the trim combiner, *selectively* weeding out undesirable classifiers provides reduction factors significantly better than simply averaging  $N - 2$  arbitrary classifiers. The *trim* combiner provides reduction factors comparable to the  $N$  classifier *ave* combiner without being susceptible to corruption by one particularly faulty classifier.

Table 4: Error Reduction Factors for Trim and two corresponding *ave* Combiners with Gaussian Noise Model.

N	<i>ave</i> (for N)	<i>trim</i> (for $N_1 = 2$ ; $N_2 = N - 1$ )	<i>ave</i> (for $N - 2$ )
3	.333	.449	1.00
4	.250	.298	.500
5	.200	.227	.333
6	.167	.184	.250
7	.143	.155	.200
8	.125	.134	.167
9	.111	.113	.143

#### 4.2.2 Trimmed mean Combiner for Biased Classifiers:

Now, if the classifier biases are non-zero, the trimmed mean combiner's output is given by:

$$f_i^{trim}(x) = \frac{1}{N_2 - N_1 + 1} \sum_{m=N_1}^{N_2} f_i^{m:N}(x) = p_i(x) + (\eta_i(x) + \beta_i)^{trim}. \quad (34)$$

In that case the boundary offset is given by:

$$b^{trim} = \frac{(\beta_i + \eta_i(x_b))^{trim} - (\beta_j + \eta_j(x_b))^{trim}}{s}. \quad (35)$$

The first moment of  $b^{trim}$  can be obtained from a manner similar to that of the spread combiner. Indeed, each mean offset introduced by a specific order statistic for class  $i$  will be offset by the one introduced for class  $j$ . Only the trimmed mean of the biases will remain, giving the first moment of  $b^{trim}$ :

$$\beta^{trim} = \frac{1}{N_2 - N_1 + 1} \sum_{m=N_1}^{N_2} \frac{\beta_i^{m:N} - \beta_j^{m:N}}{s}. \quad (36)$$

In deriving the variance of  $b^{trim}$ , we follow the same steps as in Sections 3.2 and 4.1.1. The resulting boundary variance is similar to Equation 16, but the since the reduction is due to the linear combination of multiple ordered outputs,  $\alpha$  is replaced by  $\mathcal{A}$ , where:

$$\mathcal{A} = \frac{1}{(N_2 - N_1 + 1)^2} \left( \sum_{m=N_1}^{N_2} \alpha_{m:N} + 2 \sum_{m=N_1}^{N_2} \sum_{l>m} B_{m,l:N} \right). \quad (37)$$

The model error reduction in this case is given by:

$$\begin{aligned} E_{model}^{trim}(\beta) &= \frac{s}{2} \mathcal{M}_2 = \frac{s}{2} (\sigma_{b^{trim}}^2 + \mathcal{M}_1^2) \\ &= \frac{s}{2} (\mathcal{A} (\sigma_b^2 + \sigma_\beta^2) + (\beta^{spr})^2) \\ &= \mathcal{A} E_{model}(\beta) + \frac{s}{2} (\mathcal{A} (\sigma_\beta^2 - (\beta)^2) + (\beta^{spr})^2). \end{aligned} \quad (38)$$

Once again we need to look at the interaction between the two parts of the error reduction. The first term provides the error reduction compared to the model error of an individual classifier. The smaller  $\mathcal{A}$  is, the more error reduction there will be. In the second term, on the other hand, a small value for  $\mathcal{A}$  is only useful if the variability in the individual biases is higher than the biases themselves ( $\sigma_\beta^2 > (\beta)^2$ ).

## 5 Experimental Results

The order statistics-based combining methods proposed in this article are tailored for situations where one or more of the following apply:

1. Individual classifier performance is uneven and class dependent;

2. It is not possible (insufficient data, high amount of noise) to fine tune the individual classifiers without using computationally expensive methods;
3. All the features are not available to all the classifiers.

Such situations occur, for example, in electrical logging while drilling for oil, where data from certain well sites almost completely misses out on portions of the problem space, and in imaging from airborne platforms where the classifiers receive inputs from different satellites and/or different types of sensors (e.g., thermal, optical, SAR). In this article we restrict ourselves to public domain data sets and simulate such variability in two ways, namely, by

- (i) segmenting the feature set and allowing individual classifiers to have access to only a limited portion of the feature set.
- (ii) using “early stopping” i.e., prematurely terminating the training of the individual classifiers<sup>4</sup>.

For the experiments reported below, we used a multi-layer perceptron (MLP) with a single hidden layer, whose weights were randomly initialized for each run. All classification results reported in this article are *test set error rates averaged over 20 runs, along with the differences in the mean (standard deviation divided by square root of the number of runs)*. Several types of simple combiners such as averaging, weighted averaging, voting, products, weighted products (Bayesian), using Dempster-Schafer theory of evidence, and entropy-based averaging, have been proposed in the literature. However, on a wide variety of data sets, it has been observed that simple averaging usually provides results comparable to any of these techniques (and, surprisingly, often better than most of them) [15, 31]. Furthermore, many ensemble techniques (such as subsampling the training set as in bagging) can be performed in conjunction with order statistics just as well as they can be performed with averaging or voting. For this reason, in this study, we use the average combiner as a *representative* of simple combiners, for comparison purposes.

## 5.1 Variability through Segmentation

The first group of experiments focus on classifiers that because of circumstances (e.g., geography) have access to only a part of the full feature set. Although this situation is becoming quite common [21, 33], we are not aware of any public domain data sets for collective data mining. Instead we will create variability in three data sets from the Proben1/UCI benchmarks [4, 25]. Briefly these data sets, and the corresponding size of the MLP used, are<sup>5</sup>:

- Card: a 51-dimensional, 2-class data set based on credit approval decision with 690 patterns; an MLP with 10 hidden units;
- Gene: a 120-dimensional data set with two classes, based on the detection of splice junctions in DNA sequences, with 3175 patterns; an MLP with 10 hidden units;

---

<sup>4</sup>In all the experiments reported here, “early stopping” means that classifiers in an ensemble were trained half as long as they would have been, had they been stand-alone classifiers.

<sup>5</sup>The number of hidden units was determined experimentally.

- Satellite: a 36-dimensional, 6-class data set with 6435 examples of feature vectors extracted from satellite imagery; an MLP with 20 hidden units.

These three sets were chosen as they have *relatively large number of features*, somewhat large number of data points, and have been studied by several researchers. Also note that the Proben1 benchmarks are particular training, validation and test splits of the UCI data sets which are available from URL <http://www.ics.uci.edu/~mllearn/MLRepository.html>. The results presented in this article are based on the first training, validation and test partition discussed in [25], where half the data is used for training, and a quarter each for validation and testing purposes.

We investigate two situations: one where the original features were randomly and disjointly partitioned among the different segments, and the second where there is some overlap among features in different segments. The exact segment count and number of features within each segment is specified in Table 5.

For each data set, we present the original number of features, the number of new features sets that result when the feature set is segmented (for Gene we only have two new sets, because the low dimensionality prevents any further segmentation), and the resulting number of features in each segment with and without overlap among the features.

A classifier trains on data from one segment, and different classifiers operate on different segments. When the number of classifiers in the ensemble was higher than the number of segments ( $N = 8$ ) more than on classifier (starting from a different initialization) was trained on the same features.

Table 5: Number of features in Proben1/UCI data sets

Data	Number of Original Features	Number of Segments	Features per Segment	
			no Overlap	Overlap
Card	51	4	13-13-13-12	18-18-18-18
Gene	120	4	30-30-30-30	40-40-40-40
Sat	36	4	9-9-9-9	15-15-15-15

Tables 6-7 present the results (with the best result for each case in bold font). The misclassification percentage for individual classifiers are reported in the first column. For the trimmed mean combiner, we also provide  $N_1$  and  $N_2$ , the upper and lower cutting points in the ordered average used in Equation 30, obtained through the validation set.

In this case, for two of the three data sets (Gene and Card), there are striking gains due to using order statistics combiners. One cause for these gains is the high variability in performance among the component classifiers. In such cases, a small number of poor classifiers can corrupt the average combiner. By their very nature, though, combiners based on order statistics are immune to this type of corruption. The ave combiner performs well on the Sat data sets where the performance among the individual classifiers is much more homogeneous. In this case, the ave results are only marginally worse than those for the trimmed mean.



Table 6: Segmented Features with overlap (% misclassified  $\pm\sigma/\sqrt{n}$ ).

Data	N	Ave	Max	Min	Spread	Trim ( $N_1-N_2$ )
Card	4	12.21 $\pm$ .00	<b>10.58 <math>\pm</math> .06</b>	10.61 $\pm$ .06	<b>10.58 <math>\pm</math> .00</b>	12.21 $\pm$ .00 (3-4)
30.30 $\pm$ 2.62	8	12.21 $\pm$ .00	<b>10.47 <math>\pm</math> .00</b>	10.61 $\pm$ .06	<b>10.47 <math>\pm</math> .00</b>	<b>10.47 <math>\pm</math> .00</b> (7-8)
Gene	4	18.52 $\pm$ .10	<b>14.02 <math>\pm</math> .13</b>	20.23 $\pm$ .31	14.72 $\pm$ .15	16.86 $\pm$ .15 (3-4)
34.80 $\pm$ 4.01	8	18.06 $\pm$ .06	<b>13.13 <math>\pm</math> .06</b>	17.59 $\pm$ .17	13.69 $\pm$ .11	13.39 $\pm$ .08 (7-8)
Sat	4	14.16 $\pm$ .08	14.73 $\pm$ .18	14.64 $\pm$ .16	14.24 $\pm$ .12	<b>14.00 <math>\pm</math> .07</b> (3-4)
16.40 $\pm$ 0.56	8	14.21 $\pm$ .05	15.27 $\pm$ .15	15.07 $\pm$ .15	14.49 $\pm$ .11	<b>14.01 <math>\pm</math> .04</b> (3-5)

Table 7: Segmented Features without overlap (% misclassified  $\pm\sigma/\sqrt{n}$ ).

Data	N	Ave	Max	Min	Spread	Trim ( $N_1-N_2$ )
Card	4	12.21 $\pm$ .00	<b>10.49 <math>\pm</math> .03</b>	<b>10.49 <math>\pm</math> .03</b>	<b>10.49 <math>\pm</math> .03</b>	12.21 $\pm$ .00 (3-4)
30.90 $\pm$ 2.66	8	12.21 $\pm$ .00	10.78 $\pm$ .07	10.78 $\pm$ .07	<b>10.52 <math>\pm</math> .04</b>	11.05 $\pm$ .00 (7-8)
Gene	4	24.35 $\pm$ .13	15.82 $\pm$ .15	19.15 $\pm$ .22	<b>14.09 <math>\pm</math> .11</b>	23.11 $\pm$ .15 (3-4)
36.87 $\pm$ 3.01	8	23.33 $\pm$ .19	14.99 $\pm$ .14	16.78 $\pm$ .24	<b>13.23 <math>\pm</math> .15</b>	15.03 $\pm$ .17 (7-8)
Sat lap	4	14.39 $\pm$ .09	15.66 $\pm$ .15	15.46 $\pm$ .11	15.11 $\pm$ .11	<b>14.22 <math>\pm</math> .07</b> (2-3)
17.13 $\pm$ 0.47	8	14.37 $\pm$ .05	15.93 $\pm$ .06	15.53 $\pm$ .06	15.18 $\pm$ .10	<b>14.04 <math>\pm</math> .05</b> (3-5)

## 5.2 Variability through Early Stopping

For the second set of experiments we use two classes of acoustic underwater sonar signals<sup>6</sup>. From the original sonar signals of four different underwater objects (porpoise sound, cracking ice and two different whale sounds), two feature sets are extracted [15]:

- WOC: a 25-dimensional feature set, consisting of Gabor wavelet coefficients, temporal descriptors and spectral measurements; and,
- RDO: a 24-dimensional feature set, consisting of reflection coefficients based on both short and long time windows, and temporal descriptors.

For both feature sets, an MLP with 50 hidden units was used. These data sets are available at URL <http://www.lans.ece.utexas.edu>. Further details about this 4-class problem can be found in [15, 31].

Table 8 presents the combining results for the underwater acoustic data set when the individual classifier performance is highly variable. The results of Table 8 as well as those given in [32] indicate that when the individual classifier performance is highly variable, order statistics-based combiners (particularly the *spread* combiner) typically provide better classification results than other simple combiners. This performance improvement is obtained without sacrificing the simplicity of the combiner. On the other hand, no single method based on order statistics was consistently better than the simple combiner. Thus there is no sure bet, but one can in practice benefit from using order statistics based combiners in at least two ways: (i) Since either the *max* or

<sup>6</sup>Detailed results on 6 Proben/UCI data sets were reported in [32] and hence are not repeated here.

Table 8: Combining Results in the Presence of High Variability in Individual Classifier Performance for the Sonar Data (% misclassified  $\pm\sigma/\sqrt{n}$ ).

Data	N	Ave	Max	Min	Spread	Trim ( $N_1-N_2$ )
RDO	4	11.57 $\pm$ .11	11.94 $\pm$ .12	11.52 $\pm$ .20	<b>11.04 <math>\pm</math> .09</b>	11.34 $\pm$ .14 (3-4)
13.32 $\pm$ 0.83	8	11.64 $\pm$ .09	11.47 $\pm$ .11	<b>11.29 <math>\pm</math> .13</b>	11.51 $\pm$ .09	12.30 $\pm$ .08 (4-5)
WOC	4	8.80 $\pm$ .09	<b>7.84 <math>\pm</math> .10</b>	9.31 $\pm$ .12	8.54 $\pm$ .06	8.43 $\pm$ .13 (3-4)
12.07 $\pm$ 1.12	8	8.82 $\pm$ .08	<b>7.68 <math>\pm</math> .12</b>	8.91 $\pm$ .06	8.24 $\pm$ .11	7.81 $\pm$ .08 (7-8)

*min* combiner will usually provide better classification rates than *ave*, but it is difficult to determine which of the two will be more successful for a specific data set, one can try both and use a validation set to select one over the other. (ii) Alternatively one can just use the *spread* combiner since it consistently ranks among the best order statistics results.

## 6 Conclusion

In this article we present and analyze combiners based on order statistics. These combiners blend the simplicity of averaging with the generality of meta-learners. They are particularly effective if there are significant variations among component classifiers in at least some parts of the joint input-output space. Variations can arise when the individual training sets cannot be considered as *random* samples from a common universal data set. Examples of such cases include real-time data acquisition and classification from geographically distributed sources or data mining problems with large databases, where random subsampling is computationally expensive and practical methods lead to non-random subsamples [5]. Furthermore, the robustness of order statistics combiners is also helpful when certain individual classifiers experience catastrophic failures (e.g., due to faulty sensors).

The analytical framework provided in this paper quantifies the reductions in error achieved when an order statistics based ensemble is used. It also suggests that the two methods for linear combination of order statistics introduced in this paper should provide more reliable estimates of the true posteriors than any of the individual order statistic combiners. While interpreting the results one must bear in mind the simplifying assumptions underlying this framework. Perhaps the biggest assumption is that the distribution of error  $\epsilon_i(x)$  in Eq. 1 is i.i.d across all classifiers. Furthermore, we assume that for a given classifier and input,  $\epsilon_i(x)$  is independent across all classes. The latter assumption is clearly not true if one normalizes the outputs to always sum to one, in which case one degree of freedom is lost. One can avoid both these assumptions through a more involved model, but it significantly complicates derivations without producing any additional insight.

The experimental results of Section 5 indicate that when there is high variability among the classifiers, the order statistics-based combiners significantly outperform sim-

ple combiners, whereas in the absence of such variability these combiners perform no worse. Thus the family of order statistic combiners is able to extract an appropriate amount of information from the individual classifier outputs without requiring tuning additional parameters as in meta-learners, and without being substantially affected by outliers.

A source of variability not investigated in this paper is when different and diverse feature sets are used to describe the same set of underlying physical phenomena. Experimental results show that combining can exploit such variability to further improve accuracy [12, 15, 8], and it will be instructive to see how order statistics based combiners fare in such scenarios.

**Acknowledgements:** This research was supported in part by NSF grant ECS 9900353, and by a research grant from Intel Corp.

## 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] B.C. Arnold, N. Balakrishnan, and H.N. Nagaraja. *A First Course in Order Statistics*. Wiley, New York, 1992.
- [3] R. Battiti and A. M. Colla. Democracy in neural nets: Voting schemes for classification. *Neural Networks*, 7(4):691–709, 1994.
- [4] C. Blake, E. Keogh, and C.J. Merz. UCI repository of machine learning databases, 1998. (URL: <http://www.ics.uci.edu/~mllearn/MLRepository.html>).
- [5] P.S. Bradley and U. M. Fayyad. Refining initial points for K--means clustering. In *Proceedings of the International Conference on Machine Learning (ICML-98)*, pages 91–99, July 1998.
- [6] L. Breiman. Combining predictors. In A. J. C. Sharkey, editor, *Combining Artificial Neural Nets*, pages 31–50. Springer-Verlag, 1999.
- [7] P. Chan and S. Stolfo. On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Integration of Information*, 8(1):8–25, 1997.
- [8] K. Chen. A connectionist method for pattern classification with diverse features. *Pattern Recognition Letters*, 19:545–558, 1998.
- [9] B. Dasarathy. *Decision Fusion*. IEEE CS Press, Los Alamitos, CA, 1994.
- [10] H. A. David. *Order Statistics*. Wiley, New York, 1970.
- [11] T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, pages 1–15. LNCS Vol. 1857, Springer, 2000.
- [12] R.P.W. Duin and D.M.J. Tax. Experiments with classifier combining rules. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, pages 16–29. LNCS Vol. 1857, Springer, 2000.

- [13] J. H. Friedman. On Bias, Variance, 0/1 Loss and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–78, 1997.
- [14] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.
- [15] J. Ghosh, L. Deuser, and S. Beck. A neural network based hybrid system for detection, characterization and classification of short-duration oceanic signals. *IEEE Journal of Ocean Engineering*, 17(4):351–363, October 1992.
- [16] J. Ghosh and K. Tumer. Structural adaptation and generalization in supervised feedforward networks. *Journal of Artificial Neural Networks*, 1(4):431–458, 1994.
- [17] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1000, 1990.
- [18] 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.
- [19] 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.
- [20] Robert Jacobs. Method for combining experts' probability assessments. *Neural Computation*, 7(5):867–888, 1995.
- [21] H. Kargupta and P. Chan, editors. *Advances in Distributed and Parallel Knowledge Discovery*. AAAI/MIT Press, Cambridge, MA, 2000.
- [22] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- [23] C.J. Merz and M.J. Pazzani. Combining neural network regression estimates with regularized linear weights. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems-9*, pages 564–570. M.I.T. Press, 1997.
- [24] 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*, chapter 10. Chapman-Hall, 1993.
- [25] Lutz Prechelt. PROBEN1 — A set of benchmarks and benchmarking rules for neural network training algorithms. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, D-76128 Karlsruhe, Germany, September 1994. Anonymous FTP: /pub/papers/techreports/1994/1994-21.ps.Z on ftp.ira.uka.de.
- [26] M.D. Richard and R.P. Lippmann. Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Computation*, 3(4):461–483, 1991.
- [27] 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.

- [28] R. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. 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.
- [29] A. J. C. Sharkey. (editor). *Connection Science: Special Issue on Combining Artificial Neural Networks: Ensemble Approaches*, 8(3 & 4), 1996.
- [30] K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341–348, February 1996.
- [31] K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science, Special Issue on Combining Artificial Neural Networks: Ensemble Approaches*, 8(3 & 4):385–404, 1996.
- [32] K. Tumer and J. Ghosh. Linear and order statistics combiners for pattern classification. In A. J. C. Sharkey, editor, *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, pages 127–162. Springer-Verlag, London, 1999.
- [33] K. Tumer and N. C. Oza. Decimated input ensembles for improved generalization. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-99)*, 1999.
- [34] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.