Diversity Creation Methods: A Survey and Categorisation

Gavin Brown, Jeremy Wyatt, Rachel Harris, Xin Yao
University of Birmingham, School of Computer Science
Edgbaston Park Road, Birmingham, B15 2TT, UK
www.cs.bham.ac.uk/~gxb/
g.brown@cs.bham.ac.uk

Abstract

Ensemble approaches to classification and regression have attracted a great deal of interest in recent years. These methods can be shown both theoretically and empirically to outperform single predictors on a wide range of tasks. One of the elements required for accurate prediction when using an ensemble is recognised to be error "diversity". However, the exact meaning of this concept is not clear from the literature, particularly for classification tasks. In this paper we first review the varied attempts to provide a formal explanation of error diversity, including several heuristic and qualitative explanations in the literature. For completeness of discussion we include not only the classification literature but also some excerpts of the rather more mature regression literature, which we believe can still provide some insights. We proceed to survey the various techniques used for creating diverse ensembles, and categorise them, forming a preliminary taxonomy of diversity creation methods. As part of this taxonomy we introduce the idea of implicit and explicit diversity creation methods, and three dimensions along which these may be applied. Finally we propose some new directions that may prove fruitful in understanding classification error diversity.

1 Introduction

Creating diverse sets of classifiers is one of the keys to success in multiple classifier systems. Yet this is not easy, and our understanding of classifier error diversity is still incomplete. While we have many measures of diversity from the numerical taxonomy literature [1], we do not yet have a complete grounded framework; neither do we have a useful guide through the myriad of techniques by which we could create such error diversity.

While ensemble approaches to classification usually make use of non-linear combination methods like majority voting; regression problems are naturally tackled by linearly weighted ensembles. These type of ensembles have a much clearer framework for explaining the role of diversity than voting methods. In particular the Ambiguity decomposition [2] and bias-variance-covariance decomposition [3] provide a solid quantification of diversity for linearly weighted ensembles by connecting it back to an objective error criterion: mean squared error. We believe that this can yet provide some insights into diversity in classification problems.

The significant contribution of the paper is a survey and categorisation of the many ad-hoc methods for diversity creation in a classification context. We fully acknowledge that to embark

upon such a venture welcomes debate, and can never result in a flawless categorisation of the field. With this in mind we note that a useful taxonomy should achieve two things: firstly to allow us to spot gaps in the literature, allowing us to explore new techniques, and secondly to allow us to measure the 'distance' between existing techniques; this would afford us some form of metric which could group techniques into families by how well they can perform in different application domains. We believe our study achieves the former of these two, allowing us to identify locations in the space of possible ensemble techniques that have not yet been exploited. We suggest studies akin to Friedman's ISLE framework on importance sampling ensembles [4] will give insights into the latter challenge.

In the next section we provide explanations of why ensembles can work well. This leads in Section 2.1 to a description of the existing formal accounts of diversity in the case of regression using a linearly weighted ensemble. As part of this we explicitly show the link between the bias-variance-covariance and Ambiguity decompositions. Attempts to characterise diversity for classification problems are dealt with in Section 2.3. In Section 3 we describe our proposed taxonomy of diversity creation methods. Discussions of some important issues in ensemble learning are given in Section 4. Finally, conclusions and future research directions are presented in Section 5.

2 When is an Ensemble Better than a Single Learner?

In this section we scrutinise the concept of error "diversity". We review existing explanations of why ensembles with diverse errors perform well; for completeness of the discussion we include the literature for both the regression and classification case. In the process of this we clarify a subtle point, often overlooked, to do with quantifying classifier ensemble diversity and the inherent non-ordinality of the predictor outputs. We proceed with a review of the literature on attempts to create diversity in both forms. We comment on the structure of the field and in the following section propose a novel way to categorise the many ad-hoc diversity creation techniques.

2.1 In a Regression Context

The first major study on combining regression estimators was by Perrone [5] (in fact at the same time, and independently, Hashem [6] developed many of the same results). This was the first study in the Machine Learning literature, but the topic has been covered in other research communities for several years, for example in financial forecasting: Bates and Granger [7, 8], and Clemen [9]. As a consequence, the understanding of diversity here is quite mature, as we will now show.

First, as an illustrative scenario, consider a single neural network approximating a sine wave; our network has been supplied with a limited set of data points to train on, the inputs chosen randomly at uniform from $[-\pi, \pi]$, and a small amount of Gaussian noise added to the outputs. Now, consider a single testing data point, to find the value of $sin(2)$. The true answer is ~ 0.909, yet we know our network may possibly overpredict or underpredict that value. The way in which it makes errors will follow a distribution dependent on the random training data sample it received, and also on the random initialisation of the weights. The mean of this distribution is the expectation value $E_{TW}\{f\}$, and f is a network trained with a particular dataset drawn according to a random variable T and a particular weight initialisation drawn according to a random variable W. Throughout the paper the expectation operator $E\{\cdot\}$ is with respect to these two random variables. It should also be noted that for convenience we have omitted the input vector, so $E\{f\}$ would normally be $E\{f(\mathbf{x})\}$.

In addition all observations we present are with respect to a single pattern, but the results easily generalise to the full space by integrating over the joint input density $P(\mathbf{x}, d)$. Figure 1 illustrates a typical error distribution, with the target value d shown. The four crosses marked are estimates of $sin(2)$ from a hypothetical network; the estimates differ only in that the network was started from different initial weights each time.

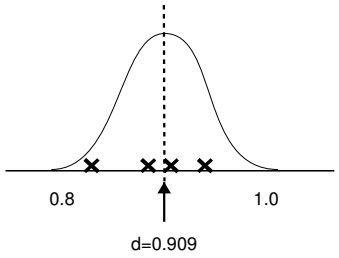

Figure 1: Typical error distribution of an unbiased estimator approximating $sin(2)$

From this viewpoint we can immediately see the parallels to an ensemble system. Each member of an ensemble is a realisation of the random variable defined by this distribution over all possible training datasets and weight initialisations. The ensemble output is the average of this set of realisations; all our diversity promoting mechanisms are there to encourage our sample mean $\bar{f}$ to be a closer approximation to $E_{T,W}\{f\}$. If we have a large ensemble, we have a large sample from this space; consequently with a large sample we can expect that we have a good approximation to the mean, $E_{T,W}\{f\}$. If we have a smaller ensemble, we cannot expect this: our sample mean may be upward or downward biased. In order to correct this, some methods, such as Bagging [10], construct our networks from different training datasets, allowing us to sample a more representative portion of the space. This illustration assumes that the expected value of our estimator is equal to the true target value, i.e. an *unbiased* estimator. If this is not the case, we may have the situation in figure 2.

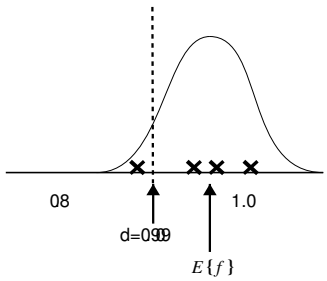

Figure 2: Typical error distribution of a biased estimator approximating $sin(2)$

Here, our estimator is upward biased, i.e. $E\{f\} \neq d$, its expected value $E\{f\}$ is high of the target d. In this case, even if we sample many times from this distribution, we will not be able to estimate the target accurately with a simple average combination as the simple average will converge to $E\{f\}$ as we add more networks. We would need to non-uniformly weight the outputs, giving higher weights to the networks predicting lower values. This is of course a purely hypothetical

scenario, we could not look this closely at every single data point to manually set the weights for the combination, but it does serve to illustrate that the chance of an error could be reduced by using a combination of several predictors rather than one. This intuition can be more formalised as the *Ambiguity Decomposition*.

2.1.1 The Ambiguity Decomposition

Krogh and Vedelsby [2] proved that *at a single data point the quadratic error of the ensemble estimator is guaranteed to be less than or equal to the average quadratic error of the component estimators*:

$$(f_{ens} - d)^2 = \sum_i w_i (f_i - d)^2 - \sum_i w_i (f_i - f_{ens})^2, \tag{1}$$

where f_{ens} is a *convex* combination ($\sum_i w_i = 1$) of the component estimators:

$$f_{ens} = \sum_i w_i f_i \tag{2}$$

The details of the original proof from [2] were omitted for the authors' space considerations. However, this can in fact be shown more simply by the same manipulations as used in the bias-variance decomposition [11], reflecting a strong relationship between the two decompositions. We present this alternative version here:

$$\begin{aligned}
\sum_i w_i (f_i - d)^2 &= \sum_i w_i (f_i - f_{ens} + f_{ens} - d)^2 \\
&= \sum_i w_i \big[(f_i - f_{ens})^2 + (f_{ens} - d)^2 + 2(f_i - f_{ens})(f_{ens} - d) \big] \\
&= \sum_i w_i (f_i - f_{ens})^2 + (f_{ens} - d)^2 \\
(f_{ens} - d)^2 &= \sum_i w_i (f_i - d)^2 - \sum_i w_i (f_i - f_{ens})^2
\end{aligned} \tag{3}$$

This was a very encouraging result for ensemble research, providing a very simple expression for the effect due to error correlation in an ensemble. In section 4 we will see how a number of researchers have exploited this result in various ways to influence the design of their ensemble learners. The significance of the Ambiguity decomposition is that it shows us that if we have any given set of predictors, the error of the convex-combined ensemble will be less than or equal to the average error of the individuals. Of course, one of the individuals may in fact have lower error than the average, and lower than even the ensemble, on a particular pattern. But, given that we have no criterion for identifying that best individual, all we could do is pick one at random. One way of looking at the significance of the Ambiguity decomposition is that it tells us that taking the combination of several predictors would be better *on average over several patterns*, than a method which selected one of the predictors at random.

The decomposition is made up of two terms. The first, $\sum_i w_i (f_i - d)^2$, is the weighted average error of the individuals. The second, $\sum_i w_i (f_i - f_{ens})^2$ is the *Ambiguity term*, measuring the amount of variability among the ensemble member answers for this pattern. Since this is always positive, it is subtractive from the first term, meaning the ensemble is guaranteed lower error than the average

individual error. The larger the Ambiguity term, the larger the ensemble error reduction. However, as the variability of the individuals rises, so does the value of the first term. This therefore shows that diversity itself is not enough, we need to get the right balance between diversity (the Ambiguity term) and individual accuracy (the average error term), in order to achieve lowest overall ensemble error.

The Ambiguity decomposition holds for convex combinations, and is a property of an ensemble trained on a single dataset. Unlike the bias-variance decomposition, it does not take into account the distribution over possible training sets or possible weight initialisations. What we are interested in, of course, is the expected error on future data points given these distributions. The Bias-Variance-Covariance decomposition [3] takes exactly this into account.

2.2 Bias, Variance and Covariance

The concept of ensemble diversity for regression estimators can be understood further if we examine the bias-variance decomposition [11] for an ensemble. The bias-variance decomposition for quadratic loss states that the generalisation error of an estimator can be broken down into two components: bias and variance. These two usually work in opposition to one other: attempts to reduce the bias component will cause an increase in variance, and vice versa. The decomposition [11] is as follows:

$$\begin{aligned} E\{(f - \langle d \rangle)^2\} &= E\{(f - E\{f\})^2\} + (E\{f\} - \langle d \rangle)^2 \\ MSE(f) &= var(f) + bias(f)^2 \end{aligned}$$

Where $\langle d \rangle$ is the expected value of the target point given the noise. If the estimator here is a convex combined ensemble, the variance component breaks down further, and we have the *Bias-Variance-Covariance* decomposition [3]. To clearly understand this, we define three concepts. The first is $\overline{bias}$, the averaged bias of the ensemble members:

$$\overline{bias} = \frac{1}{M} \sum_i (E\{f_i\} - \langle d \rangle) \tag{4}$$

The second is $\overline{var}$, the averaged variance of the ensemble members:

$$\overline{var} = \frac{1}{M} \sum_i E\{(f_i - E\{f_i\})^2\} \tag{5}$$

The third is $\overline{covar}$, the averaged covariance of the ensemble members:

$$\overline{covar} = \frac{1}{M(M-1)} \sum_i \sum_{j \neq i} E\{(f_i - E\{f_i\})(f_j - E\{f_j\})\} \tag{6}$$

This gives us the bias-variance-covariance decomposition of mean-square error:

$$E\{[\Big(\frac{1}{M} \sum_i f_i\Big) - \langle d \rangle]^2\} = \overline{bias}^2 + \frac{1}{M} \overline{var} + \left(1 - \frac{1}{M}\right) \overline{covar} \tag{7}$$

We can see the mean square error of an ensemble of networks depends *critically* on the amount of error correlation between networks, quantified in the covariance term. We would ideally like to decrease the covariance, without causing any increases in the bias or variance terms. It is worth noting that while bias and variance are constrained to be positive-valued, the covariance term can be negative.

2.2.1 The Connection Between Ambiguity and Covariance

We now show the exact link between the two decompositions we have just described. We know that $(\bar{f} - d)^2$ can be expressed in terms of the Ambiguity decomposition. We also know that this is a property of using a quadratic loss function with a convex combination of predictors, and not specific to any particular target value. Therefore we could use the expected value of the target data, $\langle d \rangle$; in this case we re-express $(\bar{f} - \langle d \rangle)^2$ in terms of the average quadratic error and the Ambiguity term. Using $\langle d \rangle$ instead of d, we substitute the right hand side of equation (1) into the left hand side of equation (7).

$$E\{\frac{1}{M}\sum_i (f_i - \langle d \rangle)^2 - \frac{1}{M}\sum_i (f_i - \bar{f})^2\} \quad = \quad \overline{bias}^2 + \frac{1}{M}\overline{var} + \left(1 - \frac{1}{M}\right)\overline{covar} \tag{8}$$

It would be interesting to understand what portions of the bias-variance-covariance decomposition correspond to the Ambiguity term and which portions to the 'average individual error' term. After some manipulations (for details see [12]) we can show:

$$\begin{aligned} E\{\frac{1}{M}\sum_i (f_i - \langle d \rangle)^2\} &= (E\{\bar{f}\} - \langle d \rangle)^2 + \frac{1}{M}\sum_i E\Big\{(f_i - E\{\bar{f}\})^2\Big\} \\ &= bias(\bar{f})^2 + \Omega \end{aligned} \tag{9}$$

$$\begin{aligned} E\{\frac{1}{M}\sum_i (f_i - \bar{f})^2\} &= \frac{1}{M}\sum_i E\{(f_i - E\{\bar{f}\})^2\} - E\{(\bar{f} - E\{\bar{f}\})^2\} \\ &= \Omega - var(\bar{f}) \\ &= \Omega - \left[\frac{1}{M}\overline{var} + \left(1 - \frac{1}{M}\right)\overline{covar}\right] \end{aligned} \tag{10}$$

where the Ω term is an interaction between the two sides:

$$\Omega = \frac{1}{M}\sum_i E\{(f_i - E\{\bar{f}\})^2\} \tag{11}$$

This Ω term is present in both sides—when we combine them by subtracting the Ambiguity in equation (10), from the average MSE in equation (9), the interaction terms cancel out, and we get the original bias-variance-covariance decomposition back, as in the RHS of equation (8). But what does the Ω term *mean*? If we examine it a little further:

$$\frac{1}{M}\sum_i E\{(f_i - E\{\bar{f}\})^2\} \quad = \quad \frac{1}{M}\sum_i E\{(f_i - E_i\{f_i\} + E_i\{f_i\} - E\{\bar{f}\})^2\}$$

$$= \frac{1}{M}\sum_i E_i\{(f_i - E_i\{f_i\})^2\} + \frac{1}{M}\sum_i (E_i\{f_i\} - E\{\bar{f}\})^2$$
$$= \overline{var} + \text{“}deviations''$$

This is the average variance of the estimators, plus the average squared deviation of the expectations of the individuals from the expectation of the ensemble. The fact that the interaction exists illustrates why we cannot simply maximise Ambiguity without affecting the other parts of the error—in effect, this interaction *quantifies* the diversity trade-off for regression ensembles.

In this section we have tried to give an intuition of why diversity among ensemble members that are averaged together can be a good idea. It should be noted that the framework around these two decompositions has told us merely how to *quantify* diversity, not how to *achieve it* in these types of ensemble. In addition we have said nothing yet about ensemble schemes involving other types of combination, such as voting or other non-linear schemes. We will address this in the next section.

2.3 In a Classification Context

We have shown that in a regression context, we can rigorously define why and how differences between individual predictor outputs contribute toward overall ensemble accuracy. In a classification context, there is no such neat theory. There is a subtle point here, often overlooked. The difficulty in quantifying classification error diversity *is not intrinsic* to ensembles tackling classification problems. It is possible to reformulate any classification problem as a regression one by choosing to approximate the class posterior probabilities; this allows the theory we have already discussed to apply, and work is progressing in this area, notably Tumer and Ghosh [13] and Roli and Fumera [14, 15]. For the regression context discussed in the previous section, the question can be clearly phrased as *“how can we quantify diversity when our predictors output real-valued numbers and are combined by a convex combination?”*. For the case that Tumer and Ghosh [13] study, the question is the same, just that the “real-valued” numbers are probabilities. A much harder question appears when we are restricted such that our predictors *can only output discrete class labels*, as we have with Decision Trees or k-nearest neighbour classifiers. In this case, the outputs have *no intrinsic ordinality* between them, and so the concept of “covariance” is undefined. This non-ordinality also implies that we have to change our combination function—a popular one is *majority voting* between the individual votes. The harder question can therefore be phrased as, *“how can we quantify diversity when our predictors output non-ordinal values and are combined by a majority vote?”*.

Taking all these into account, there is simply no clear analogue of the bias-variance-covariance decomposition when we have a zero-one loss function. We instead have a number of highly restricted theoretical results, each with their own assumptions that are probably too strong to hold in practice. We first describe the very well-known work by Tumer and Ghosh, on combining posterior probability estimates (ordinal values), and then turn to considering the harder question of non-ordinal outputs.

2.3.1 Ordinal Outputs

Tumer and Ghosh [16, 13] provided a theoretical framework for analysing the simple averaging combination rule when our predictor outputs are estimates of the posterior probabilities of each class, as in figure 3.

For a one dimensional feature vector x, the solid curves show the true posterior probabilities of classes a and b, these are $P(a)$ and $P(b)$, respectively. The dotted curves show *estimates* of the

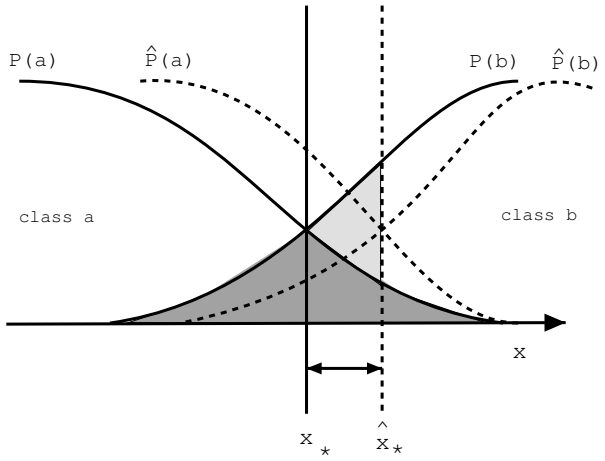

Figure 3: Tumer & Ghosh's framework [16, 13] for analysing classifier error

posterior probabilities, from one of our predictors, these are $\hat{P}(a)$ and $\hat{P}(b)$. The solid vertical line at x_* indicates the *optimal decision boundary*, that is the boundary that will minimise error given that our posterior probabilities overlap. This overlap, the dark shaded area, is termed the *Bayes error*, and is an irreducible quantity. The dotted vertical line at $\hat{x}_*$ indicates the boundary placed by our predictor, which is a certain distance from the optimal. The light shaded area indicates the *added error* that our predictor makes in addition to the Bayes error. The individual predictor i approximates the posterior probability of class a as:

$$\hat{P}_i(a|x) = P(a|x) + \eta_i(a|x) \tag{12}$$

where $P(a|x)$ is the true posterior probability of class a and $\eta_i(a|x)$ is the estimation error. Let us assume the estimation errors on different classes a and b are independent and identically distributed random variables [16, p5] with zero mean and variance $\sigma^2_{\eta_i}$. Consider the predictor's *expected added error* in distinguishing classes a and b, i.e. the *expected* size of the light shaded area given the variance $\sigma^2_{\eta_i}$. This can be stated as:

$$E_{add,i} = \frac{2\sigma^2_{\eta_i}}{P'(a|x) - P'(b|x)} \tag{13}$$

where $P'(a|x)$ is the derivative of the true posterior probability of class a. If the decision boundary was instead placed by an ensemble of predictors, Tumer and Ghosh show the expected added error of the ensemble estimator is:

$$E^{ens}_{add} = E_{add}\Big(\frac{1 + \delta(M-1)}{M}\Big) \tag{14}$$

where M is the number of classifiers. E_{add} is the expected added error of the individual classifiers: they are assumed to have the same error. The δ is a correlation coefficient (see [17] for details) measuring the correlation between errors in approximating the posterior probabilities, therefore this

is a direct measure of diversity[1]. If δ is zero, i.e. the classifiers in the ensemble are statistically independent in their posterior probability estimates. Then we have $E_{add}^{ens} = \frac{1}{M} E_{add}$, the error of the ensemble will be M times smaller than the error of the individuals. If δ is 1, i.e. perfect correlation, then the error of the ensemble will just be equal to the average error of the individuals.

To achieve this simple expression they assume that the errors of the different classifiers have the same variance $\sigma_{\eta_i}^2$. Another, possibly less critical assumption is that the posterior probabilities are monotonic around the decision boundary. This work has recently been extended by Roli and Fumera [14, 15], allowing for some of the assumptions to be lifted, the most important of which is that it allows for non-uniformly weighted combinations. This demonstrates that the understanding of this particular diversity formulation (when outputting posterior probabilities) *is progressing*. What seems to be a sticking point for ensemble research is the non-ordinal output case, as we will now illustrate.

2.3.2 Non-Ordinal Outputs

In ensemble research, Hansen and Salamon [18] is seen by many as the seminal work on diversity in neural network classification ensembles. They stated that a necessary and sufficient condition for a majority voting[2] ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are *accurate* and *diverse*. An accurate classifier is one that has an error rate of better than random guessing, on new input. Two classifiers are diverse if they make different errors on new data points. They used the binomial theorem to explain this; we assume that all networks arrive at the correct classification for a given pattern with probability $(1-p)$, and that they make statistically independent errors. This gives the following probability of a majority voted ensemble being in error:

$$P(\textit{ensemble incorrect}) = \sum_{k>(M/2)}^{M} \binom{M}{k} p^k (1-p)^{(M-k)} \tag{15}$$

This expressed the probability of ensemble error at a single testing data point. For example, with an ensemble of 21 predictors, combined with a majority vote, there would have to be at least 11 members wrong in order for the whole ensemble to be wrong. The area under the binomial distribution for 11 or more, and therefore the probability of the ensemble being incorrect, is 0.026. It should be stressed though, this *only* applies when the predictors are *statistically independent*, an assumption that is too strong to ever hold in practice, but it does represent the theoretical (if unachievable) ideal.

Breiman [19] presents an upper bound on the error of Random Forests—ensembles of decision trees constructed via a particular randomization technique. This is a bound on the generalisation error, so integrated over all possible training sets:

$$P(\textit{ensemble generalization error}) \leq \bar{\rho}(1-s^2)s^2 \tag{16}$$

where s is the 'strength' of the ensemble, expressed as the expected size of the margin by which the ensemble achieves the correct labelling, and $\bar{\rho}$ is the averaged pairwise correlation between Oracle

[1] In fact Tumer and Ghosh rely fundamentally on many of the same manipulations as used in the bias-variance-covariance decomposition—it would be very interesting to see this link made explicit.

[2] They also present results for plurality voting, which turns out to be far more complex.

outputs. Although the bound is not very tight, Kuncheva comments that it can be regarded as a piece of *"that yet missing more general theory of diversity"* [20]. We will comment further on the nature of this problem in section 4.

From this point onwards, when referring to "classification error diversity", it can be assumed that we are referring to this difficult task of quantifying non-ordinal output diversity. From the description of the problem as we have clearly stated, it seems deriving an expression for classification diversity is not as easy as it is for regression. The ideal situation would be a parallel to the regression case, where the squared error of the ensemble can be re-expressed in terms of the squared errors of the individuals and a term that quantifies their correlation. We would like to have an expression that similarly decomposes the classification error rate into the error rates of the individuals and a term that quantifies their 'diversity'. At present, this is beyond the state of the art, however a number of empirical investigations have gone into deriving heuristic expressions that may approximate this unknown diversity term.

2.3.3 A Heuristic Metric for Classification Error Diversity?

A number of authors have tried to qualitatively define classification error diversity. Sharkey [21] suggested a scheme by which an ensemble's pattern of errors could be described in terms of a *level of diversity*. It should be noted that this work concentrated on neural networks, but the ideas are equally applicable to *any* type of classifier. Four levels were proposed, each one describing the incidence and effect of coincident errors amongst the members of the ensemble and the degree to which the target function is covered by the ensemble. A *coincident error* occurs when, for a given input, more than 1 ensemble member gives an incorrect answer. Function *coverage* is an indication of whether or not a test input yields a correct answer on ANY of the individuals in the ensemble.

Level 1 No coincident errors, the target function is covered. Majority vote always produces the correct answer.

Level 2 Some coincident errors, but the majority is always correct and the function is completely covered. The ensemble size must be greater than 4 for this to hold.

Level 3 A majority vote will not always yield the right answer, but the members of the ensemble cover the function such that at least one always has the correct answer for a given input.

Level 4 The function is not always covered by the members of the ensemble.

Sharkey acknowledges that ensembles exhibiting level 2 or 3 diversity could be "upwardly mobile" as it is possible that an ensemble labelled as having level 2 diversity could contain a subset of members displaying level 1 diversity using the test set and a level 3 ensemble could contain subsets of members displaying level 1 and/or level 2 diversity on the test set, thus removal of certain members could result in a change of diversity level for the better.

One problem with this heuristic is that it gives no indication of how *typical* the error behaviour described by the assigned diversity level is, with respect to the test data, nor does it tell us how to *generate* this diversity. According to this metric for denoting error diversity, Sharkey assigns the level which describes the worst case observed; this could happen on the basis of the ensemble performance on only one example from the test data. For example, an ensemble could perform consistently with level 1 diversity on most of a test dataset, then fail on only a single pattern, which would mean that an otherwise good ensemble could be demoted to level 4 diversity. Table 1 shows

the maximum number of classifiers allowed to be in error for a single test pattern in order for each diversity level to hold.

Level	Maximum networks in error
1	1
2	$\frac{M}{2} - 1$ if M is even, or $\frac{M-1}{2}$ if M is odd
3	$M - 1$
4	M

Table 1: Maximum allowable networks in error for each of Sharkey's diversity levels to hold on a single test pattern.

This heuristic is therefore sufficient for illustrative purposes, but intuitively it can be seen that a particular multi-classifier system could exhibit different diversity levels on different subsets of a dataset. These levels may act as a better indication of the ensemble's pattern of errors if they are coupled with the proportions of the test data for which it performs at the described levels. Thus, a distribution of error diversity levels observed during testing could be produced to describe the performance of an ensemble. For example, given a test set of 100 patterns it is plausible that all 4 diversity levels could be observed, as we show in table 2. Here, for the first ensemble, 20 of the data examples produce level 1 diversity, whereas on the second ensemble, 55 examples produce level 1, indicating that the second ensemble has lower error.

	Level 1	Level 2	Level 3	Level 4
Ensemble1	20	25	40	15
Ensemble2	55	25	15	5

Table 2: Distribution of diversity levels across a dataset (number of patterns exhibiting each level of diversity) for two hypothetical ensembles.

According to Sharkey both of these ensembles would be assigned to level 4 diversity despite the fact that the second ensemble performs better than the first as shown by adding in the proportions with which the different levels of diversity occur. This suggested improvement does not, however, give any indication as to *which* members of the ensemble are responsible for which proportions of the different levels of error.

Carney and Cunningham [22] suggested an entropy-based measure, though this does not allow calculation of an individual's contribution to overall diversity. Zenobi and Cunningham [23] proposed a measure of *classification Ambiguity*. The Ambiguity of the i*th* classifier, averaged over N patterns, is

$$A_i = \frac{1}{N} \sum_{n=1}^{N} a_i(\mathbf{x}_n) \tag{17}$$

where $a_i(\mathbf{x}_n)$ is 1 if the output of individual i disagrees with the ensemble output, and 0 otherwise. The overall ensemble Ambiguity is:

$$\bar{A} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{M}\sum_{i=1}^{M}a_i(\mathbf{x}_n) \tag{18}$$

The vast majority of empirical evidence examining classifier diversity is due to Kuncheva [24, 25, 20, 1, 26, 27, 28, 29]. These studies have explored several measures of diversity from the numerical taxonomy literature.

Kuncheva's work emphasizes the existence of two *styles* of measuring diversity, pairwise and non-pairwise. Pairwise measures calculate the average of a particular distance metric between all possible pairings of classifiers in the ensemble. Which distance metric is used therefore determines the characteristics of the diversity measure. The non-pairwise measures either use the idea of entropy or calculate a correlation of each ensemble member with the averaged output. Among the myriad of metrics studied was the *Q-statistic*, which we will now consider. Take two classifiers, f_i and f_j. Over a large set of testing patterns, they will exhibit certain coincident errors, and therefore a probability of error coincidence can be calculated. These are also referred to as the *Oracle* outputs. Table 3 illustrates this, assigning a label a,b,c, or d to each type of coincidence.

	f_i correct	f_j wrong
f_i correct	a	b
f_i wrong	c	d

Table 3: Probabilities of coincident errors between classifier f_i and f_j. It should be noted, by definition, $a + b + c + d = 1$.

The Q statistic between classifier i and classifier j is:

$$Q_{i,j} = \frac{ad - bc}{ad + bc} \tag{19}$$

The Q statistic overall for an ensemble is calculated by taking the average Q-value from all possible pairings of the ensemble members. In addition to the Q statistic, several other metrics were examined. Extensive experimental work was conducted to find a measure of diversity that would correlate well with majority vote accuracy. In a summary paper of her own work, Kuncheva states:

"although a rough tendency was confirmed ... no prominent links appeared between the diversity of the ensemble and its accuracy. Diversity alone is a poor predictor of the ensemble accuracy." [20]

An alternative measure of diversity was advocated by Margineantu and Dietterich [30], the **kappa**-statistic, κ. Using the coincidence matrix as before, *kappa* is defined as:

$$\kappa = \frac{2(ac - bd)}{(a + b)(c + d) + (a + c)(b + d)} \tag{20}$$

Margineantu and Dietterich [30] produced *kappa-error* plots for AdaBoost and Bagging. Each possible pair of classifiers is plotted in a 2-D space of κ against the average error of the pair. This showed up distinctive clouds of points that indicated a definite trade-off between individual accuracy

and the κ measure. Comparing clouds of points for AdaBoost versus Bagging, they verified that AdaBoost produces more diverse ensembles of classifiers.

Yet, in spite of these seemingly intuitive definitions for diversity, none has yet *been proved* to have a definite link to overall ensemble error. It seems the amorphous concept of "diversity" is elusive indeed. We have now reviewed the state of the field with regard to explanations of what the term 'diversity of errors' means, and why it can lead to improved ensemble performance. In conclusion, the community puts great stead in the concept of classification error "diversity", though it is still an ill-defined concept. The lack of a definition for diversity has not stopped researchers attempting to achieve it. So how do you make an effective, well-performing ensemble?

3 Towards A Taxonomy of Methods for Creating Diversity

We determined in the previous section that in both regression and classification contexts, the correlation between the individual predictor outputs has a definite effect on the overall ensemble error, though for classification it is not yet formalised in the literature. In this section we attempt to move towards a possible way to understand the many different methods which researchers use to create an ensemble exhibiting error diversity.

In constructing the ensemble, we may choose to take information about diversity into account, or we may not; i.e. we may or may not explicitly try to optimise some metric of diversity during building the ensemble. We make a distinction between these two types, *explicit* and *implicit* diversity methods. A technique such as Bagging [10] is an *implicit method*, it randomly samples the training patterns to produce different sets for each network; at no point is a measurement taken to ensure diversity will emerge. Boosting [31] is an *explicit method*, it directly manipulates the training data distributions to ensure some form of diversity in the base set of classifiers (although it is obviously not guaranteed to be the 'right' form of diversity).

During learning, a function approximator follows a trajectory in hypothesis space. We would like the individuals in our ensemble to occupy different points in that hypothesis space. While implicit methods rely on randomness to generate diverse trajectories in the hypothesis space, explicit methods deterministically choose different paths in the space. In addition to this high level dichotomy, there are several other possible dimensions for ensuring diversity in the ensemble.

Sharkey [32] proposed that a neural network ensemble could be made to exhibit diversity by influencing one of four things: the initial weights, the training data used, the architecture of the networks, and the training algorithm used. This means providing each ensemble member with a *different* training set, or a *different* architecture, and so on. Though at first this seems a sensible way to group the literature, we found it difficult to group all ensemble techniques under these umbrellas. If we add a penalty to the error function, as in [12, 33, 34], we are changing none of Sharkey's four factors. We instead came to the following categories upon which we believe the majority of ensemble diversity techniques can be placed.

Starting point in Hypothesis Space Methods under this branch vary the starting points within the hypothesis space, thereby influencing where in the space we converge to.

Set of Accessible Hypotheses These methods vary the set of hypotheses that are accessible by the ensemble. Given that certain hypotheses may be made accessible or inaccessible with a particular training subset and function approximator architecture, these techniques either vary training data used, or the architecture employed, for different ensemble members.

Traversal of Hypothesis Space These alter the way we traverse the space, thereby leading different networks to converge to different hypotheses.

It should of course be noted that hypothesis space and search space are not necessarily the same thing. For neural networks, we do not search hypothesis space directly — the search space is the space of possible weight values, which in turn causes different network behaviour. For decision trees, the hypothesis space is directly searched as we construct the trees.

3.1 Starting Point in Hypothesis Space

Starting each network with differing random initial weights will increase the probability of continuing in a different trajectory to other networks. This is perhaps the most common way of generating an ensemble, but is now generally accepted as the least effective method of achieving good diversity; many authors use this as a default benchmark for their own methods [35]. We will first discuss *implicit* instances of this axis, where weights are generated randomly, and then discuss *explicit* diversity for this, where networks are directly *placed* in different parts of the hypothesis space.

Sharkey [36] investigated the relationship between initialisation of the output weight vectors and solutions converged upon with backpropagation. They systematically varied the initial output weight vectors of neural networks throughout a circle of radius 10 and then trained them using the fuzzy XOR task with a fixed set of training data. The resulting networks differed in the number of cycles in which they took to converge upon a solution, and in whether they converged at all. However, the trained neural networks were not found to be statistically independent in their generalisation performance, i.e. they displayed very similar patterns of generalisation despite having been derived from different initial weight vectors. The networks had converged to the same (or very similar) local optima in spite of starting in different parts of the space. Thus, varying the initial weights of neural networks, although important when using a deterministic training method such as backpropagation, seems not to be an effective stand-alone method for generating error diversity in an ensemble of neural networks.

These observations are supported by a number of other studies. Partridge [37, 38] conducted several experiments on large ($> 150,000$ patterns) synthetic data sets, and concludes that after network type, training set structure, and number of hidden units, the random initialization of weights is the least effective method for generating diversity. Parmanto, Munro and Doyle [39] used one synthetic dataset and two medical diagnosis datasets to compare 10-fold cross-validation, Bagging, and random weight initializations; again the random weights method comes in last place.

The methods above are *implicit* diversity methods for manipulating the starting point in hypothesis space. There are very few explicit methods for this, where *randomisation* of weights does not occur; the literature on this topic is disappointingly small. Maclin and Shavlik [40] present an approach to initializing neural network weights that uses competitive learning to create networks that are initialised far from the origin of weight space, thereby potentially increasing the set of reachable local minima; they show significantly improved performance over the standard method of initialization on two real world datasets. A relevant technique is Fast Committee Learning [41] trains a single neural network, taking M snapshots of the state of its weights at a number of instances during the training. The M snapshots are then used as M different ensemble members. Although the performance was not as good as when using separately trained nets, this offers the advantage of reduced training time as it is only necessary to train one network. A modification to this method could be in explicitly *choosing* the M stopping points according to some metric.

3.2 Set of Accessible Hypotheses

It can be argued that there are two ways to manipulate the set of hypotheses accessible to a learner: firstly to alter the training data it receives, and secondly to alter the architecture of the learner itself. We will now discuss these and how they have been used to create error diversity.

3.2.1 Manipulation of Training Data

Several methods attempt to produce diverse or complementary networks by supplying each learner with a slightly different training set. This is probably the most widely investigated method of ensemble training. Regard figure 4; the frontmost bold square represents the training set for our ensemble. Different ensemble members can be given different parts of this set, so they will hopefully learn different things about the same task. Some methods will divide it by training pattern, supplying each member with all the K features, but a different subset of the rows (patterns). Other methods will divide it by feature, supplying each member with all the N patterns in the set, but each consists of a different subset of the columns (features). Both of these are termed *resampling methods*, and could provide overlapping or non-overlapping subsets of the rows or columns (or both) to different learners. Another alternative would be to pre-process the features in some way to get a different representation, for example using a log-scaling of the features. This can be viewed in our diagram as using a different plane, moving in the space of all possible features. The data techniques which transform features are termed *distortion methods* [42].

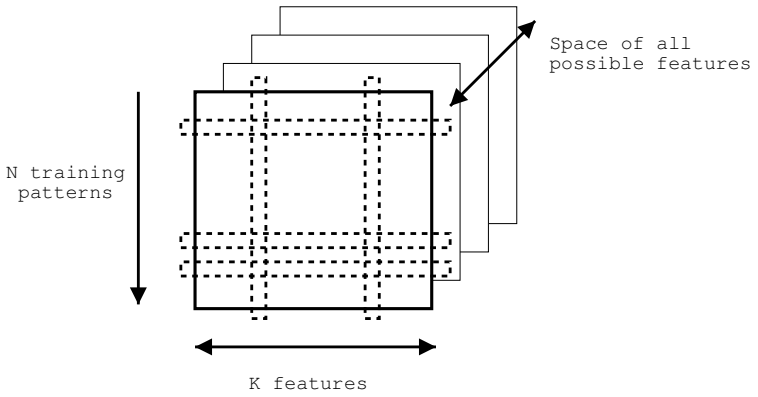

Figure 4: Space of possible training sets for an ensemble

Duin and Tax [43] find that combining the results of one type of classifier on different feature sets is far more effective than combining the results of different classifiers on one feature set. They conclude that the combination of independent information from the different feature sets is more useful than the different approaches of the classifiers on the same data.

The most well-known resampling method is probably *k-fold cross-validation* [2]. By dividing the dataset randomly into k disjoint pattern subsets, new overlapping training sets can be created for each ensemble member, by leaving out one of these k subsets and training on the remainder. The *Bagging* algorithm [10], is another example, randomly selecting N patterns *with replacement* from the original set of N patterns.

Sharkey [44, 45] uses a distortion method to re-represent some features in an engine-fault diagnosis task. Two classifiers are provided with the original data to learn on, while a third is provided with the data after it has been passed through an *untrained* neural network. This essentially applies a random transformation to the features, yet Sharkey shows an ensemble using this technique can outperform an ensemble of classifiers using only the non-transformed data. Intrator and Raviv [46] report that simply adding Gaussian noise to the input data can help. They create a bootstrap resample, like Bagging, but then add a small amount of noise to the input vector. Several ensembles are then trained, using weight regularisation, to get a good estimate of the generalisation error. The process is then repeated with a different noise variance, to determine the optimal level. On test data, they show significant improvements on synthetic and medical datasets.

So far we have discussed how the input patterns could be resampled or distorted; Breiman [47] proposed adding noise to the *outputs* in the training data. This technique showed significant improvements over Bagging on 15 natural and artificial datasets; however when comparing to AdaBoost [31], no improvements were found.

We have now covered a number of papers which use randomisation of the training data to create diversity, therefore these are all so far *implicit diversity methods*. We will now turn to considering *explicit methods*, which deterministically change the data supplied to each network.

The very popular AdaBoost algorithm [31] explicitly alters the distribution of training data fed to each member. The distribution is recalculated in each round, taking into account the errors of the immediately previous network. Oza [48] presents a variant of AdaBoost that calculates the distribution with respect to *all networks* trained so far. In this way the data received by each successive network is explicitly 'designed' so that their errors should be diverse and compensate for one another.

Zenobi and Cunningham [23] use their own metric of classification diversity, as defined in equation (18) to select subsets of features for each learner. They build an ensemble by adding predictors successively, and use their metric to estimate how much diversity is in the ensemble so far. The feature subset used to train a predictor is determined by a hill-climbing strategy, based on the individual error and estimated ensemble diversity. A predictor is rejected if it causes a reduction in diversity according to a pre-defined threshold, or an increase in overall ensemble error. In this case a new feature subset is generated and another predictor trained. The DECORATE algorithm, by Melville and Mooney [49] utilises the same metric to decide whether to accept or reject predictors to be added to the ensemble. Predictors here are generated by training on the original data, plus a 'diversity set' of artificially generated new examples. The input vectors of this set are first passed through the current ensemble to see what its decision would be. Each pattern in the diversity set has its output re-labelled as the *opposite* of whatever the ensemble predicted. A new predictor trained on this set will therefore have a high disagreement with the ensemble, increasing diversity and hopefully decreasing ensemble error. If ensemble error is not reduced, a new diversity set is produced and a new predictor trained. The algorithm terminates after a desired ensemble size or a specified number of iterations is reached.

Oza and Tumer [50] present *Input Decimation Ensembles*, which seeks to reduce the correlations among individual estimators by using different subsets of the input features. Feature selection is achieved by calculating the correlation of each feature individually with each class, then training predictors to be specialists to particular classes or groups of classes. This showed significant benefits on several real [51] and artificial [50] datasets. Liao and Moody [52] demonstrate an information-theoretic technique for feature selection, where all input variables are first grouped based on their mutual information [53, p492]. Statistically similar variables are assigned to the same group, and each

member's input set is then formed by input variables extracted from different groups. Experiments on a noisy and nonstationary economic forecasting problem show it outperforms Bagging and random selection of features.

Several authors use domain knowledge to divide features between predictors. For example, Sharkey and Chandroth [45] use pressure and temperature information to indicate properties of an engine; and Wan [54] combines information from the fossil record with sunspot time series data to predict future sunspot fluctuations.

Most of the methods we have discussed manipulate *input* data. Dietterich and Bakiri [55] manipulate the output targets with *Error-Correcting Output Coding*. Each output class in the problem is represented by a binary string, chosen such that it is orthogonal (or as close as possible) from the representation of the other classes. For a given input pattern, a predictor is trained to reproduce the appropriate binary string. During testing, if the string produced by the predictor does not exactly match the string representing one of the classes, the Hamming distance is measured to each class, and the closest is chosen. Kong and Dietterich [56] investigate why this technique works. They find that, like Bagging, ECOC reduces the variance of the ensemble, but in addition can correct the bias component. An important point to note for this result is that the 0-1 loss bias-variance decomposition utilised assumes a Bayes rate of zero, i.e. zero noise.

3.2.2 Manipulation of Architectures

The number of investigations into using different types of neural network (or different types of learners in general) in ensembles is disappointingly small. If we want diverse errors in our ensemble, it makes sense that using different types of function approximator may produce this. Partridge [37, 57] concludes that variation in numbers of hidden nodes is, after initial weights, the least useful method of creating diversity in neural network ensembles, due to the methodological similarities in the supervised learning algorithms. However, the number of hidden nodes was only varied between 8 and 12, and on a single dataset; such a limited experiment indicates there is still some work to be done here. Partridge also used MLPs and radial basis functions in an ensemble to investigate the effect of network type on diversity, finding this was a more productive route than varying hidden nodes [38].

We have to consider though, that it may be the case that the problem of choosing "compatible" network topologies to place together in an ensemble is simply too hard for a human. Opitz and Shavlik's Addemup algorithm [58], used an evolutionary algorithm to optimise the network topologies composing the ensemble. Addemup trains with standard backpropagation, then selects groups of networks with a good error diversity according to the measurement of diversity. Another recently proposed algorithm, CNNE [59], constructively builds an ensemble, monitoring diversity during the process. CNNE simultaneously designs the ensemble architecture along with training of individual NNs, whilst directly encouraging error diversity. It determines automatically not only the number of NNs in an ensemble, but also the number of hidden nodes in individual NNs. It uses incremental training based on Negative Correlation Learning [33, 12] in training individual NNs. It is entirely possible for an ensemble consisting of networks with very different architectures to emerge in such incrementally constructed ensembles.

Few experiments have been done with *hybrid ensembles*. Wang [60] combined decision trees with neural networks. They found that when the neural networks outnumbered the decision trees, but there was at least one decision tree, the system performed better than any other ratio. Langdon [61] combines decision trees with neural networks in an ensemble, and uses Genetic Programming to

evolve a suitable combination rule. Woods [62] combines neural networks, k-nearest neighbour classifiers, decision trees, and Quadratic Bayes classifiers in a single ensemble, then uses estimates of local accuracy in the feature space to choose one classifier to respond to a new input pattern.

Conclusions from the studies on hybrid ensembles seem to indicate that they will produce estimators with differing specialities and accuracies in different regions of the space—it seems sensible that two systems which represent a problem and search the space in radically different ways may show different strengths, and therefore different patterns of generalisation. This specialisation implies that with hybrid ensembles, *selection* of a single estimator rather than *fusion* of the outputs of all estimators may be more effective. The dynamic selection method by Woods et al [62] could easily be applied to an ensemble containing networks with differing numbers of hidden nodes, having first used an algorithm like CNNE [59] to ensure the network architectures are established appropriately.

3.3 Hypothesis Space Traversal

Given a particular search space, defined by the architecture of the network and training data provided, we could occupy any point in that space to give us a particular hypothesis. How we choose to traverse the space of possible hypotheses determines what type of ensemble we will end up with. We will first discuss *penalty* methods, where the error function of each network is warped with a penalty term to encourage emergence of diverse hypotheses, and secondly we will mention evolutionary search methods, which engage in a *population-based* search of a landscape, and enforce diversity within that population.

3.3.1 Penalty Methods

Some authors have found benefit from using a *penalty term* in the error function of a neural network ensemble. It should be noted that this is not a *regularization* term in the sense of Tikhonov Regularization [63], as much previous ensemble research has shown that regularization of the learners (smoothing the error landscape to prevent overfitting) can be detrimental. We in fact would desire overfitting in individual learners to emerge, as this reduces the bias component of the error, leaving the variance component to be reduced in the ensemble combination [64, 65]. Using a penalty term, the error of network i becomes:

$$e_i = \frac{1}{2}(f_i - d)^2 + \lambda R \tag{21}$$

where λ is a weighting parameter on the penalty term R. The λ parameter controls a trade-off between the two terms; with $\lambda = 0$ we would have an ensemble with each network training with plain backpropagation, and as λ increases more and more emphasis would be placed on minimising whatever the penalty term is chosen to be.

Rosen [34] used a penalty term:

$$R = \sum_{j=1}^{i-1} c(j,i)p_i \tag{22}$$

where $c(j,i)$ is an *indicator function* specifying decorrelation between networks j and i, and p_i is a penalty function:

$$p_i = (f_i - d)(f_j - d) \tag{23}$$

the product of the ith and jth network biases. The indicator function $c(j,i)$ specifies which networks are to be decorrelated. To penalise a network for being correlated with the previous trained network,

the indicator function is:

$$c(j,i) = \begin{cases} 1 & \text{if} \quad i = j-1, \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Negative Correlation (NC) Learning [33] extended Rosen's work by training the networks in parallel. NC has a regularisation term:

$$R = p_i = (f_i - \bar{f}) \sum_{j \neq i} (f_j - \bar{f}) \tag{25}$$

where $\bar{f}$ is the average output of the whole ensemble of M networks at the previous timestep. NC has seen a number of empirical successes [33, 66, 67], consistently *outperforming* a simple ensemble system. In previous work [68] we formalised certain aspects of the NC algorithm, showing it could be applied to *any learning machine* that could minimise the mean square error function, and also defining an upper bound on its parameters. Figure 5 shows our revised version of NC, assuming a gradient descent method is used for learning.

1. Let M be the final number of predictors required.
2. Take a training set $t = \{(\mathbf{x}_1, d_1), ..., (\mathbf{x}_N, d_N)\}$.
3. For each training pattern from $n = 1$ to N do :
   (a) Calculate $\bar{f} = \frac{1}{M} \sum_i f_i(\mathbf{x}_n)$
   (b) For each network from $i = 1$ to M do:
   - Perform a single update for each weight in network i using:

$$e_i = \tfrac{1}{2}(f_i(\mathbf{x}_n) - d_n)^2 - \gamma(f_i(\mathbf{x}_n) - \bar{f})^2$$

$$\tfrac{\partial e_i}{\partial f_i} = (f_i(\mathbf{x}_n) - d_n) - 2\gamma \tfrac{M-1}{M}(f_i(\mathbf{x}_n) - \bar{f})$$

For any new testing pattern $\mathbf{x}$, the ensemble output is given by:

$$\bar{f} = \frac{1}{M} \sum_i f_i(\mathbf{x})$$

Figure 5: The Negative Correlation Learning Algorithm [33, 12]

NC is a particularly important technique when considering the "diversity" issue. For regression ensembles, it has been shown [12] that NC directly controls the covariance term within the bias-variance-covariance trade-off. Therefore it directly adjusts the amount of diversity in the ensemble.

For classification (non-ordinal outputs, as we have described), it does not apply. It would be a very interesting step forward for the field to see an equivalent technique for classification, *directly* controlling the classification diversity term.

McKay and Abbass [69, 70] recently presented *Root Quartic Negative Correlation Learning* (RTQRT), based on an alternative penalty term. The term used was:

$$p_i = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(f_i - d)^4}, \tag{26}$$

The RTQRT-NC technique was applied to a Genetic Programming [71] system, and shown to outperform standard NC on larger ensembles—it is yet to be explained exactly why this is the case. The standard mean squared error function presents a certain error landscape to an ensemble system. NC and RTQRT add penalty terms to this, providing each learner within the ensemble with a different landscape to work in. It has been shown that the minimisation of error in each of these landscapes can reduce error further than using the unmodified landscape. Therefore these can be considered as *explicit* diversity methods, directly encouraging convergence to different parts of the hypothesis space.

3.3.2 Evolutionary Methods

The term "diversity" is also often used in the evolutionary computation literature, in the context of maintaining diversity in the population of individuals you are evolving [72, 73, 74]. This has a very different meaning to the concept of diversity as discussed in this article. In evolutionary algorithms, we wish to maintain diversity in order to ensure we have *explored a large area of the search space* and not focussed our search onto an unprofitable area. When we decide our search has continued for long enough, we will typically take the best performing individual found so far, *ignoring the other individuals in the population.* The optimal diversity in this context will be that which optimises the explore-exploit trade off, such that the best points in the search space are found quickly and efficiently. This is in contrast to ensemble diversity methods, which create diversity with the intention that the 'population' of ensemble members will *be combined.* As such, evolutionary diversity methods do not concern themselves with creating a population that is *complementary* in any way, but instead with just ensuring the maximum amount of the hypothesis space is being explored in order to find the best single individual. In spite of these differences, some researchers have found interesting parallels. Yao and Liu [75] evolves a population of neural networks, using *fitness sharing* techniques to encourage diversity, then combines the *entire* population as an ensemble instead of just picking the best individual. Khare and Yao [74] extend the concept for classification data by using the Kullback-Leibler entropy as a diversity metric to optimise during the evolutionary search.

4 Discussion

4.1 How to Categorise Multiple Classifier Systems?

We have now concluded our categorisation of the diversity literature. We fully accept that we have presented a rather absract taxonomy, categorising the field around very broad axes, however we

believe a clear statement of the problem we are tackling and a clear representation of the state of the field may help progress research. The problem of how to describe and summarise a field concisely is a difficult one. Our categorisation is in fact the result of several prototypes, none of which (including this one) will be entirely satisfactory for all. Ho [76] divides up the space of classifier combination techniques into *coverage* optimisation and *decision* optimisation—the diversity creation methods we have described would seem to come under the branch of coverage optimisation. Sharkey [77] proposed a categorisation scheme for multi-network architectures. An architecture is categorised on whether it is *competitive or cooperative*, and whether it is *top-down or bottom-up*. A competitive architecture will in some fashion select a single network to give the final answer; a cooperative architecture will fuse the outputs of all networks. An architecture is top-down if the combination mechanism is based on something other than the network outputs. The mere fact that there are so many equally valid ways to categorise a field pays tribute to it—the wide interdisciplinarity in multiple classifier systems means we will always have multiple ways to divide up the space.

As a conclusion to our proposed decomposition of the field, we note one interesting family of algorithms that all exploit the Ambiguity decomposition. Krogh and Vedelsby [2] showed that the ensemble error could be broken down into two terms, one of which is dependent on the correlations between networks. A number of techniques have explicitly measured Ambiguity and used it as a guide for constructing an ensemble, being utilised in almost every aspect of ensemble construction. Krogh and Vedelsby themselves used an active learning scheme [2], based on the method of query by committee, selecting patterns to train on that had a large Ambiguity. This showed significant improvements over passive learning in approximating a square wave function. In the same paper an estimate of Ambiguity is used to optimise the ensemble combination weights; this showed in some cases it is optimal to set a network weight to zero, essentially removing it from the ensemble.

In previous work [68] we showed that Negative Correlation learning [33] uses the Ambiguity term as a penalty in the error function of each network. This means we can optimise ensemble performance by tuning the emphasis on diversity in the error function used the strength parameter. Opitz [78] selected feature subsets for the ensemble members to train on, using a Genetic Algorithm (GA) with an Ambiguity-based fitness function; this showed gains over Bagging and Adaboost on several classification datasets from the UCI repository. A precursor to this work was Opitz and Shavlik's Addemup algorithm [58], which used the same fitness function to optimise the network topologies composing the ensemble. Interestingly, both these GA-based approaches also used a strength parameter, λ, to vary the emphasis on diversity. The difference between their work and NC is that NC incorporates Ambiguity into the backpropagation weight updates, while Addemup trains with standard backpropagation, then selects networks with a good error diversity.

In summary, Ambiguity has been utilised in many ways: pattern selection [2], feature selection [78], optimising the topologies [58] of networks in the ensemble, optimising the combination function [2], and finally optimising the network weights themselves [68].

4.2 How to Quantify Classification Diversity?

As we stated earlier, the "diversity" problem is really one of quantifying correlation between non-ordinal classifier outputs. There is as yet no natural analogue of the bias-variance-covariance decomposition. A step toward understanding this question further can be taken by considering where the bias-variance-covariance decomposition comes from: it falls neatly out of the bias-variance decomposition of the ensemble error. However, when our classification of a data point is either correct or incorrect, we have a zero-one loss function (instead of the usual quadratic loss function we used for

the regression context). A number of authors have attempted to define a bias-variance decomposition for zero-one loss functions [56, 79, 80, 81], each with their own assumptions and shortcomings. Most recently Domingos [82] and James [83] propose general definitions which include the original quadratic loss function as a special case. This leads us naturally to ask the question, *does there exist an analogue to the bias-variance-covariance decomposition that applies for zero-one loss functions?* If so, its formulation of the "covariance" term will be a major stepping stone in our understanding of the role of classification error diversity. The optimal classification error diversity will then be understood in terms of this trade-off for zero-one loss functions.

5 Conclusions

In this article we reviewed the existing qualitative and quantitative explanations of what error "diversity" is, and how it affects the error of the overall ensemble. For a clear specification of the problem, we covered diversity in both regression and classification contexts. We described the two most prominent theoretical results for regression ensembles: the Ambiguity decomposition and the bias-variance-covariance decomposition. We demonstrated what we believe to be the first formal link between these two, making it explicit how they relate. We described the current state of the art in formalising the concept of diversity for classification ensembles, illustrating clearly that the problem is actually one of quantifying some form of correlation between non-ordinal predictor outputs.

In the process of this we suggested a modification to an existing heuristic measure [21] of classification error diversity, that accounts for variability in the test data. We believe this allows more fine-grained judgement about whether an ensemble is performing well. In addition, we suggested directions to take in understanding a more formal grounding for diversity, around studies of the bias-variance-covariance decomposition [3] and the generalised bias-variance decomposition [83] for zero-one loss. This is the subject of our current research.

The main contribution of this article has been a thorough survey and categorisation of the literature according to how ensemble techniques choose to encourage diversity. We dichotomised techniques according to whether they choose to explicitly enforce diversity via some metric, or whether they implicitly encourage diversity by randomisation methods. We then grouped techniques according to three factors: how they initialise the learners in the hypothesis space, what the space of accessible hypotheses is, and how that space is traversed by the learning algorithm.

Though we note that such a taxonomy is bound to be the subject of heated debate (and this is healthy for the field) we believe this categorisation could help to identify gaps in this exciting, rapidly expanding, field.

References

[1] L. Kuncheva, C.Whitaker, Measures of diversity in classifier ensembles, Machine Learning (51) (2003) 181–207.

[2] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, NIPS 7 (1995) 231–238.

[3] N. Ueda, R. Nakano, Generalization error of ensemble estimators, in: Proceedings of International Conference on Neural Networks, 1996, pp. 90–95.

[4] J. Friedman, B. Popescu, Importance sampling learning ensembles, Tech. rep., Stanford University (September 2003).
URL `http://www-stat.stanford.edu/~jhf/ftp/isle.pdf`

[5] M. Perrone, Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization, Ph.D. thesis, Brown University, Institute for Brain and Neural Systems (1993).

[6] S. Hashem, Optimal Linear Combinations of Neural Networks, Ph.D. thesis, School of Industrial Engineering, University of Purdue (1993).

[7] J. M. Bates, C. W. J. Granger, The combination of forecasts, Operations Research Quarterly (20) (1969) 451–468.

[8] C. Granger, Combining forecasts – twenty years later, Journal of Forecasting 8 (1989) 167–174.

[9] R. Clemen, Combining forecast: A review and annotated bibliography, International Journal on Forecasting 5 (1989) 559–583.

[10] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.

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

[12] G. Brown, Diversity in neural network ensembles, Ph.D. thesis, School of Computer Science, University of Birmingham (2004).
URL `http://www.cs.bham.ac.uk/~gxb/research/`

[13] K. Tumer, J. Ghosh, Error correlation and error reduction in ensemble classifiers, Connection Science 8 (3-4) (1996) 385–403.

[14] G. Fumera, F. Roli, Linear combiners for classifier fusion: Some theoretical and experimental results, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 2709), Springer, Guildford, Surrey, 2003, pp. 74–83.
URL `http://link.springer.de/link/service/series/0558/tocs/t2709.htm`

[15] F. Roli, G. Fumera, Analysis of linear and order statistics combiners for fusion of imbalanced classifiers, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 2364), Springer, Calgiari, Italy, 2002, pp. 252–261.
URL `http://link.springer.de/link/service/series/0558/tocs/t2364.htm`

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

[17] K. Tumer, J. Ghosh, Theoretical foundations of linear and order statistics combiners for neural pattern classifiers, Tech. Rep. TR-95-02-98, Computer and Vision Research Center, University of Texas, Austin (1995).

[18] L. K. Hansen, P. Salamon, Neural network ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (10) (1990) 993–1001.

[19] L. Breiman, Random forests—random features, Tech. Rep. 567, University of California, Berkley (Dept of Statistics) (1999).

[20] L. Kuncheva, That Elusive Diversity in Classifier Ensembles, in: First Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), available as LNCS volume 2652, 2003, pp. 1126–1138.

[21] A. Sharkey, N. Sharkey, Combining diverse neural networks, The Knowledge Engineering Review 12 (3) (1997) 231–247.

[22] P. Cunningham, J. Carney, Diversity versus quality in classification ensembles based on feature selection, in: LNCS - European Conference on Machine Learning, Vol. 1810, Springer, Berlin, 2000, pp. 109–116.

[23] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, Lecture Notes in Computer Science 2167 (2001) 576–587.

[24] L. Kuncheva, C.Whitaker, C.Shipp, R.Duin, Limits on the majority vote accuracy in classifier fusion, Pattern Analysis and Applications 6 (1) (2003) 22–31.

[25] L. Kuncheva, C. Whitaker, C. Shipp, R. Duin, Is independence good for combining classiers, in: Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 2000, pp. 168–171.

[26] L. Kuncheva, R. Kountchev, Generating classifier outputs of fixed accuracy and diversity, Pattern Recognition Letters (23) (2002) 593–600.

[27] L. I. Kuncheva, C. J. Whitaker, Ten measures of diversity in classifier ensembles: Limits for two classifiers, in: IEE Workshop on Intelligent Sensor Processing, IEE, 2001, Birmingham, UK.

[28] C. Shipp, L. Kuncheva, Relationships between combination methods and measures of diversity in combining classifiers, Information Fusion (3) (2002) 135–148.

[29] M. Skurichina, L. Kuncheva, R. P. W. Duin, Bagging and boosting for the nearest mean classifier: Effects of sample size on diversity and accuracy, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 2364), Springer, Calgiari, Italy, 2002, pp. 62–71.
URL `http://link.springer.de/link/service/series/0558/tocs/t2364.htm`

[30] D. D. Margineantu, T. G. Dietterich, Pruning adaptive boosting, in: Proc. 14th International Conference on Machine Learning, Morgan Kaufmann, 1997, pp. 211–218.
URL `http://citeseer.nj.nec.com/margineantu97pruning.html`

[31] Y. Freund, R. E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the 13th International Conference on Machine Learning, Morgan Kaufmann, 1996, pp. 148–156.

[32] A. Sharkey, Multi-Net Systems, Springer-Verlag, 1999, Ch. Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems, pp. 1–30.

[33] Y. Liu, Negative correlation learning and evolutionary neural network ensembles, Ph.D. thesis, University College, The University of New South Wales, Australian Defence Force Academy, Canberra, Australia (1998).

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

[35] D. Opitz, R. Maclin, Popular ensemble methods: An empirical study, Journal of Artificial Intelligence Research 11 (1999) 169–198.

[36] N. Sharkey, J. Neary, A. Sharkey, Searching Weight Space for Backpropagation Solution Types, Current Trends in Connectionism: Proceedings of the 1995 Swedish Conference on Connectionism (1995) 103–120.

[37] D. Partridge, W. B. Yates, Engineering multiversion neural-net systems, Neural Computation 8 (4) (1996) 869–893.

[38] W. Yates, D. Partridge, Use of methodological diversity to improve neural network generalization, Neural Computing and Applications 4 (2) (1996) 114–128.
URL `http://citeseer.nj.nec.com/partridge95use.html`

[39] B.Parmanto, P.W.Munro, H.R.Doyle, Improving committee diagnosis with resampling techniques, Advances in Neural Information Processing Systems 8 (1996) 882–888.

[40] R. Maclin, J. W. Shavlik, Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995, pp. 524–530.

[41] A. Swann, N. Allinson, Fast committee learning: Preliminary results, Electronics Letters 34 (14) (1998) 1408–1410.

[42] A. J. C. Sharkey, N. E. Sharkey, U. Gerecke, G. O. Chandroth, The test and select approach to ensemble combination, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 1857), Springer, Calgiari, Italy, 2000, pp. 30–44.

[43] R. P. W. Duin, D. M. J. Tax, Experiments with classifier combining rules, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 1857), Springer, Calgiari, Italy, 2000, pp. 16–29.
URL `http://link.springer.de/link/service/series/0558/tocs/t1857.htm`

[44] A. Sharkey, N. Sharkey, Diversity, selection, and ensembles of artificial neural nets, in: Neural Networks and their Applications (NEURAP'97), 1997, pp. 205–212.

[45] A. Sharkey, N. Sharkey, G. Chandroth, Diverse neural net solutions to a fault diagnosis problem, Neural Computing and Applications 4 (1996) 218–227.

[46] Y. Raviv, N. Intrator, Bootstrapping with noise: An effective regularisation technique, Connection Science 8 (1996) 355–372.

[47] L. Breiman, Randomizing outputs to increase prediction accuracy, Technical Report 518, Statistics Department, University of California (May 1998).
URL `http://www.boosting.org/papers/Bre98.pdf`

[48] N. C. Oza, Boosting with averaged weight vectors, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 2709), Springer, Guildford, Surrey, 2003, pp. 15–24.

[49] P. Melville, R. Mooney, Constructing diverse classifier ensembles using artificial training examples, in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Mexico, 2003, pp. 505–510.

[50] N. C. Oza, K. Tumer, Input decimation ensembles: Decorrelation through dimensionality reduction, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 2096), Springer, Cambridge, UK, 2001, pp. 238–247.
URL `http://link.springer.de/link/service/series/0558/tocs/t2096.htm`

[51] N. Oza, K. Tumer, Dimensionality reduction through classifier ensembles, Tech. Rep. NASA-ARC-IC-1999-126, NASA Ames Labs (1999).

[52] Y. Liao, J. Moody, Constructing heterogeneous committees using input feature grouping, Advances in Neural Information Processing Systems 12.

[53] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan Co., New York, 1994.

[54] E. A. Wan, Combining fossil and sunspot data: Committee predictions, in: International Conference On Neural Networks (ICNN97), 1997.
URL `http://citeseer.ist.psu.edu/146595.html`

[55] T. G. Dietterich, G. Bakiri, Error-correcting output codes: a general method for improving multiclass inductive learning programs, in: T. L. Dean, K. McKeown (Eds.), Proceedings of the Ninth AAAI National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 1991, pp. 572–577.

[56] E. B. Kong, T. G. Dietterich, Error-correcting output coding corrects bias and variance, in: Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann, 1995, pp. 313–321.

[57] D. Partridge, Network generalization differences quantified, Neural Networks 9 (2) (1996) 263–271.
URL `http://citeseer.nj.nec.com/partridge94network.html`

[58] D. W. Opitz, J. W. Shavlik, Generating accurate and diverse members of a neural-network ensemble, NIPS 8 (1996) 535–541.

[59] M. M. Islam, X. Yao, K. Murase, A constructive algorithm for training cooperative neural network ensembles, IEEE Transactions on Neural Networks 14 (4) (2003) 820–834.

[60] W. Wang, P. Jones, D. Partridge, Diversity between neural networks and decision trees for building multiple classifier systems, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 1857), Springer, Calgiari, Italy, 2000, pp. 240–249.
URL `http://link.springer.de/link/service/series/0558/tocs/t1857.htm`

[61] W. B. Langdon, S. J. Barrett, B. F. Buxton, Combining decision trees and neural networks for drug discovery, in: Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002, Kinsale, Ireland, 2002, pp. 60–70.

[62] K. Woods, W. Kegelmeyer, K. Bowyer, Combination of multiple classiers using local accuracy estimates, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 405–410.

[63] A. N. Tikhonov, V. Y. Arsenin, Solutions of Ill-posed problems, W.H.Winston and Sons, Washington D.C., 1977.

[64] P. Sollich, A. Krogh, Learning with ensembles: How overfitting can be useful 8 (1996) 190–196.

[65] A. K. Husmeier D., Modelling conditional probabilities with network committees: how overfitting can be useful, Neural Network World 8 (1998) 417–439.

[66] Y. Liu, X. Yao, Negatively correlated neural networks can produce best ensembles, Australian Journal of Intelligent Information Processing Systems 4 (3/4) (1997) 176–185.

[67] Y. Liu, X. Yao, Ensemble learning via negative correlation, Neural Networks 12 (10) (1999) 1399–1404.

[68] G. Brown, J. L. Wyatt, The use of the ambiguity decomposition in neural network ensemble learning methods, in: T. Fawcett, N. Mishra (Eds.), 20th International Conference on Machine Learning (ICML'03), Washington DC, USA, 2003.

[69] R. McKay, H. Abbass, Anticorrelation measures in genetic programming, in: Australasia-Japan Workshop on Intelligent and Evolutionary Systems, 2001, pp. 45–51.

[70] R. McKay, H. Abbass, Analyzing anticorrelation in ensemble learning, in: Proceedings of 2001 Conference on Artificial Neural Networks and Expert Systems, Otago, New Zealand, 2001, pp. 22–27.

[71] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.

[72] J.-H. Ahn, S.-B. Cho, Speciated neural networks evolved with fitness sharing technique, in: Proceedings of the Congress on Evolutionary Computation, Seoul, Korea, 2001, pp. 390–396.

[73] P. J. Darwen, X. Yao, Speciation as automatic categorical modularization, IEEE Trans. on Evolutionary Computation 1 (2) (1997) 100–108.

[74] V. Khare, X. Yao, Artificial speciation of neural network ensembles, in: J.A.Bullinaria (Ed.), Proc. of the 2002 UK Workshop on Computational Intelligence (UKCI'02), University of Birmingham, UK, 2002, pp. 96–103.

[75] X. Yao, Y. Liu, Making use of population information in evolutionary artificial neural networks, in: IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 28, IEEE Press, 1998, pp. 417–425.

[76] T. K. Ho, Data complexity analysis for classifier combination, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 2096), Springer, Cambridge, UK, 2001, pp. 53–67.
URL `http://link.springer.de/link/service/series/0558/tocs/t2096.htm`

[77] A. J. C. Sharkey, Types of multinet system, in: Proc. Int. Workshop on Multiple Classifier Systems (LNCS 2364), Springer, Calgiari, Italy, 2002, pp. 108–117.
URL `http://link.springer.de/link/service/series/0558/tocs/t2364.htm`

[78] D. Opitz, Feature selection for ensembles, in: Proceedings of 16th National Conference on Artificial Intelligence (AAAI), 1999, pp. 379–384.

[79] R. Kohavi, D. H. Wolpert, Bias plus variance decomposition for zero-one loss functions, in: L. Saitta (Ed.), Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kaufmann, 1996, pp. 275–283.

[80] L. Breiman, Bias, variance, and arcing classifiers, Tech. Rep. 460, Statistics Department, University of California, Berkeley (1996).

[81] J. Friedman, Bias, variance, 0-1 loss and the curse of dimensionality, Tech. rep., Stanford University (1996).

[82] P. Domingos, A unified bias-variance decomposition and its applications, in: Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 231–238.
URL `http://citeseer.nj.nec.com/domingos00unified.html`

[83] G. James, Variance and bias for general loss functions, Machine Learning 51 (2003) 115–135.