Ensemble Learning in Linearly Combined Classifiers Via Negative Correlation

Manuela Zanda[1], Gavin Brown[1], Giorgio Fumera[2], and Fabio Roli[2]

[1] School of Computer Science, University of Manchester, UK
{zandam,gbrown}@cs.man.ac.uk
[2] Dept of Electrical and Electronic Engineering, University of Cagliari, Italy
{fumera,roli}@diee.unica.it

Abstract. We investigate the theoretical links between a regression ensemble and a linearly combined classification ensemble. First, we reformulate the Tumer & Ghosh model for linear combiners in a regression context; we then exploit this new formulation to generalise the concept of the "Ambiguity decomposition", previously defined only for regression tasks, to classification problems. Finally, we propose a new algorithm, based on the Negative Correlation Learning framework, which applies to ensembles of linearly combined classifiers.

1 Introduction

The field of Multiple Classifier Systems (MCSs) has now firmly established itself as able to produce state-of-the-art learning techniques. It enjoys an abundance of heuristic methods for improving performance, though on the whole is lacking in theoretical contributions. As such, one of the most highly cited references in the MCS literature is Tumer & Ghosh [1]; this was the first work to show that *correlations* between classifier outputs[1] had a quantifiable effect on the ensemble error. A parallel field to MCS is that of *regression ensembles*; that is, ensembles of estimators that solve a regression problem. In this field, the theoretical framework is far more established and can claim a heritage as far back as Laplace [2], or further. A central result here is the *bias-variance-covariance* decomposition of the mean squared error (MSE). This illustrated that the performance of the ensemble is critically dependent on the three-way balance between bias, variance, and covariance; the latter accounting for correlations between estimators. This trade-off is the analog of the often cited "diversity" in the MCS literature.

In previous work we proposed a learning algorithm, Negative Correlation (NC) learning [3] which *explicitly manages* the bias-variance-covariance (diversity) trade-off using a penalty term in the error function. In this work we extend this to the classification domain, by clearly relating the Tumer & Ghosh model to the bias-variance-covariance decomposition, and deriving a novel learning method based on NC learning.

[1] It should be noted that the model applies only to ensembles that average class probability estimates—the equivalent work for ensembles using majority voting is an outstanding question in the MCS community.

M. Haindl, J. Kittler, and F. Roli (Eds.): MCS 2007, LNCS 4472, pp. 440–449, 2007.

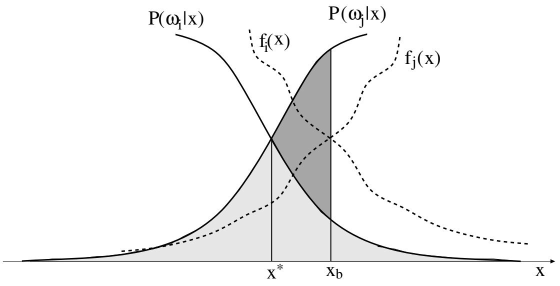


Fig. 1. Estimating posterior probabilities shifts the ideal decision boundary x^* by a quantity $b = x_b - x^* > 0$. Misclassification error is due to irreducible error (light-grey area) and added error (dark-grey area).

2 Background

In this section we describe the background of our research. Firstly we introduce a framework developed by Tumer & Ghosh [1,4] for linearly combined classifiers, and then discuss the equivalent problem in a regression framework.

2.1 Tumer & Ghosh Framework for Linearly Combined Classifiers

It is well known that for a given class k a classifier can only provide an estimate $f_k(\boldsymbol{x})$ of the posterior probability $P(\omega_k|\boldsymbol{x})$. Therefore if we choose the maximum probability class, non-optimal decisions are taken for patterns where $arg\ max_k\ f_k(\boldsymbol{x}) \neq arg\ max_k\ P(\omega_k|\boldsymbol{x})$. In a series of studies [1,4] Tumer & Ghosh analysed the case in which there is a shift of the ideal class boundary. This is shown in Fig. 1 for a two class problem.

According to their framework, the estimated posterior probability for a class ω_i is the sum of the true posterior probability $P(\omega_i|x)$ and an estimation error ϵ_i. Under the simplifying assumptions of

- a shift of the decision boundary x_b around the ideal decision boundary x^* caused by estimation errors
- a first order approximation of the posterior probabilities
- a zero order approximation of the input space distribution x around the ideal decision boundary x^*

they showed that the added error for a single classifier is proportional to the square of the boundary shift b

$$\mathrm{E} = \frac{p(x^*)t}{2} b^2 \tag{1}$$

and that the shift itself can be expressed as a function of the estimation errors $\epsilon_i(x_b)$ and $\epsilon_j(x_b)$:

$$b = \frac{\epsilon_i\left(x_b\right) - \epsilon_j\left(x_b\right)}{t} \quad , \tag{2}$$

where t is the difference between derivatives of posteriors at the optimal boundary: $t = P'(\omega_j|x^*) - P'(\omega_i|x^*)$.[2]

They proved that the expected added error $\mathrm{E}_{\mathrm{add}} = \mathcal{E}\{\mathrm{E}\}$ for a single classifier can be decomposed in terms of the bias and variance of this shift b. The authors then extended this to an expression of the expected added error for a simple average combination of M classifiers, deriving an expression that accounted for the effect of classifier correlations on the added error. As shown in Fig. 1, the added error in (1) is just a portion of the overall misclassification error evaluated around the decision boundary x^*.

2.2 The Regression Context

In a regression context quantifying diversity among component individuals of an ensemble is a well defined problem. Here, the combiner function is a linear combination (as in the Tumer & Ghosh model) and the loss function of interest is not the classification error, but instead the MSE.

In this context, Geman et al. [6] showed that the MSE can be broken into separate components, termed *bias and variance*:

$$\mathcal{E}\left\{(f-d)^2\right\} = \left(\mathcal{E}\left\{f\right\} - d\right)^2 + \mathcal{E}\left\{\left(f - \mathcal{E}\left\{f\right\}\right)^2\right\} \tag{3}$$

where f denotes the estimator, d the target, and the expectation is with respect to all possible training sets. Ueda and Nakano [7] extended this concept for a linearly combined regression ensemble (i.e. where the estimator is $\bar{f} = \frac{1}{M}\sum_{m=1}^{M} f^m$), providing the *bias-variance-covariance* decomposition. Krogh and Vedelsby [8] developed another important decomposition for the MSE, termed the *Ambiguity decomposition.* They proved that at a single data point the MSE can be broken into an accuracy and Ambiguity term:

$$\left(\bar{f} - d\right)^2 = \frac{1}{M}\sum_{m=1}^{M}\left(f^m - d\right)^2 - \frac{1}{M}\sum_{m=1}^{M}\left(f^m - \bar{f}\right)^2 \ . \tag{4}$$

The first term is an index of the accuracy of the individuals, while the second one characterizes diversity among individuals, being a measure of how individual answers differ from the ensemble answer on this single data point.

What is interesting to point out is that Brown et al. [3] showed that the expectation of the Ambiguity decomposition leads strictly to the bias-variance-covariance decomposition, and there exists a common term which quantifies the accuracy-diversity trade-off in this case. The diversity cannot be maximized without affecting the accuracy of the individual components, and the often cited 'diversity dilemma' is in fact a three-way balance between bias, variance, and covariance.

[2] In this paper we follow the notation used by Fumera and Roli in [5].

3 Linking the Regression and Classification Frameworks

The equivalence between the Ambiguity and the bias-variance-covariance decomposition [9] and its exploitation through the NC framework [3] represent a well-grounded theoretical basis for the understanding of MCSs in terms of the accuracy-diversity trade-off between its individual components. The classification context lacks such a neat theory. The main result reached so far is the Tumer & Ghosh model, that shows how correlation among individual classifiers can affect the performance of a MCS. It would be then useful to understand how they relate to each other. In this section we will show that *a regression problem is implicit* in the Tumer & Ghosh model, but it is not obvious what is the estimator and what is the target that are to be considered. Our contribution will be to make it clear.

3.1 Which Random Variable to Consider?

As we already mentioned in Sect. 2.2, in regression contexts we want to minimise the MSE, that is the squared difference between the estimator function f and the true target d. Thanks to well known bias-variance decomposition [6], the expected mean squared error can be decomposed into bias and variance, as illustrated in (3).

In the Tumer & Ghosh model, the random variable (RV) in question is the boundary shift b in Fig. 1. Intuitively, b can be regarded as the 'key' variable to reformulate this in a regression framework. As b decreases towards 0, the added error drops accordingly; though bias and variance of b are discussed, it should be noted that this model differs from other bias-variance decompositions for classification problems, e.g. [10], because it treats the error as a regression random variable.

The connection between the bias-variance-covariance and the Tumer & Ghosh model is not immediately apparent; the main question is: *what are the corresponding 'estimator' and 'target' variables in this framework?*

In order to answer this question, we can first observe that the shaded area in Fig. 1 has approximately the shape of a *triangle*. The area S of a triangle is $S = \frac{1}{2}$ (base × height).

After some manipulations we can rewrite (1) as

$$\mathrm{E} = p\left(x^*\right)\ \frac{1}{2}\ \left(\epsilon_i - \epsilon_j\right)\ \frac{\epsilon_i - \epsilon_j}{t}\quad . \tag{5}$$

If we do not take into consideration the constant $p(x^*)$, it is easy to see that the added error is the area of a triangle having base $(\epsilon_i - \epsilon_j)$ and height $b = \frac{\epsilon_i - \epsilon_j}{t}$.

Let us denote $P_i = P(\omega_i|x)$ and $P_j = P(\omega_j|x)$ the posterior probabilities of classes ω_i and ω_j conditioned on point x. The posterior probability for the k-th class can be written as:

$$f_k = P_k + \epsilon_k\quad . \tag{6}$$

The base $(\epsilon_i - \epsilon_j)$ of the triangle can be expressed as:

$$\epsilon_i - \epsilon_j = (f_i - f_j) - (P_i - P_j)\quad . \tag{7}$$

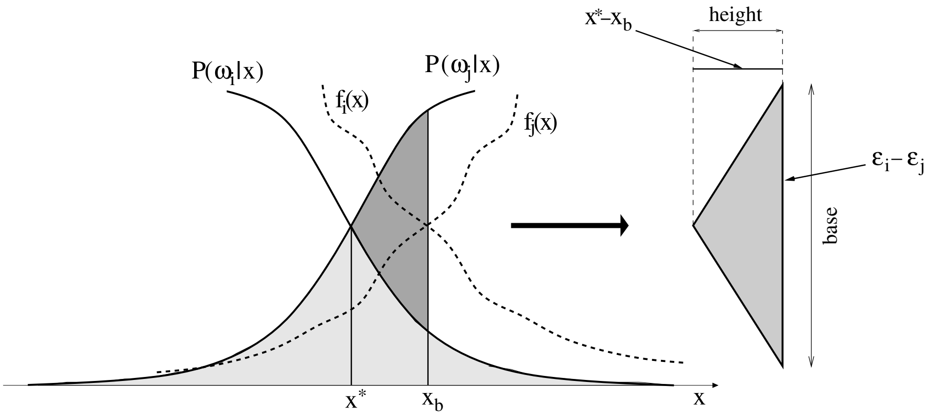


Fig. 2. The added error has approximately the shape of a triangle

If we look at the picture in Fig. 2 the base of the triangle is not only proportional to b (it is t times b) but is also a more meaningful random variable. Indeed the error, that is proportional to b^2 is equal to 0 whenever b is equal to 0. At the optimum boundary, the base of the triangle is equal to 0:

$$(f_i - f_j) - (P_i - P_j) = 0 \ . \tag{8}$$

The error drops to 0 when the difference between the two function estimation equals the difference between the posterior probabilities.

Tumer & Ghosh model can be interpreted as a regression problem by simply considering the base instead of the height of the triangle. In this case we have an estimator $f_{ij} = (f_i - f_j)$, that is the difference between two posterior probability estimators. Furthermore we can think of the difference $d_{ij} = (P_i - P_j)$ as the target of our new regression problems. The aim of the regression problem is to make the function estimator $(f_i - f_j)$ as close as possible to the new target $(P_i - P_j)$. This is true for every point $x \in \mathrm{I\!R}$, as shown in Fig. 2.

This change of random variables increases the understanding of the model, because it makes possible to point out a valid estimator function and target for the Tumer & Ghosh model. Indeed this looking at the Tumer & Ghosh model from another perspective determines to re-define not only the RV of interest, but also its bias-variance decomposition as summarised in Table 1.

Table 1. Some key aspects of the original T & G model are compared with our new interpretation in a regression context

	T & G Model	New Interpretation
RV	$b = \frac{1}{t}\left[(f_i - f_j) - (P_i - P_j)\right]$	$f_i - f_j$
Target	0	$P_i - P_j$
Bias	$\beta_b = \frac{\beta_i - \beta_j}{t}$	$\beta_{ij} = t\beta_b + (P_i - P_j)$
Variance	$\sigma_b^2 = \frac{\sigma_i^2 + \sigma_j^2}{t^2}$	$\sigma_{ij}^2 = t^2\sigma_b^2$

Now that we have found a formulation of the Tumer & Ghosh model in a regression context, it would be interesting to investigate the idea of diversity and to develop an algorithm able to show significative improvements whenever we try to minimise the added error.

4 Optimizing Diversity by NC Learning

A way of exploiting this inter-dependency is through the Negative Correlation algorithm [11]. Removing an assumption made by Liu [11], Brown [9] proved that NC learning can be seen to be exploiting the Ambiguity decomposition. In his formulation [3] NC algorithm uses the Ambiguity decompositon as it tries to minimize a "diversity-encouraging" error function:

$$e^{\text{div}} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\left(f^m - d\right)^2 - \gamma\frac{1}{M}\sum_{m=1}^{M}\frac{1}{2}\left(f^m - \bar{f}\right)^2 \quad . \tag{9}$$

The algorithm works iteratively by performing a single weight update for each neural network in the ensemble, according to (9), proceeding in a pattern-by-pattern updating scheme. The error function in (9) allows to train a simple averaged ensemble of estimators in parallel, in contrast to the alternative of training each network independently, by putting $\gamma = 0$ [3]. In a number of benchmark studies [9,3] it was found that a γ value less than 1 showed significant improvements in both convergence speed and generalization ability. It is easy to notice that, except for linear scaling factors, the last term is equal to the Ambiguity term from (4). Given this, we now show how this algorithm can be extended to work on linearly combined ensembles exploiting the theoretical framework described earlier.

Given an ensemble of M classifiers combined by simple averaging and two classes i and j, let us denote with $\bar{f}_i$ is the ensemble estimator function for class i

$$\bar{f}_i = \frac{1}{M}\sum_{m=1}^{M} f_i^m \quad , \tag{10}$$

and with $\bar{f}_{ij} = \bar{f}_i - \bar{f}_j$

$$\bar{f}_{ij} = \frac{1}{M}\sum_{m=1}^{M}\left(f_i^m - f_j^m\right) \quad . \tag{11}$$

Following Krogh and Vedelsby [8], we define the Ambiguity decomposition for the Tumer & Ghosh model as:

$$\left(\bar{f}_{ij} - d_{ij}\right)^2 = \frac{1}{M}\sum_{m=1}^{M}\left(f_{ij}^m - d_{ij}\right)^2 - \frac{1}{M}\sum_{m=1}^{M}\left(f_{ij}^m - \bar{f}_{ij}\right)^2 \quad . \tag{12}$$

[3] Equation 9 is equal to an independent MSE function for each network when $\gamma = 0$.

The NC framework applied to this gives us:

$$\mathrm{E_{ij}} = \left(\bar{f}_{ij} - d_{ij}\right)^2 = \frac{1}{M}\sum_{m=1}^{M}\left(f_{ij}^m - d_{ij}\right)^2 - \gamma\Big\{\frac{1}{M}\sum_{m=1}^{M}\left(f_{ij}^m - \bar{f}_{ij}\right)^2\Big\} \ , \tag{13}$$

where γ is a scaling factor that allows us to vary the covariance component on E_{ij}. If we adopt a gradient descent procedure on (13), it follows that given two classes i and j the partial derivative for the m-th classifier and the i-th class is

$$\frac{\partial \mathrm{E_{ij}}}{\partial f_i^m} = \frac{2}{M}\left(f_{ij}^m - d_{ij}\right) - \frac{2}{M}\,\gamma\,\left(f_{ij}^m - \bar{f}_{ij}\right) \ . \tag{14}$$

In a real multi-class problem it is unknown which pair of classes will contribute to the added error around any point of the feature space. In this case, we have to take into account every possible pair of classes $i, j \mid j \neq i$ and $i, j = 1 \ldots C$:

$$\mathrm{E_{TOT}} = \sum_{i=1}^{C}\sum_{j>i}\left[\frac{1}{M}\sum_{m=1}^{M}\left(f_{ij}^m - d_{ij}\right)^2\right] - \gamma\sum_{i=1}^{C}\sum_{j>i}\left[\frac{1}{M}\sum_{m=1}^{M}\left(f_{ij}^m - \bar{f}_{ij}\right)^2\right] \ . \tag{15}$$

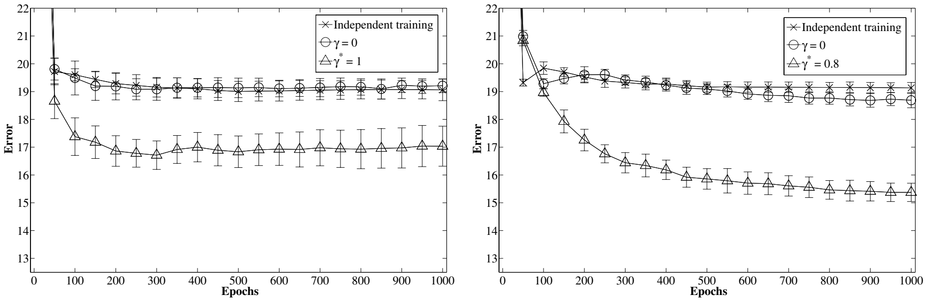
The partial derivative of the overall error with respect to the class i and the estimator function m is

$$\frac{\partial \mathrm{E_{TOT}}}{\partial f_i^m} = \frac{2}{M}\sum_{\substack{j=1\\ j\neq i}}^{C}\left(f_{ij}^m - d_{ij}\right) - \gamma\Big[\frac{2}{M}\sum_{\substack{j=1\\ j\neq i}}^{C}\left(f_{ij}^m - \bar{f}_{ij}\right)\Big] \ . \tag{16}$$

Nevertheless, (14) still holds for each pair of classes, and is the true added error for the two class involved around a decision boundary. Equations (14) and (16) can be used for training in parallel a simple averaged system of neural networks, like (9) does in regression problems as an alternative to the standard independent training with the error function $\frac{1}{2}\sum_{m=1}^{M}\left(f^m - d\right)^2$.

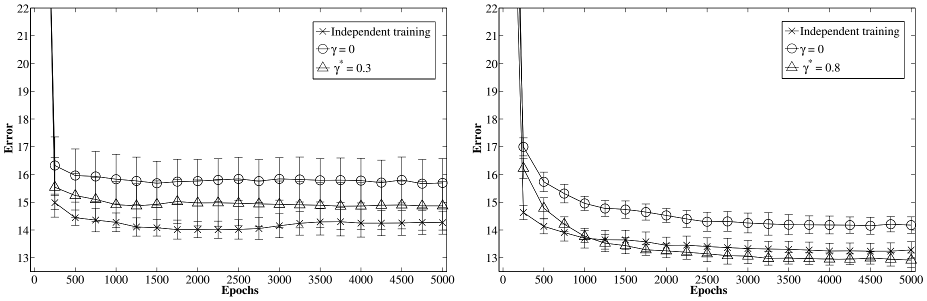
5 Experiments

The aim of these experiments was to assess the performance of the NC ensemble learning algorithm we derived from the new interpretation of Tumer & Ghosh model in a regression context. We have applied this new NC algorithm on three real-world classification problems. The first dataset we used is a random sample of 3602 items from Phoneme dataset, from the ELENA project. The aim of the dataset is phoneme recognition—to distinguish between nasal (class 0) and oral sounds (class 1). There are 3602 data items, 5 continuous features, and the class distribution is approximately 70% class 0 and 30% class 1. The other two datasets were taken from the UCI repository. The Wine dataset has 178 instances, 13 continuous features, and 3 classes; the Heart Disease dataset has 270 instances, 13 features (mixture of continuous/discrete), and 2 classes. In both cases the input features were rescaled to zero mean and unit variance.



$M = 3$ $M = 10$

Fig. 3. Phoneme test error for an ensemble with relatively simple MLPs (each has 3 hidden nodes). On the left is an ensemble of size $M = 3$ (optimum $\gamma^* = 1$). On the right is a larger ensemble of size $M = 10$ (optimum $\gamma^* = 0.8$). The larger ensemble clearly faster convergence.



$M = 3$ $M = 10$

Fig. 4. Phoneme test error for an ensemble with relatively complex MLPs (each has 10 hidden nodes). On the left is an ensemble of size $M = 3$ (optimum $\gamma^* = 0.3$). On the right is a larger ensemble of size $M = 10$ (optimum $\gamma^* = 0.8$). The NC technique shows no significant improvements over independent training with such complex networks.

Experiments have been conducted with multilayer perceptrons, with a single hidden layer, two outputs and logistic activation functions on all nodes. In order to understand the inter-dependency between the number of networks M and the complexity H of networks[4] we have tested four different possible combinations of small/large systems made of low/high complexity neural networks, where we consider 3 and 10 to respectively be a suitable value for small/low and for large/high. Ten runs of the algorithm have been done for each of these combinations. Then, results have been compared with the performance of a single classifier (neural network with two outputs) and with an identical system[5] of individuals trained independently.

[4] i.e. The number of hidden nodes H, considered that every single component of MCS has the same configuration, that is the same number of hidden nodes.

[5] That is same size and same complexity.

Table 2. Mean (and 95% confidence intervals) improvement of systems trained with the NC algorithm over independent training after 1000 epochs for low complexity systems ($H = 3$) and after 5000 epoch for high complexity systems ($H = 10$). Note that the best gains are made with large ensembles of relatively simple networks.

Dataset	M = 3, H = 3	M = 10, H = 3	M = 3, H = 10	M = 10, H = 10
Phoneme	2.0 (0.7)	3.8 (0.4)	−0.6 (1.1)	0.4 (0.3)
Wine	20.5 (2.1)	16.2 (1.4)	1.4 (0.5)	0.9 (0.1)
Heart	1.7 (0.1)	3.4 (0.5)	2.7 (0.4)	3.0 (0.2)

Figure 3 shows results on Phoneme dataset for ensembles of simple networks, while Fig. 4 illustrates results obtained with ensembles of complex networks. In these figures the performance of the independent training MCS, and both performances for the special case $\gamma = 0$ and the optimum γ value γ^*[6] on the test dataset have been reported. Table 2 summarises results—the largest improvement is from a large ensemble of relatively simple networks (3.76%); whereas a small ensemble of complex networks is 0.73% worse than the independent case.

It can also be observed that system improvements can be always obtained for optimum γ values $\gamma^* > 0$. Furthermore, every system has always shown better performances than a single network. Results obtained on Phoneme dataset illustrates that the NC learning algorithm applied in the Tumer & Ghosh framework behaves very similarly to the NC algorithm on regression problems [9]. The observations are consistent with the commonly held idea in the field that MCS benefits are best levied from a large system of relatively simple classifiers. This principle of using a large ensemble of weak classifiers is echoed by other works, such as Boosting or Stochastic Discrimination [12].

6 Discussion and Conclusions

We have run several experiments by testing our NC algorithm on real classification problems. The work done so far, shows that our interpretation is consistent with results obtained, that is the NC learning applied to the new interpretation of the Tumer & Ghosh model shows improvements in terms of performance with reference to a system of networks trained independently. Its success supports the original Tumer & Ghosh idea of decreasing correlations among classifiers as a tool for increasing MCS accuracy, also illustrating that this "diversity" can be *engineered* by an appropriate technique, in this case, the Negative Correlation Learning framework.

An important point to note in this discussion is the assumptions of noise on the target data. If we wish to maximise the log-likelihood of the data, under the assumption of Gaussian noise, the appropriate error function is the mean squared error. For classification problems it is usual to assume binomial/multinomial noise, leading to the cross-entropy error function. It should be noted here that in adopting the regression framework we have implicitly made the assumption

[6] the γ that gives the best performance of the ensemble.

of *Gaussian* distributed noise on the posterior probability estimates. We leave the analysis under different noise assumptions for future work.

A full empirical investigation is out of the scope of this paper and will be conducted in later work. The main contribution of this paper has been to investigate the theoretical links between two different frameworks, that is: the well known regression ensemble and a linearly combined classifier ensemble.

References

1. Tumer, K., Ghosh, J.: Analysis of decision boundaries in linearly combined neural classifiers. Pattern Recognition **29** (1996) 341–348
2. de Laplace, P.S.: Deuxieme supplement a la theorie analytique des probabilites. Paris, Gauthier-Villars (1818) Reprinted (1847) in Oeuvres Completes de Laplace, vol. 7.
3. Brown, G., Wyatt, J., Tino, P.: Managing diversity in regression ensembles. Journal of Machine Learning Research **6** (2005) 1621–1650
4. Tumer, K., Ghosh, J.: Linear and order statistics combiners for pattern classification. In Sharkey, A.J.C., ed.: Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. Springer-Verlag, London (1999) 127–162
5. Fumera, G., Roli, F.: A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 942–956
6. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural Computation **4** (1992) 1–58
7. Ueda, N., Nakano, R.: Generalization error of ensemble estimators. In: Proceedings of International Conference on Neural Networks. (1996) 90–95
8. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. NIPS **7** (1995) 231–238
9. Brown, G.: Diversity in Neural Network Ensembles. PhD thesis, School of Computer Science, University of Birmingham (2004)
10. Kohavi, R., Wolpert, D.H.: Bias plus variance decomposition for zero-one loss functions. In Saitta, L., ed.: Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kaufmann (1996) 275–283
11. Liu, Y., Yao, X.: Ensemble learning via negative correlation. Neural Networks **12** (1999) 1399–1404
12. Kleinberg, E.M.: Stochastic discrimination. Annals of Mathematics and Artificial Intelligence **1** (1990)