

Multi-objective Ensemble Construction, Learning and Evolution

Arjun Chandra and Xin Yao

The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom
{A.Chandra|X.Yao}@cs.bham.ac.uk

Abstract. An ensemble of learning machines has been theoretically and empirically shown to generalise better than single learners. Diversity and accuracy are two key properties that ensemble members should possess in order for this generalisation principle to hold. Viewing these properties as objectives, we take the position of rendering multi-objective evolutionary algorithms as effective solution concepts to the problem of ensemble construction and learning.

1 Introduction

A good portion of machine learning research is devoted to studying the issue of generalisation. A learner is said to possess the ability to generalise, if it performs well with predicting unseen input-output mappings. The importance of generalisation in neural computing research is aptly captured in the theoretical result known as the ‘bias-variance dilemma’ [21].

An ensemble of learning machines has been shown to outperform (generalise better than) single learners both theoretically and empirically [9]. Tumer and Ghosh [39] present the formal proof of this. Although ensembles perform better than their members, constructing them is not an easy task. Krogh and Vedelsby [27] formally show that an ideal ensemble is one that consists of highly correct (accurate) predictors which at the same time disagree as much as possible (i.e. substantial diversity amongst members is exhibited). This has also been tested and empirically verified as referred to in [35] and shown in [36]. Thus, diversity and accuracy are two key properties that ensemble members should possess [19, 22] in order for the ensemble to generalise effectively. However, there exists a trade-off between these two properties, as will be shown later on (also mentioned in [41]), giving a multi-objective flavour to the problem of constructing ensembles.

Of the multitude of approaches available for ensemble construction and learning, one of the more recent ones views learning as a multi-objective optimisation problem [3]. Recently, we proposed an algorithm called DIVACE (DIVERse and ACCurate Ensemble learning algorithm) [12, 14] which uses good ideas from Negative Correlation Learning (NCL)[30] and the Memetic Pareto Artificial Neural

Network (MPANN)[1, 3] algorithm, and formulates the ensemble learning problem as a multi-objective problem explicitly within an evolutionary setup aiming at finding a good trade-off between the key properties mentioned above viz. diversity and accuracy.

One very strong motivation for the use of evolutionary multi-criterion optimisation in the construction of an ensemble is that, due to the presence of multiple conflicting objectives, the evolutionary approach engenders a *set* of near optimal solutions. The presence of more than one optimal solution indicates that if one uses multi-objectivity while creating ensembles, one can actually generate an ensemble automatically where the member networks would inadvertently be near optimal [10, 12].

This is essentially a position paper, promoting multi-objective evolutionary algorithms as solution concepts to the problem of ensemble construction and learning. We consider the general problem of learning and discuss it with the all pervasive notion of generalisation in the backdrop. A formal discussion of why and how ensembles help achieve better generalisation (as compared to single learners) reveals the properties of diversity and accuracy, which are ascertained to be the key elements that ensemble members should be built around. These properties can be treated as objectives for the evolution of learners and the construction of the ensemble thereof. We review and discuss some recent work along these lines.

2 Learning, Generalisation and the Bias-Variance Dilemma

Learning the exact input-output mapping to recognise an unseen input pattern becomes hard in practice due to the dearth in the amount of, and noise in, the data used for learning. The idea behind training is simply to inculcate a statistical model (a statistical oracle) of the mapping being learnt within a base model, manifested in the settings of its parameters and complexity, and not to memorise this mapping [6]. In most training algorithms, there exists a bound on the amount of training necessary beyond which the base model generated would tend towards becoming more and more inflexible, in that, it would try to ‘memorise’ the exact representation of the data, thus making it more susceptible to giving a wrong input-output mapping for unseen inputs. There also exists a lower bound on the amount of training necessary as below this the base model would become a random guesser.

Given that a learning algorithm is presented with only a subset of the training patterns, the learning objective is to make the statistical model as fitting as possible to the training data and at the same time let it remain smooth enough such that if the predictor is trained using another related data set and fitted on it, the model does not become significantly different. This statement exposes what is expressed by two much touted terms in neural computation research: *bias* and *variance*. Bias and variance are conflicting terms as far as achieving a good fit to the training data is concerned and there exists a trade-off between

them [6], i.e. if we try to reduce the bias, the variance increases and vice-versa. Geman et al. [21] explain this trade-off very well and show that the expected error function can be decomposed into *bias* and *variance* components.

Let the actual mapping function be $f(x)$ and the desired function be $\Phi(x)$. Consider the mean square error function,

$$(f(x) - \Phi(x))^2, \quad (1)$$

the value of which depends ($E\{\cdot\}$ is used to get rid of this dependence) on the data set used for training. An error function (Equation 2) which is independent of the initial conditions for a predictor as well as the choice of the training set is given by:

$$E\{(f(x) - \Phi(x))^2\} \quad (2)$$

The bias-variance decomposition just alluded to (and according to [21]), can be expressed as follows (refer to [11, 14] for details):

$$\begin{aligned} E\{(f(x) - \Phi(x))^2\} &= E\{(f(x) - E\{f(x)\})^2\} + (E\{f(x)\} - \Phi(x))^2 \\ &= \text{Variance} + \text{Bias}^2 \end{aligned} \quad (3)$$

Let us say we have a training set D and there are two curves (polynomial functions) expressing the functional mapping this data set represents. Considering two extrema for these polynomial functions viz. a straight line/constant linear function (simple model) and a function that fits the points perfectly (complex model), the decomposition and the terms therein can be elaborated on. Note that the more complex a model, the more configurable parameters it has [8].

Figure 1 shows 3 curves: desired function, simple model and complex model, the dots representing the data set.

Bias: In simple terms, bias signifies the model being different (on an average over all data sets and initial conditions) from the desired mapping. From equation 3,

$$\text{Bias}^2 = (E\{f(x)\} - \Phi(x))^2 \quad (4)$$

With the simple model, $f(x)$ is independent of the data set D and differs from the desired function $\Phi(x)$ mainly because D was not considered at all. Consequently, the simple model can be thought of as a random function and it has a high bias.

With the complex model, the bias term becomes very small. This happens because from equation 4,

$$E\{f(x)\} = \text{complex overfitted function} \approx \Phi(x) \quad (5)$$

Hence obviously,

$$(\Phi(x) - \Phi(x))^2 = 0 \quad (6)$$

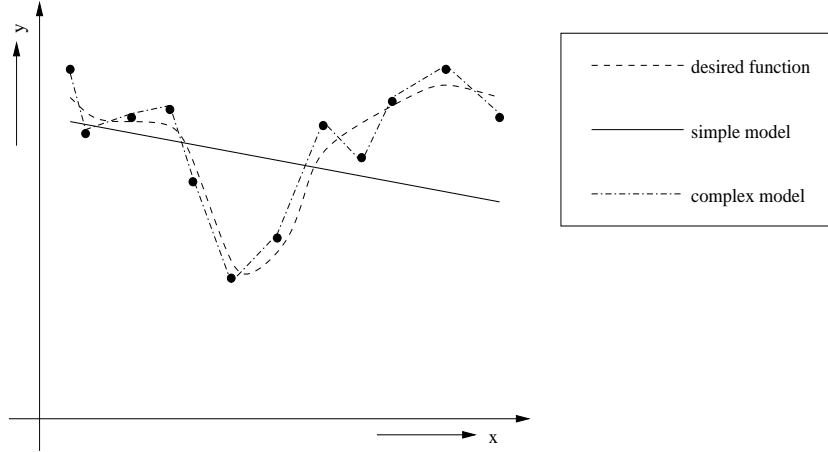


Fig. 1. The role played by bias and variance with respect to the network function complexity.

Variance: Variance can also be termed as the *sensitivity* of a model to the training set. Here again, from equation 3 we have

$$Variance = E\{(f(x) - E\{f(x)\})^2\}. \quad (7)$$

Equation 7 describes how different the statistical model is from its expected value over all possible weight initialisations and choice of training sets. A point to emphasise here is that, $E\{f(x)\}$ depends on the choice of the training set which means that if $f(x)$ is independent of the training set, the variance term for such a model is zero. However, for a perfectly fitting $f(x)$, this term plays the main role as far as the mean square error is concerned (in which case the bias term is zero).

The simple model, being independent of the training set D , will exhibit *zero* variance. For the complex model, from equations 5 and 7,

$$E\{(f(x) - E\{f(x)\})^2\} = E\{(f(x) - \Phi(x))^2\}. \quad (8)$$

Since $f(x)$ and $\Phi(x)$ are not equal, due mainly to the fact that $f(x)$ is largely dependent on the choice of the training set D and always overfits it, we could imagine the variance as being quite large.

So, the dilemma is that if we try to use a simple model, we get a large bias and small variance, whereas, if we use a complex model, we get a small bias, but a large variance. All in all, the average generalisation error remains large. We need to find an optimal balance between bias and variance in order to reduce this error and make it as small as possible. Obviously, the solution here is to choose a model which is neither too simple, nor too complex, i.e. a model that exhibits an optimum trade-off between bias and variance. The simplest

strategy to achieve a good trade-off between bias and variance is to use more data (well distributed over the input space) for training [6]. Bishop [6] gives a comprehensive account of techniques in tackling this dilemma. Some of these are regularisation, early stopping, training with noise, soft weight sharing, growing and pruning algorithms etc. Another way is to use ensembles.

3 Ensembles for Generalisation

3.1 Need for Ensembles

Here, we elaborate on some key properties which make an ensemble¹ click. The problem of designing a learner is often posed as a search problem. Machine learning algorithms usually work by searching for the best possible hypothesis (learner) in the space of possible hypotheses \mathbf{H} [19].

Firstly, the space \mathbf{H} is very large and a large amount of training data is required to restrict the search to the most feasible regions. By eliminating the infeasible regions in accordance with the training data, we might end up with good approximations of the optimal hypothesis. Not combining these may lead to loss of useful information as all the remaining hypotheses are equally attractive (equally accurate).

Secondly, more often than not, being greedy search algorithms applying local optimisation techniques (e.g. gradient decent for multi-layer perceptrons), learning algorithms get stuck in a local optima. Finding the best hypothesis which is optimal both in terms of complexity and accuracy is considered as an NP-complete problem [7], connoting the need for search heuristics. In theory, even if the best hypothesis can be found, the use of heuristics prevents it from finding it in practise. The best solution here is to train multiple base models starting from different points in the hypothesis space and combine the hypotheses found.

Additionally, the deficiencies of learning algorithms and the nature of search spaces together let ensembles provide a complementary solution. According to [19], there may be problems for which there is no hypothesis in the possible space of hypotheses due to limitations in the representation capabilities of learning algorithms [41], i.e. the actual solution is not part of the infinitum of solutions that can be obtained from the algorithms. In such cases, there may be good approximations that can be combined in some way to actually shift the resulting model towards the true hypothesis. As Valentini et al. [41] state "...combination of hypotheses... can expand the space of representable functions, embracing also the true one".

Apart from this, ensembles tend to be more robust/reliable as compared to single learners. With an ensemble, there is always a chance that some subset of the learners will perform well even if others are not. Some researchers have addressed the issue of designing ensembles from a software engineering perspective

¹ An ensemble, as the name suggests, is a collection of learning machines essentially used to replace a single learner/predictor in order to achieve a co-operative (or competitive) decision when mapping an input space onto an output space.

[38, 42, 44] where the idea of N-version programming/multi-version programming has been considered. Multi-version programming deals with producing multiple versions of a program such that each fails independently (or in an uncorrelated manner) [29] and where these versions can be combined by some means (such as a majority vote) to give a more reliable system [38]. Sharkey et al. [38] refer to this as being a reliability through redundancy approach which is a standard way of designing conventional software (as shown by Littlewood and Miller in [29]) and hardware [37] for safety critical systems.

Perhaps a more defining viewpoint is that of Liu and Yao [32] where they stress the fact that ensembles adopt a divide and conquer strategy where the individuals learn to subdivide the problem (decompose the problem) at hand and solve the respective subproblems more efficiently. There are various ways this problem decomposition can take place and diversity² plays an important role in order for the decomposition to be successful, as will be seen shortly.

Section 3.2 gives a more formal view on why ensembles work.

3.2 Why Ensembles Click?

Here, we formally discuss how single learners affect the performance of an ensemble they form part of, leading to the revelation of certain properties they should possess in order for the ensemble to generalise effectively.

Ambiguity Decomposition: Brown et al. [9] give a very clear description on the utility of ensembles (ensembles of neural networks) in the regression context although the whole idea presented can be applied to a classification task (if we consider the outputs of the predictors as the posterior class probabilities). However, no formal proof has been proposed for base models that output non-ordinal values, although there is empirical evidence which models the unknown formalism [9].

Suppose that we have trained a neural network using backpropagation (hereafter referred to as BP) on some data set D . For any test pattern (\mathbf{x}, \mathbf{y}) with input \mathbf{x} , we will get some prediction for \mathbf{y} , which might not be the same as \mathbf{y} . Training another network on the same data set may result in another hypothesis where the output might again not be the same as what is desired but it might also *differ* from the output of the first network.

According to [9], if many such networks are considered, each differing on their initial weight settings (each network, to begin with, has its weights initialised randomly), the outputs from each network for the test data point (\mathbf{x}, \mathbf{y}) will follow a distribution dependent on the data set D used for training them and the weight initialisations. Consequently, this would also be true for the errors that they make. The mean of the distribution is the expected value $E\{f(\mathbf{x})\}$ for any given input \mathbf{x} .

² By diversity, we mean decision failures of ensemble members on any given dataset being uncorrelated.

Since we have many networks, each will have a different estimate/prediction of the output, given an input \mathbf{x} , but *if we take the average of these predictions we might get a better estimate of the output as compared to estimates from individual networks*. This, in essence, is what an ensemble does. Consequently, if we have an infinite number of networks, we would certainly come very close to, if not be the same as the exact value for the output \mathbf{y} from $E\{f(\mathbf{x})\}$ given an input \mathbf{x} , provided the individual networks *differ* in their predictions and hence *differ* in the errors that they make. This point emphasises the notion of *diversity* which is a very important ingredient in constructing good ensembles. The main idea is that if we combine estimates from a number of predictors which have been trained on the same data set (yet have different error estimates for any given test input mainly because of the different random weight initialisations) a combined estimate of the outputs is likely to be better than the output from individual predictors.

Following is a more formal result as to the effectiveness of ensemble methods, which was presented by Krogh and Vedelsby [27] and is also considered in [9].

Let f_{ens} be the ensemble output given by

$$f_{ens} = \sum_i w_i f_i,$$

where f_i is the output of network i . f_{ens} is a convex combination, i.e. $\sum_i w_i = 1$ and $\forall i, w_i \geq 0$. Let Φ be the desired output. The average quadratic error of the member networks, i.e. $\sum_i w_i (f_i - \Phi)^2$, can be shown (refer to [11, 14] for details) to be:

$$\begin{aligned} \sum_i w_i (f_i - \Phi)^2 &= \sum_i w_i (f_i - f_{ens})^2 + (f_{ens} - \Phi)^2 \\ \Rightarrow (f_{ens} - \Phi)^2 &= \sum_i w_i (f_i - \Phi)^2 - \sum_i w_i (f_i - f_{ens})^2. \end{aligned} \quad (9)$$

Equation 9 says that the mean square error/quadratic error of the ensemble (the left hand side of the equation) is guaranteed to be less than or equal to the average quadratic error of the member networks (the first term in the right hand side of the equation).

Mathematically,

$$(f_{ens} - \Phi)^2 \leq \sum_i w_i (f_i - \Phi)^2, \quad (10)$$

In Equation 9, the second term on the right hand side (i.e. $\sum_i w_i (f_i - f_{ens})^2$) will always be greater than or equal to zero. This is often called the ambiguity term and it tells how different individual members within the ensemble are for some test pattern (data point), i.e. emphasises diversity. It means that if this term is somehow made large, the left hand side will become small, i.e. the error estimate of the ensemble for a given data point will decrease. In other words, if we have more diversity in the ensemble, the accuracy of the combined predictor

will increase. It can also be seen that if we try to maximise this term i.e. cross some limit, the first term (i.e. $\sum_i w_i (f_i - \Phi)^2$) may also increase, which then would negate the whole effect caused due to the amplification of the ambiguity term. Here, the first term signifies the accuracy of an individual for some test pattern. *The second term signifies that we should have diverse members in the ensemble but there is a limit, crossing which would make the individual predictors less accurate.* This is what is often referred to as the *trade-off between diversity and accuracy* in ensembles. In the classification context, according to Hansen and Salamon [22] and as also mentioned in [9], a necessary and sufficient condition for a majority voting ensemble of classifiers to be more accurate than any of its component classifiers is that the components be both accurate and diverse. So the trade-off applies to both regression and classification problems.

A more general decomposition of the quadratic error function given in [40, 9] for the ensemble is presented next in order to help fully understand the way in which individual predictors affect the ensemble as a whole. This is achieved by decomposing the quadratic error into *bias*, *variance* and *covariance*. Since we are concerned with the error expectation on future/unseen data points and need to be consistent with the bias-variance decomposition literature, we need to view the expected value of the error over all possible choices of data sets on which the predictors could be trained on and over all possible parameter initialisations.

The bias-variance-covariance decomposition will also be used to explain the *sensitivity* of the quadratic error of an ensemble towards the *covariance* (in the output estimates) between the individual networks. We will see that the less correlated (having low covariance) the networks are, the better is the ensemble formed.

Bias-Variance-Covariance Decomposition: Specifically, from equation 3,

$$E\{(f(x) - \Phi(x))^2\} = E\{(f(x) - E\{f(x)\})^2\} + (E\{f(x)\} - \Phi(x))^2. \quad (11)$$

In case of an ensemble, let $f(x)$ be the ensemble output. Considering simple averaging as our combination rule,

$$f(x) = \frac{1}{M} \sum_{i=1}^M f_i(x),$$

where M is the number of networks in the ensemble. For simplicity, we will write

$$f(x) = \frac{1}{M} \sum_i f_i. \quad (12)$$

The variance component of Equation 3 can be further subdivided into 2 parts: variance and covariance. Substituting Equation 12 into the variance component of Equation 3 we have

$$Variance(ensemble) = E\left\{\left(\frac{1}{M} \sum_i f_i - E\left\{\frac{1}{M} \sum_i f_i\right\}\right)^2\right\} \quad (13)$$

$$= \frac{1}{M^2} E \left\{ \sum_i \sum_{j=1, j \neq i}^M (f_i - E\{f_i\})(f_j - E\{f_j\}) \right\} + \frac{1}{M^2} E \left\{ \sum_i (f_i - E\{f_i\})^2 \right\} \quad (14)$$

$$= \frac{1}{M^2} \text{Covariance} + \frac{1}{M^2} \text{Variance}. \quad (15)$$

Refer to [11, 14] for details of the above decomposition. We also use a variant of the *Covariance* term in DIVACE [12, 14]. This variant was initially used in one of the established ensemble learning algorithms called Negative Correlation Learning (NCL) proposed by Liu and Yao [30] as a regularisation term.

The bias-variance-covariance decomposition can now be expressed as:

$$\begin{aligned} E \left\{ \left(\frac{1}{M} \sum_i f_i - \Phi(x) \right)^2 \right\} &= (E\{f(x)\} - \Phi(x))^2 + \\ &\frac{1}{M^2} E \left\{ \sum_i \sum_{j=1, j \neq i}^M (f_i - E\{f_i\})(f_j - E\{f_j\}) \right\} + \\ &\frac{1}{M^2} E \left\{ \sum_i (f_i - E\{f_i\})^2 \right\}. \end{aligned} \quad (16)$$

Equation 16 expresses the fact that the quadratic error of the ensemble depends on bias, variance and also on the relationships between individual members of the ensemble. The *covariance* term here can be said to indicate the *diversity* or *disparity* between the member networks as far as their error estimates are concerned.

Hence, it is believed that the more diverse the individual members an ensemble has, the less correlated they would be, which seems obvious. This suggests that the covariance term should be as low as possible. The lower the covariance term, the less the error correlation amongst the networks, which implies reduced error and better performance at the ensemble level. This is the main reason why *diversity* in neural network ensembles is extremely important. This is also true for any kind of ensemble, be it one having support vector machines as its members, decision trees, or for that matter, a mixture of various types of learning machines. A thorough discussion of diversity with regard to neural network ensembles is covered in [8, 9]. In any case, we can see that an ensemble should be composed of members which are both accurate and diverse (uncorrelated in the errors that they make).

4 Multi-objective Evolution for Ensembles

Multi-objectivity in ensembles, as an area of research, has not been explored extensively yet. According to our knowledge, the idea of designing neural networks within a multi-objective setup was first considered by Kottathra and Attikiouzel [26] where they used a branch and bound method to determine the number of hidden neurons (the second objective being the mean square error)

in feed forward neural networks. Kupinski and Anastasio [28] used the Niche Pareto GA [23] to learn classifiers, taking sensitivity (true negative rate) and specificity (true positive rate) as two objectives to optimise. ROC [33] analysis of the resulting solutions (in the Pareto set) is said to have resulted in better ROC curves when compared with generating curves using the conventional approach. Recently, Abbass [4] proposed an evolutionary multi-objective neural network learning approach where the multi-objective problem formulation essentially involved setting up of two objectives viz. complexity of the network and the training error (quadratic error/mean square error). The network complexity here could mean the number of synaptic weights, number of hidden units or a combination of both. An algorithm called MPANN was proposed which uses Pareto differential evolution [5]. MPANN was later on considered [2, 3] for learning and formation of neural network ensembles, albeit, with a different multi-objective formulation (as opposed to that in [4]). Based on multi-objective evolutionary optimisation, taking inspiration from MPANN [2, 3], and exploiting the main idea of NCL [30] we recently proposed DIVACE [12, 14]. Reformulating the regularisation process as a multi-objective optimisation problem, as is done in DIVACE by reformulating NCL, was also studied in [17, 25]. Apart from classification, multi-objective evolutionary optimisation for constructing ensembles for regression problems, finding trade-offs between regression accuracy and complexity of networks, was exhaustively studied in [24]. In cases where individual networks overfit the training data, regression accuracy was shown to improve.

The main reason for using a multi-objective evolutionary approach to designing ensembles is that multi-objectivity enforces the search/ optimisation process (the search process here being finding neural networks with good overall performance) to yield a set of near optimal solutions instead of just a single solution. Getting a set of solutions essentially means that we get a set of near optimal neural networks. These near optimal neural networks could in turn be used as members of an ensemble. With a population based approach we will, in essence, be generating a set of networks and the underlying multi-objective framework would take care of the selection of a set of near optimal solutions/networks from the population. The whole process of generating a Pareto set of neural networks which could be used as members of an ensemble will be automatic as the whole population would, with the passage of time, move towards the Pareto front. Thus, the idea of using multi-objective evolutionary algorithms for the purpose of designing neural network ensembles is very promising.

The main problem in actually using such an approach is the formulation of the multi-objective optimisation problem such that not only the final ensemble created have accurate members but the members also be uniformly distributed on the Pareto optimal front, i.e. diversity is catered for. Remarkably, for multi-objective optimisation to be effective, the optimisation process should lead to convergence to the Pareto optimal front while at the same time maintaining as diverse a distribution of solutions as possible on the Pareto front [18]. A striking *parallel* between multi-criterion optimisation and the necessity of having diverse enough members within an ensemble is evident. Hence, formulating a

problem properly would surely do justice to both (multi-criterion optimisation and ensembles as disparate yet related computational paradigms).

Thus, incorporating the idea of multi-objective evolutionary optimisation into constructing ensembles is an attractive proposition. Next, we present some recent work on integrating ensemble learning with multi-objective evolutionary optimisation.

4.1 Diverse and Accurate Ensemble Learning Algorithm (DIVACE)

DIVACE [12, 14] takes in ideas from MPANN and NCL algorithms. For the evolutionary process, MPANN was used. Diversity was treated as a separate objective and the negative correlation penalty function from NCL was used to quantify it. The two objectives on which to optimise the performance of the ensemble were accuracy and diversity.

Objective 1 – Accuracy. Given a training set T with N patterns. For each network k in the ensemble,

$$\text{(Minimise) Accuracy}_k = \frac{1}{N} \sum_{i=1}^N (f_k^i - o^i)^2, \quad (17)$$

where o^i is the desired output and f_k^i the posterior probability of the class (classification task) or the observed output (regression task) for one training sample i .

Objective 2 – Diversity. From NCL, the correlation penalty function is used as the second objective on which to optimise the ensemble performance. Let N be the number of training patterns and let there be M members in the ensemble, so for each member k , the following term gives an indication of how different it is from other members.

$$\text{(Minimise) Diversity}_k = \sum_{i=1}^N (f_k^i - f^i) \left[\sum_{j \neq k, j=1}^M (f_j^i - f^i) \right], \quad (18)$$

where f^i is the ensemble output for a training sample i . In the information theoretic sense, mutual information is a measure of the correlation between two random variables. A link between the diversity term used here (equation 18) and mutual information was shown in [31]. Minimisation of mutual information between variables extracted (outputs) by two neural networks can be regarded as a condition to ensure that they are different. It has been shown that negative correlation learning, due to the use of the penalty function, can minimise mutual information amongst ensemble members [31, 43]. Hence the use of this penalty function as the diversity term in DIVACE.

It was mentioned in [12] that DIVACE is in no way limited to the use of one particular term for diversity. Also, different accuracy measures and evolutionary processes could well be used. The idea was to address the diversity-accuracy

trade-off in a multi-objective evolutionary setup and DIVACE was shown to achieve a good trade-off. More details on the algorithm can be found in [12, 14].

Also, using the aforementioned multi-objective formulation as the basis, a multi-level (each level catering for enforcement of diversity in a manner reminiscent of the hierarchy presented in [44]) evolutionary framework for the construction of hybrid ensembles was also developed [13, 15]. A majority of the ideas expressed by researchers in order to construct ensembles having both diverse and accurate members so as to have good generalisation capabilities were incorporated into the framework. The framework represents a generic model from which new hybrid ensemble construction algorithms can be developed. It represents a class of ensemble learning methodologies aimed at enforcing diversity and accuracy explicitly within an evolutionary framework. DIVACE-II (as a successor of DIVACE) can be thought of as being one instance of the framework and tries to find an ensemble composed of feed forward neural networks, radial basis function neural networks and support vector machines. We model all levels in our algorithm, however, due to the computationally intensive nature of evolutionary methods, one should limit the ensemble construction approach to as fewer levels as possible depending on domain knowledge. More on DIVACE-II can be found in [13, 15].

As a word of caution, considering the above problem formulation, it may sometimes be necessary to bias the search towards solutions on the pareto front which are more accurate. Essentially, the search should focus towards the use (selection) and procreation thereof, of the more accurate individuals in the pareto front and try to make these highly accurate individuals less correlated when they fail. Essentially, for hard learning problems, the evolutionary process may lead to the inclusion of a large number of solutions (which may not be very accurate due to the hard nature of the problem) which are very different (achieving the diversity objective) from each other, even including individuals which are highly inaccurate, leading to the search for solutions in less promising regions. We do want a diverse ensemble, but at the same time, we want the members to be accurate enough and only to be different in the errors that they make (which should be kept to a minimum) i.e. members making a lot of errors should not be encouraged. Our work reviewed here does not take this point into account and may suffer loss in performance on hard problems. However, the biased approach mentioned above could alleviate matters and is currently being experimented with.

5 Application of the Algorithms on Benchmark Datasets

Both DIVACE and DIVACE-II were tested on 2 benchmark data sets (Australian credit card assessment dataset and Diabetes dataset), available by anonymous ftp from ice.uci.edu in /pub/machine-learning-databases. We also experimented with the multi-objective formulation for both. Pairwise Failure Crediting (PFC) [14] was recently proposed as a diversity measure which credits individuals in the

ensemble with differences in the failure patterns³, taking each pair of individuals and accruing credits in a manner similar to implicit fitness sharing [16, 20]. We consider results obtained using this and the NCL penalty function term diversity measures with both algorithms.

We compare⁴ DIVACE-II with MPANN (both variants from [3] - we refer to these as MPANN1 and MPANN2 here), DIVACE and EENCL [32] due to the experimental setup⁵ in all these being similar and a direct comparison with other evolutionary ensemble construction methods being difficult [32]. We also compare the algorithms with 21 other learning algorithms from the literature.

Table 1 shows interesting properties of the DIVACE-II (both versions) in that, the mean test accuracy is higher than the mean training accuracy for both datasets, which mainly suggests that (on an average) the generalisation ability of DIVACE-II is good i.e. it does not seem to overfit. MPANN2, DIVACE and EENCL on the other hand have higher mean training accuracies and so it can be said that, although these methods hold promise and do show good signs of generalisation, DIVACE-II performs even better due to its test accuracy being much higher. MPANN1 is the other algorithm having a mean test accuracy greater than its mean training accuracy but here again, the mean test accuracy (and mean training accuracy) is not better than DIVACE-II. DIVACE-II does compare well with previously studied approaches.

Table 1. Confidence intervals with a confidence level of 95% for training and testing of DIVACE-II and other algorithms on both datasets. Results computed using accuracy/correct classification rates obtained from 10 and 12 folds for the Australian and Diabetes datasets respectively.

Algorithm	Training		Testing	
	Australian	Diabetes	Australian	Diabetes
DIVACE-II	.877 ± .005	.771 ± .004	.895 ± .0223	.789 ± .0146
DIVACE-II with PFC	.876 ± .003	.776 ± .003	.889 ± .0159	.785 ± .0222
MPANN1	.854 ± .011	.771 ± .013	.862 ± .0303	.779 ± .0186
MPANN2	.852 ± .009	.755 ± .011	.844 ± .0347	.744 ± .0192
DIVACE	.867 ± .004	.783 ± .003	.857 ± .0303	.766 ± .0322
EENCL	.891 ± .006	.802 ± .004	.857 ± .0241	.764 ± .0237

³ A failure pattern is a string of 0s and 1s indicating success or failure of the learning machine on the training instances in the original training set.

⁴ The comparison is mainly with respect to the majority voting combination rule as DIVACE-II only employs this decision fusion strategy

⁵ n-fold cross validation used here. $n = 10$ for Australian and $n = 12$ for Diabetes dataset. Learning rate for NNs is not the same as that used in [3, 32] for DIVACE-II as the evolutionary process is inherently very different and we use methodologically different learners. Moreover, we evolve the population for 50 generations as opposed to 200 in [3, 32].

Tables 2 and 3 put both DIVACE and DIVACE-II against 24 previously studied learning algorithms. All approaches are not evolutionary and this has been done mainly due to the lack of results in the field. The tables shows the average test error rates ⁶ (lower means better) for many learning algorithms.

The algorithms used for comparison can be categorised into five classes: evolutionary ensemble methods (MPANN [3], EENCL [32]), statistical methods (Discrim, Quadisc, Logdisc, SMART, ALLOC80, k-NN, CASTLE, NaiveBay), decision trees based methods (CART, IndCART, NewID, AC^2 , Baytree, Cal5, C4.5), rule based methods (CN2, ITrule) and neural network based methods (BP, LVQ, RBF, DIPOL92). Details of the algorithms in the latter four classes can be found in [34]. The error rates refer to the percentage of wrong classifications on the test set.

As is evident from the tables, DIVACE-II has been able to achieve a good generalisation performance (due to it having lower average test error rates on both datasets). It is generally better than most of the algorithms shown for both datasets which shows the promise of not only DIVACE-II but also the evolutionary hybrid ensemble construction framework from which it is derived. Also, DIVACE compares well with most of the algorithms and is better than a majority of them.

Table 2. Comparison of DIVACE-II with other learning algorithms from [34] in terms of the average test error rates for the Australian credit card assessment dataset. Results are averaged on 10-fold cross validation.

Algorithm	Error Rate	Algorithm	Error Rate
DIVACE-II	0.105, 0.111	IndCART	0.152
DIVACE	0.138, 0.125	NewID	0.181
MPANN1	0.135	AC^2	0.181
MPANN2	0.156	Baytree	0.171
EENCL	0.135	NaiveBay	0.151
Discrim	0.141	CN2	0.204
Quadisc	0.207	C4.5	0.155
Logdisc	0.141	ITrule	0.137
SMART	0.158	Cal5	0.131
ALLOC80	0.201	DIPOL92	0.141
k-NN	0.181	BP	0.154
CASTLE	0.148	RBF	0.145
CART	0.145	LVQ	0.197

In general, it can be said that both DIVACE and DIVACE-II certainly do find a good trade-off between diversity and accuracy.

⁶ The two rates for both DIVACE and DIVACE-II are from the use of different diversity measures (original formulation and PFC respectively).

Table 3. Comparison of DIVACE-II with other learning algorithms from [34] in terms of the average test error rates for the Diabetes dataset. Results are averaged on 12-fold cross validation.

Algorithm	Error Rate	Algorithm	Error Rate
DIVACE-II	0.211, 0.215	IndCART	0.271
DIVACE	0.227, 0.226	NewID	0.289
MPANN1	0.221	AC^2	0.276
MPANN2	0.254	Baytree	0.271
EENCL	0.221	NaiveBay	0.262
Discrim	0.225	CN2	0.289
Quadisc	0.262	C4.5	0.27
Logdisc	0.223	ITrule	0.245
SMART	0.232	Cal5	0.25
ALLOC80	0.301	DIPOL92	0.224
k-NN	0.324	BP	0.248
CASTLE	0.258	RBF	0.243
CART	0.255	LVQ	0.272

6 Conclusion

We consider the general problem of learning (in the regression and classification context) together with its relationship with generalisation. Theoretical aspects of an ensemble of learners are discussed in order to support the importance of having both diversity and accuracy and explaining the trade-off between them in the construction of such aggregate learners, leading to effective generalisation. Multi-objective evolutionary algorithms are presented as being solutions to constructing ensembles and achieving a good trade-off between diversity and accuracy. Some recent work (learning algorithms viz. DIVACE and DIVACE-II together with the framework for the construction of diverse hybrid ensembles) along these lines is reviewed.

DIVACE tackles the ensemble learning problem within a multi-objective evolutionary setup. The explicit use of diversity and accuracy in the formulation of the problem tries to make the evolutionary process search for a good trade-off by continuously moving towards the Pareto front. Bringing together diversity enforcement mechanisms with DIVACE at the backdrop results in this multi-level ensemble learning framework (and the algorithm DIVACE-II) where individual predictors trained using disparate learning methodologies are generated automatically by successively competing and co-operating with each other. The trade-off between diversity and accuracy is addressed at various levels and leads to better generalisation as demonstrated by results on the application of the approach to benchmark datasets.

We also note that for harder learning problems, it may be necessary to bias the search direction towards more accurate individuals in the Pareto front, leading to highly accurate (yet uncorrelated when it comes to failure) learners and them being used as an ensemble. This idea is currently under investigation.

References

1. H. A. Abbass. A memetic pareto evolutionary approach to artificial neural networks. In *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence*, pages 1–12, Berlin, 2000. Springer-Verlag.
2. H. A. Abbass. Pareto neuro-ensemble. In *16th Australian Joint Conference on Artificial Intelligence*, pages 554–566, Perth, Australia, 2003. Springer.
3. H. A. Abbass. Pareto neuro-evolution: Constructing ensemble of neural networks using multi-objective optimization. In *The IEEE 2003 Conference on Evolutionary Computation*, volume 3, pages 2074–2080. IEEE Press, 2003.
4. H. A. Abbass. Speeding up backpropagation using multiobjective evolutionary algorithms. *Neural Computation*, 15(11):2705–2726, November 2003.
5. H. A. Abbass, R. Sarker, and C. Newton. Pde: A pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2001)*, volume 2, pages 971–978. IEEE Press, 2001.
6. C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
7. A. Blum and R. L. Rivest. Training a 3-node neural network is NP-complete. In *Machine Learning: From Theory to Applications*, pages 9–28, 1993.
8. G. Brown. *Diversity in Neural Network Ensembles*. PhD thesis, School of Computer Science, University of Birmingham, 2004.
9. G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: A survey and categorisation. *Journal of Information Fusion (Special issue on Diversity in Multiple Classifier Systems)*, 6:5–20, March 2005.
10. A. Chandra. Evolutionary approach to tackling the trade-off between diversity and accuracy in neural network ensembles. Technical report, School of Computer Science, The University of Birmingham, UK, April 2004.
11. A. Chandra, H. Chen, and X. Yao. Trade-off between diversity and accuracy in ensemble generation. In Y. Jin, editor, *Multi-objective Machine Learning*, Computational Intelligence. Springer, 2006.
12. A. Chandra and X. Yao. DIVACE: Diverse and Accurate Ensemble Learning Algorithm. In *Proc. 5th Intl. Conference on Intelligent Data Engineering and Automated Learning (LNCS 3177)*, pages 619–625, Exeter, UK, August 2004. Springer-Verlag.
13. A. Chandra and X. Yao. Evolutionary framework for the construction of diverse hybrid ensembles. In M. Verleysen, editor, *Proc. 13th European Symposium on Artificial Neural Networks*, pages 253–258, Brugge, Belgium, April 2005. d-side.
14. A. Chandra and X. Yao. Ensemble learning using multi-objective evolutionary algorithms. *Journal of Mathematical Modelling and Algorithms (to appear)*, 2006.
15. A. Chandra and X. Yao. Evolving hybrid ensembles of learning machines for better generalisation. *Neurocomputing*, 69(7-9):686–700, March 2006.
16. P. J. Darwen and X. Yao. Every niching method has its niche: Fitness sharing and implicit sharing compared. In *Proc. of the 4th International Conference on Parallel Problem Solving from Nature (PPSN-IV)*, (LNCS-1141), pages 398–407, Berlin, September 1996. Springer-Verlag.
17. R. de Albuquerque Teixeira, A. P. Braga, R. H. Takahashi, and R. R. Saldanha. Improving generalization of mlps with multi-objective optimization. *Neurocomputing*, 35:189–194, 2000.

18. K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK : Wiley, 2001.
19. T. G. Dietterich. Machine-learning research: Four current directions. *The AI Magazine*, 18(4):97–136, 1998.
20. S. Forrest, R. E. Smith, B. Javornik, and A. S. Perelson. Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation*, 1(3):191–211, 1993.
21. S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–5, 1992.
22. L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
23. J. Horn and N. Nafpliotis. Multiobjective Optimization using the Niche Pareto Genetic Algorithm. Technical Report IlliGAI Report 93005, University of Illinois, Urbana-Champaign, July 1993.
24. Y. Jin, T. Okabe, and B. Sendhoff. *Applications of Evolutionary Multi-objective Optimization (Advances in Natural Computation)*, volume 1, chapter Evolutionary multi-objective approach to constructing neural network ensembles for regression, pages 653–672. World Scientific, 2004.
25. Y. Jin, T. Okabe, and B. Sendhoff. Neural Network Regularization and Ensembling Using Multi-objective Evolutionary Algorithms. In *2004 Congress on Evolutionary Computation (CEC'2004)*, volume 1, pages 1–8, Portland, Oregon, USA, June 2004. IEEE Service Center.
26. K. Kottathra and Y. Attikiouzel. A novel multicriteria optimization algorithm for the structure determination of multilayer feedforward neural networks. *Journal of Network and Computer Applications*, 19:135–147, 1996.
27. A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. *Neural Information Processing Systems*, 7:231–238, 1995.
28. M. A. Kupinski and M. A. Anastasio. Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characteristic curves. *IEEE Transactions on Medical Imaging*, 18(8):675–685, August 1999.
29. B. Littlewood and D. R. Miller. Conceptual modeling of coincident failures in multiversion software. *IEEE Transactions on Software Engineering*, 15(12):1596–1614, December 1989.
30. Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404, 1999.
31. Y. Liu and X. Yao. Learning and evolution by minimization of mutual information. In J. J. M. Guervós, P. Adamidis, H.-G. Beyer, J.-L. Fernández-Villacañas, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature VII (PPSN-2002)*, volume 2439 of *LNCS*, pages 495–504, Granada, Spain, 2002. Springer Verlag.
32. Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380, November 2000.
33. C. E. Metz. Basic principles of roc analysis. *Seminars in Neuclear Medicine*, 8(4):283–298, 1978.
34. D. Michie, D. Spiegelhalter, and C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Limited, 1994.
35. D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.
36. D. W. Opitz and J. W. Shavlik. Generating accurate and diverse members of a neural-network ensemble. *Neural Information Processing Systems*, 8:535–541, 1996.

37. T. Schnier and X. Yao. Using negative correlation to evolve fault-tolerant circuits. In *Proceedings of the 5th International Conference on Evolvable Systems: From Biology to Hardware (ICES'2003)*, pages 35–46. Springer-Verlag. Lecture Notes in Computer Science, Vol. 2606, March 2003.
38. A. Sharkey and N. Sharkey. Combining diverse neural networks. *The Knowledge Engineering Review*, 12(3):231–247, 1997.
39. K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341–348, February 1996.
40. N. Ueda and R. Nakano. Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks*, pages 90–95, 1996.
41. G. Valentini and F. Masulli. Ensembles of learning machines. In R. Tagliaferri and M. Marinaro, editors, *Neural Nets WIRN Vietri-2002 (LNCS 2486)*, pages 3–19. Springer-Verlag, June 2002.
42. W. Wang, P. Jones, and D. Partridge. Diversity between neural networks and decision trees for building multiple classifier systems. In *Proc. Int. Workshop on Multiple Classifier Systems (LNCS 1857)*, pages 240–249, Cagliari, Italy, June 2000. Springer.
43. X. Yao and Y. Liu. Evolving neural network ensembles by minimization of mutual information. *International Journal of Hybrid Intelligent Systems*, 1(1), January 2004.
44. W. Yates and D. Partridge. Use of methodological diversity to improve neural network generalization. *Neural Computing and Applications*, 4(2):114–128, 1996.