Classifier Combining through Trimmed Means and Order Statistics

Kagan Tumer, NASA Ames Research Center, Moffett Field, CA
Joydeep Ghosh, Electrical and Computer Engineering, The University of Texas, Austin, TX

Abstract— **Combining the outputs of multiple neural networks has led to substantial improvements in several difficult pattern recognition problems. In this article, we introduce and investigate robust combiners, a family of classifiers based on order statistics. We focus our study to the analysis of the decision boundaries, and how these boundaries are affected by order statistics combiners. In particular, we show that using the *i*th order statistic, or a linear combination of the ordered classifier outputs is quite beneficial in the presence of outliers or uneven classifier performance. Experimental results on several public domain data sets corroborate these findings.**

I. Introduction

In recent years, a great deal of attention has been focused on pooling as a means to improve the generalization ability of neural networks [22]. Approaches to pooling classifiers can be separated into two main categories: simple combiners, e.g., voting [4], [5] or averaging [16], and computationally expensive combiners, e.g., stacking [3], [28]. 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 [1], [9], [28]. 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. Weighted averaging of classifier outputs appears to provide some flexibility [7], [12]. Unfortunately, the optimal weights are determined by the inverse of an estimated "mismatch" covariance matrix, leading to inaccuracies for small training sets. Also, the weights are 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 work 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 II provides the background, by summarizing the relationship between classifier errors and decision boundaries, and describes the order statistics combiners [25]. Based on these concepts, in Section III we derive the errors associated with the linear combination of ordered classifier outputs. Section V discuses the implications of using linear combination of order statistics as a strategy for pooling the outputs of individual classifiers.

II. Background

A. Classification Error

Based on the well-known result that the outputs of certain classifiers, trained to minimize mean square error functions, approximate the *a posteriori* probability densities of the corresponding classes [19], one can model the *i*th output of the *m*th such classifier as:

$$f_i^m(x) = p_i(x) + \eta_i^m(x), \tag{1}$$

where $p_i(x)$ is the true posterior for *i*th class on input x. $\eta_i^m(x)$ is the error of the *m*th classifier in estimating that posterior, and has variance $\sigma^2_{\eta_i(x)}$. Let b denote the offset between the ideal class boundary, x^* (based on $p_i(x) = p_j(x)$) and the realized boundary, x_b (based on $f_i(x) = f_j(x)$). This boundary offset ($b = x_b - x^*$) has variance

$$\sigma_b^2 = \frac{\sigma^2_{\eta_i(x)} + \sigma^2_{\eta_j(x)}}{s^2}, \tag{2}$$

where s is a constant depending on the derivatives of the class posteriors at x [26].

It has been shown in [26] that in the simplest case (ηs are zero mean) the extra error (i.e., error in addition to the Bayes error) due to a single classifier, to a first order approximation, is given by $E_{add} = \frac{s\sigma_b^2}{2}$. Moreover, this error is reduced in an ensemble by the same factor as the reduction in σ_b^2, i.e.,

$$E_{add}^{ensemble} = \gamma E_{add}^{individual}, \tag{3}$$

where $\gamma = \frac{\sigma^2_{b^{ensemble}}}{\sigma_b^2}$. Therefore, the power of an ensemble method is determined by how much it can reduce the boundary variance [26], [27].

B. Order Statistics Combiners

Order Statistics has well known characteristics of robustness and simplicity. To combine multiple classifiers through order

statistics, the network outputs of each of the N classifiers for each class i are first ordered so that

$$f_i^{1:N}(x) \leq f_i^{2:N}(x) \leq \cdots \leq f_i^{N:N}(x)\,. \tag{4}$$

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

When the posterior estimation errors (ηs) of the various classifiers are i.i.d., the relationship between the ordered errors can be ascertained[1]. The reduction in the variance of an order statistic are known for several distributions [2], [21]. In particular, we obtain:

$$\sigma_{b^k}^2 = \alpha_k \sigma_b^2; \tag{5}$$

with α_k given in Table I for the Gaussian distribution based on [21] (note that because the Gaussian distribution is symmetric, the reductions are the same for the ith and $N+1-i$th order statistics). If the kth order statistic is used as the output of the combiner, Equation 5 leads to:

$$E_{model}^{os;k} = \alpha_k\, E_{model}, \tag{6}$$

and can be used as the basis for different types of combiners, such as the trimmed mean or the spread combiners discussed in the following section.

III. Linear Combining of Ordered Outputs

We now propose two combinations of averaging and order statistics for pooling classifier outputs.

A. Trimmed Means

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 who 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. Let us formally define the trimmed mean combiner as follows:

$$\begin{aligned} f_i^{trim}(x) &= \frac{1}{M_2 - M_1 + 1} \sum_{m=M_1}^{M_2} f_i^{m:N}(x) \\ &= p(c_i|x) + \eta_i^{trim}(x)\,, \end{aligned} \tag{7}$$

where:

$$\eta_i^{trim}(x) = \frac{1}{M_2 - M_1 + 1} \sum_{m=M_1}^{M_2} \eta_i^m(x)\,.$$

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

$$\begin{aligned} \sigma^2_{\eta_i^{trim}} &= \frac{1}{(M_2 - M_1 + 1)^2} \sum_{l=M_1}^{M_2} \sum_{m=M_1}^{M_2} cov(\eta_i^{m:N}(x), \eta_i^{l:N}(x)) \\ &= \frac{1}{(M_2 - M_1 + 1)^2} \sum_{m=M_1}^{M_2} \sum_{l>m}^{M_2} 2\, cov(\eta_i^{m:N}(x), \eta_i^{l:N}(x)) \\ &+ \frac{1}{(M_2 - M_1 + 1)^2} \sum_{m=M_1}^{M_2} \sigma^2_{\eta_i^{m:N}(x)} \end{aligned} \tag{8}$$

where $cov(\cdot,\cdot)$ represents the covariance between two variables.

Note that because of the ordering, each variance in the first term of Equation 8 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 II provides these values for a Gaussian distribution based on [21] (note that because the Gaussian distribution is symmetric, the covariance between the kth and lth ordered samples is the same as that between the $N+1-k$th and $N+1-l$th ordered samples). Therefore, Equation 8 leads to:

$$\begin{aligned} \sigma^2_{\eta_i^{trim}} &= \frac{1}{(M_2 - M_1 + 1)^2} \sum_{m=M_1}^{M_2} \alpha_{m:N}\ \sigma^2_{\eta_i(x)} \\ &+ \frac{2}{(M_2 - M_1 + 1)^2} \sum_{m=M_1}^{M_2} \sum_{l>m} \beta_{m,l:N}\ \sigma^2_{\eta_i(x)} \end{aligned} \tag{9}$$

where $\alpha_{m:N}$ is the variance of the mth ordered sample and $\beta_{m,l:N}$ is the covariance between the mth and lth ordered samples, given that the initial samples had unit variance [21]. By using the theory highlighted in Section II-A and Equation 9, we obtain the following model error reduction:

$$\begin{aligned} \frac{E_{model}^{trim}}{E_{model}} &= \frac{1}{(M_2 - M_1 + 1)^2} \\ &\left(\sum_{m=M_1}^{M_2} \alpha_{m:N} + 2 \sum_{m=M_1}^{M_2} \sum_{l>m} \beta_{m,l:N} \right) \end{aligned} \tag{10}$$

B. Spread Combiner

Instead of deleting the extreme values as is the case with the trimmed mean combiner, one can base a decision on those values. The maximum and minimum of a set of classifier outputs carry specific meanings. Indeed the maximum can be viewed as the class with the most evidence for it. Similarly the minimum deletes classes with little evidence. In order to avoid a single classifier from having too large 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 and is formally defined as:

$$\begin{aligned} f_i^{spr}(x) &= \frac{1}{2}\,(f_i^{1:N}(x) + f_i^{N:N}(x)) \\ &= p(c_i|x) + \eta_i^{spr}(x)\,, \end{aligned} \tag{11}$$

where:

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

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

$$\sigma^2_{\eta_i^{spr}} = \frac{1}{4}\sigma^2_{\eta_i^{1:N}(x)} + \frac{1}{4}\,\sigma^2_{\eta_i^{N:N}(x)}$$

[1]The linear combining of classifiers without assuming that the errors are i.i.d. is presented in [27].

TABLE I
REDUCTION FACTORS α FOR THE GAUSSIAN NOISE MODEL [21].

N	k	α	N	k	α	N	k	α
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

TABLE II
REDUCTION FACTORS β FOR THE GAUSSIAN NOISE MODEL [21].

N	k,l	β	N	k,l	β	N	k,l	β	N	k,l	β
2	1,2	.318		2,3	.189		1,4	.095		1,6	.059
3	1,2	.276	6	2,4	.140		1,5	.075		1,7	.049
	1,3	.165		2,5	.106		1,6	.060		1,8	.040
	1,2	.246		3,4	.183		1,7	.048		1,9	.031
4	1,3	.158		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	8	2,4	.123		2,5	.093
	1,2	.224		1,5	.077		2,5	.098		2,6	.077
	1,3	.148		1,6	.060		2,6	.079	9	2,7	.063
5	1,4	.106	7	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
	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
6	1,4	.102		3,5	.130		1,3	.121		4,6	.113
	1,5	.077		1,2	.186	9	1,4	.091			
	1,6	.056	8	1,3	.126		1,5	.073			

$$+ \quad \frac{1}{2} cov(\eta_i^{1:N}(x), \eta_i^{N:N}(x)). \tag{12}$$

This expression can be further simplified for symmetric distributions where $\sigma^2_{\eta^{1:N}} = \sigma^2_{\eta^{N:N}}$ (e.g., Gaussian):

$$\sigma^2_{\eta_i^{spr}} \quad = \quad \frac{1}{2} \left(\alpha_{1:N} + \beta_{1,N:N}\right) \sigma^2_{\eta_{i(x)}}, \tag{13}$$

which leads to:

$$\frac{E^{spr}_{model}}{E_{model}} \; = \; \frac{\alpha_{1:N} + \beta_{1,N:N}}{2} \; . \tag{14}$$

Table III presents the error reductions based on Tables I-II and Equation 14.

IV. RESULTS

The two methods proposed in this paper are expected to do best when:

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 computationally expensive methods –e.g., cross-validation.

In order to simulate such conditions, we report results on combining classifiers where only half the classifiers are fine tuned (i.e., some classifiers are not trained sufficiently). This procedure produces an artificially created variation in the pool of classifiers. In all these experiments we used feed forward neural networks with a single hidden layer[2].

[2]The size of the hidden layer was determined experimentally.

TABLE IV

COMBINING RESULTS IN THE PRESENCE OF HIGH VARIABILITY IN INDIVIDUAL CLASSIFIER PERFORMANCE FOR THE PROBEN1/UCI BENCHMARKS (% MISCLASSIFIED).

Data	N	Ave	Max	Min	Spread	Trim (M_1-M_2)
Cancer	4	$1.38 \pm .28$	$1.38 \pm .28$	$1.38 \pm .28$	$1.38 \pm .28$	$1.32 \pm .27$ (2-3)
$1.49 \pm .82$	8	$1.32 \pm .26$	$1.44 \pm .29$	$1.44 \pm .29$	$1.44 \pm .29$	$1.32 \pm .26$ (2-6)
Card	4	$13.60 \pm .46$	$13.37 \pm .45$	$13.49 \pm .44$	$13.37 \pm .45$	$13.60 \pm .31$ (3-4)
$14.33 \pm .74$	8	$13.66 \pm .39$	$13.08 \pm .29$	$13.02 \pm .29$	$12.97 \pm .26$	$13.20 \pm .37$ (7-8)
Diabetes	4	$25.26 \pm .78$	$25.00 \pm .96$	$25.00 \pm .87$	$25.00 \pm .87$	$25.26 \pm .78$ (3-4)
26.09 ± 2.64	8	$24.84 \pm .74$	$25.05 \pm .68$	$25.05 \pm .68$	$25.05 \pm .68$	$24.84 \pm .62$ (6-8)
Gene	4	$12.90 \pm .47$	$12.90 \pm .55$	$12.94 \pm .52$	$12.66 \pm .43$	$12.67 \pm .46$ (3-4)
15.01 ± 1.62	8	$12.89 \pm .45$	$12.76 \pm .51$	$12.41 \pm .21$	$12.43 \pm .46$	$12.56 \pm .42$ (7-8)
Glass	4	$33.77 \pm .57$	40.19 ± 1.5	$33.21 \pm .92$	$33.21 \pm .92$	$33.77 \pm .57$ (2-3)
42.78 ± 1.57	8	$33.96 \pm .00$	$39.43 \pm .57$	$33.77 \pm .57$	$33.40 \pm .86$	$33.77 \pm .57$ (1-6)
Soybean	4	$7.76 \pm .23$	$7.94 \pm .29$	$12.88 \pm .81$	$7.71 \pm .32$	$7.82 \pm .38$ (3-4)
1.71 ± 3.51	8	$7.65 \pm .00$	$7.82 \pm .27$	13.41 ± 1.1	$7.71 \pm .32$	$7.65 \pm .00$ (4-8)

TABLE V

COMBINING RESULTS IN THE PRESENCE OF HIGH VARIABILITY IN INDIVIDUAL CLASSIFIER PERFORMANCE FOR THE SONAR DATA (% MISCLASSIFIED).

Data	N	Ave	Max	Min	Spread	Trim (M_1-M_2)
RDO	4	$11.57 \pm .46$	$11.94 \pm .53$	$11.52 \pm .84$	$11.04 \pm .40$	$11.34 \pm .58$ (3-4)
13.32 ± 3.46	8	$11.64 \pm .38$	$11.47 \pm .45$	$11.29 \pm .57$	$11.51 \pm .37$	$12.30 \pm .36$ (4-5)
WOC	4	$8.80 \pm .37$	$7.84 \pm .42$	$9.31 \pm .49$	$8.54 \pm .24$	$8.43 \pm .55$ (3-4)
12.07 ± 4.64	8	$8.82 \pm .35$	$7.68 \pm .47$	$8.91 \pm .28$	$8.24 \pm .45$	$7.81 \pm .33$ (7-8)

TABLE III

REDUCTION FACTORS FOR THE SPREAD COMBINER WITH GAUSSIAN NOISE MODEL.

N	$E^{spr}_{model}/E_{model}$
2	.500
3	.362
4	.299
5	.261
6	.236
7	.219
8	.205
9	.194
10	.186

The first 6 data sets (Tables IV , VI) were selected from the Proben1 benchmarks [17][3]. Briefly these data sets are:

- Cancer: a 9-dimensional, 2-class data set based on breast cancer data [11], with 699 patterns, 350 of which are used for training;
- Card: a 51-dimensional, 2-class data set based on credit approval decision [18], with 690 patterns, 345 of which are used for training;
- Diabetes: an 8-dimensional data set with two classes based on personal data from 768 (half used for training) Pima Indians obtained from the National institute of Diabetes and Digestive and Kidney Diseases [23];
- Gene: a 120-dimensional data set with two classes, based on the detection of splice junctions in DNA sequences [14], with 3175 patterns, 1588 of which are used for training;
- Glass: a 9-dimensional, 6-class data set based on the chemical analysis of glass splinters, with 214 patterns half of which are used for training; and
- Soybean: an 82-dimensional, 19-class problem with 683 patterns of which 343 are used for training [13].

The next two data sets (Tables V , VII) are based on underwater sonar signals. Details about this 4-class problem can be found in [6], [27]. Briefly:

- WOC is a 25-dimensional feature set mainly based on Gabor wavelet coefficients; and
- RDO is a 24-dimensional feature set based on autoregressive coefficients.

Tables IV and V present the combining results for the Proben1 benchmarks and the underwater acoustic data sets respectively, when the individual classifier performance was highly variable.

[3]These data sets are also available from the UCI repository at URL http://www.ics.uci.edu/~mlearn/MLRepository.html. The results presented in this article are based on the first training/validation/test partition discussed in [17].

The expected misclassification percentage of individual classifiers is reported in the first column, under the name of the data set[4]. The results of the averaging combiner is also presented in order to provide a basis for comparison. For the trimmed mean combiner, we also provide M_1 and M_2, the upper and lower cutting points in the ordered average used in Equation 7.

These results indicate that when the individual classifier performance is highly variable, order statistics based combiners (particularly the *spread* combiner) provide better classification results than the *ave* combiner on five of the eight data sets. For the other three data sets, no statistically significant differences were detected among the various combiners.

A close inspection of these results reveals that using either the *max* or *min* combiners can provide better classification rates than *ave*, but it is difficult to determine which of the two will be more successful given a data set. A validation set may be used to select one over the other, but in that case, potentially precious training data is used solely for determining which combiner to use. The use of the *spread* combiner removes this dilemma, by consistently providing results that are comparable to or better than the best of the *max-min* duo.

When there is ample data, and all the classifiers are fine tuned and perform well, the average combiner is expected to perform well. However, it is not always possible to determine whether all conditions that lead to such an ideal situation are satisfied. Therefore, it is important to know that the trimmed mean and spread combiners presented in this article do not perform worse than the average combiners under such conditions. To that end we have combined finely tuned feed forward neural networks using the methods proposed in this article and compared the results the traditional averaging method. In this experiment, all the conditions favor the averaging combiner (i.e., all possible difficulties for the average combiner have been removed). The results displayed in Tables VI and VII indicate that even under such circumstances, both the *spread* and *trim* combiners provide results that are comparable to the *ave* combiner and even provide improvements on two data sets.

V. Conclusion

In this article we present and analyze combiners based on order statistics. They are motivated by their ability to blend the simplicity of averaging with the generality of meta-learners. When classifier performance is sample dependent (e.g., significant differences in class to class accuracy) the flexibility of order statistics combiners becomes a great asset. Similarly, when one can expect certain individual classifiers to have catastrophic failures (e.g., based on real time sensors) using order statistics adds a level of robustness that is absent in simple or weighted averaging.

The linear combination of order statistics introduced in this paper provides a more reliable estimate of the true posteriors than any of the individual order statistic combiners. The experimental results suggest that when there is high variability among the classifiers, the order statistics based combiners outperform the traditional averaging combiner, whereas in the absence of such variability these combiners perform no worse than the average combiner. In other words, the family of order statistics combiners extract the "right" amount of information from the classifier outputs without requiring numerous additional parameters.

[4]All results reported in these tables are misclassification percentages on the test set based on 20 runs, followed by the standard deviation.

Acknowledgements

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

References

[1] 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] L. Breiman. Stacked regression. Technical Report 367, Department of Statistics, University of California, Berkeley, 1993.

[4] L. Breiman. Bagging predictors. Technical Report 421, Department of Statistics, University of California, Berkeley, 1994.

[5] P.K. Chan and S.J. Stolfo. A comparative evaluation of voting and meta-learning on partitioned data. In *Proceedings of the Twelfth International Machine Learning Conference*, pages 90–98, Tahoe City, CA, 1995. Morgan Kaufmann.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] R.S. Michalski and R.L. Chilausky. Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, 4(2), 1980.

[14] M. O. Noordewier, G. G. Towell, and J. W. Shavlik. Training knowledge-based neural networks to recognize genes in DNA sequences. In R.P. Lippmann, J.E. Moody, and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems-3*, pages 530–536. Morgan Kaufmann, 1991.

[15] D. W. Opitz and J. W. Shavlik. Generating accurate and diverse members of a neural-network ensemble. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems-8*, pages 535–541. M.I.T. Press, 1996.

[16] M.P. Perrone and L. N. Cooper. Learning from what's been learned: Supervised learning in multi-neural network systems. In *Proceedings of the World Congress on Neural Networks*, pages III:354–357. INNS Press, 1993.

TABLE VI

COMBINING RESULTS WITH FINE-TUNED CLASSIFIERS FOR THE PROBEN1/UCI BENCHMARKS (% MISCLASSIFIED).

Data	N	Ave	Max	Min	Spread	Trim (M_1-M_2)
Cancer	4	$0.69 \pm .23$	$0.69 \pm .23$	$0.69 \pm .23$	$0.69 \pm .23$	$0.69 \pm .23$ (2-3)
$.69 \pm .23$	8	$0.69 \pm .23$	$0.57 \pm .02$	$0.57 \pm .02$	$0.57 \pm .02$	$0.57 \pm .23$ (7-8)
Card	4	$13.14 \pm .47$	$12.91 \pm .23$	$13.02 \pm .47$	$12.91 \pm .23$	$13.14 \pm .47$ (2-3)
$13.87 \pm .76$	8	$13.14 \pm .47$	$12.79 \pm .02$	$12.79 \pm .02$	$12.79 \pm .02$	$12.80 \pm .02$ (7-8)
Diabetes	4	$23.33 \pm .61$	$23.23 \pm .63$	$23.33 \pm .51$	$23.23 \pm .63$	$23.33 \pm .61$ (3-4)
$23.52 \pm .72$	8	$22.92 \pm .47$	$23.23 \pm .71$	$23.12 \pm .71$	$23.23 \pm .71$	$22.92 \pm .47$ (4-8)
Gene	4	$12.41 \pm .43$	$12.46 \pm .49$	$12.51 \pm .37$	$12.41 \pm .36$	$12.41 \pm .26$ (3-4)
$13.49 \pm .44$	8	$12.26 \pm .30$	$12.46 \pm .38$	$12.16 \pm .17$	$12.11 \pm .40$	$12.16 \pm .19$ (1-6)
Glass	4	$32.08 \pm .01$	$32.45 \pm .76$	$32.08 \pm .01$	$32.08 \pm .01$	$32.08 \pm .01$ (3-6)
$32.26 \pm .57$	8	$32.08 \pm .01$	$32.08 \pm .01$	$32.08 \pm .01$	$32.08 \pm .01$	$32.08 \pm .01$ (3-6)
Soybean	4	$7.06 \pm .00$	$7.18 \pm .23$	8.12 ± 1.6	$7.06 \pm .00$	$7.06 \pm .00$ (3-6)
$7.36 \pm .90$	8	$7.06 \pm .00$	$7.18 \pm .23$	9.06 ± 1.7	$7.06 \pm .00$	$7.06 \pm .00$ (3-6)

TABLE VII

COMBINING RESULTS WITH FINE-TUNED CLASSIFIERS FOR THE SONAR DATA (% MISCLASSIFIED).

Data	N	Ave	Max	Min	Spread	Trim (M_1-M_2)
RDO	4	$9.26 \pm .66$	$9.67 \pm .41$	$9.45 \pm .40$	$9.33 \pm .42$	$9.28 \pm .58$ (2-3)
$9.95 \pm .74$	8	$8.94 \pm .13$	$9.62 \pm .34$	$9.36 \pm .32$	$9.48 \pm .37$	$8.92 \pm .21$ (1-6)
WOC	4	$7.05 \pm .26$	$7.31 \pm .31$	$7.44 \pm .36$	$7.31 \pm .33$	$7.05 \pm .33$ (2-3)
$7.47 \pm .44$	8	$7.17 \pm .17$	$7.19 \pm .25$	$7.41 \pm .34$	$7.22 \pm .15$	$7.07 \pm .21$ (2-6)

[17] Lutz Prechelt. PROBEN1 — A set of benchmarks and benchmarking rules for neural network training algorithms. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, Germany, September 1994.

[18] J.R. Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27:221–234, December 1987.

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

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

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

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