DIVACE: Diverse and Accurate Ensemble Learning Algorithm

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

Abstract. In order for a neural network ensemble to generalise properly, two factors are considered vital. One is the diversity and the other is the accuracy of the networks that comprise the ensemble. There exists a tradeoff as to what should be the optimal measures of diversity and accuracy. The aim of this paper is to address this issue. We propose the DIVACE algorithm which tries to produce an ensemble as it searches for the optimum point on the diversity-accuracy curve. The DIVACE algorithm formulates the ensemble learning problem as a multi-objective problem explicitly.

1 Introduction

A key issue in neural computation is that of generalisation. Multi-layer perceptrons have been established as good neural computation models in addition to the fact that numerous techniques have been developed in order for such networks to be trained effectively resulting in better generalisation. A problem that perpetuates and haunts neural computation researchers and is akin to the problem of generalisation ability in neural networks is called the 'bias-variance' dilemma. Geman et al. [6] explained this dilemma very well.

One solution to tackling this dilemma is the use of a collection of predictors instead of one. Brown et al. [5] describe the notion of an ensemble of predictors, and show that such an architecture can be applied to any classification/regression problem. Ensembles have been shown to perform better than their members [7, 8].

Given that ensmebles work better when compared with a single predictor, there have been many inroads into improving the prediction ability of such aggregate systems in recent years. Liu and Yao [8] proposed the Negative Correlation Learning (NCL) algorithm wherein a penalty term is added to the error function which helps in making the individual predictors as different from each other as possible while encouraging the accuracy of individual predictors. This enables the mapping function learnt by the ensemble to generalise better when an unseen input is to be processed. As far as learning and ensemble creation are concerned there are techniques which invovle some manual interference/control as opposed to techniques which are inherently automatic [7]. An evolutionary approach to learning and creation of ensembles automatically was proposed in [7].

Abbass [4] proposed the Pareto-frontier Differential Evolution (PDE) method, which is an extension of the Differential Evolution (DE) algorithm proposed by Storn and Price [11]. In [1], an algorithm for ensemble learning called Memetic Pareto Artificial Neural Network (MPANN), which is a customised version of PDE for evolving neural networks, was proposed.

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

Much of this paper deals with a formal description of our approach which is inspired by the MPANN and NCL algorithms but has a few differences. The idea was to evolve an ensemble such that the evolutionary process automatically takes care of the individual members being diverse and at the same time accurate enough in order that the final ensemble generalises well.

2 Diversity and Accuracy in Ensembles

This section gives a brief account of the notion of diversity and that of accuracy in the ensemble context. Then, certain aspects of MPANN and NCL algorithms, which are the main source of inspiration for our approach, are discussed.

Diversity. Brown et al. [5] gives a good account of why diversity is necessary in neural network ensembles and presents a taxonomy of methods that enforce it in practice. If two neural networks make different errors on the same data points/inputs, they are said to be diverse [5].

Accuracy. Accuracy could be defined as the degree of a network (ensemble member) performing better than random guessing on a new input [5].

The Trade-off. If we train a network on the same dataset more than a certain number of times, its generalisation ability degrades. Since there is no proven equivalence between generalisation and training error, a network with minimum error on the training set may not have best generalisation [12]. Also, since a population is bound to have at least as much information as any single individual [12] it is always beneficial to make use of all these networks instead of just one. In addition, the more accurate the networks are, the more similar they are likely to be. We need ensembles for better generalisation and given the fact that similar members would preclude the need for ensembles, members have to be diverse. The more diverse the members are, the more well spread will their outputs be around the target value resulting in the expected/mean value of the member outputs being closer to this target value. The fact that the distribution of outputs should be around the target value necessiates accuracy. Hence, in order for an ensemble system to generalise well, the member networks, apart

from being diverse, should also be accurate. More precisely, the member outputs should be *compactly well distributed around the desired value*. Herein lies the trade-off between diversity and accuracy under the ensemble setup. There has to be an optimum point on the diversity-accuracy curve where the networks are as diverse as they are accurate i.e. a point at which if the networks become more accurate, they would not be as diverse as they ought to be and vice-versa. The gist here is that since all networks in the ensemble try to provide a solution for the same task, so in order for them to work well, they should locally complement each other. The essence of our approach lies in the fact that it tries to figure out this near optimal trade-off in a multi-objective evolutionary setup.

2.1 Inspiration from MPANN and NCL

MPANN was proposed and successfully tested by Abbass [1] as a PDE approach to evolving neural networks. Our approach incorporates the evolutionary process from the MPANN algorithm in that the control structure of our approach remains similar to that of MPANN, but the way we formulate the problem at hand (the multi-objective problem) and the manner in which offspring are generated in this evolutionary process are different.

As far as NCL [8] is concerned, we use the notion of negative correlation in the formulation of our multi-objective problem. The penalty function used in NCL has an information theoritic aspect to it which we have tried to exploit in our approach by utilising this function for defining diversity.

Changes in MPANN for our approach. In MPANN [1] and in PDE [4], the use of a Gaussian distribution for crossover generates well-spread children around the main parent and along the directions of the supporting parents. With our approach, we tried to make the offspring generation process somewhat adaptive in the sense that, at the beginning of the evolutionary process, the children generated are more widely spread around the main parent. As the evolutionary process proceeds further, this spread is reduced such that in the end the offspring are generated in a manner similar to what happens with PDE i.e. using a Gaussian distribution with mean 0 and variance 1. In essence, we start with a Gaussian with mean 0 and large variance σ^2, where σ^2 was set as 2 and then we anneal this value such that initially the search mechanisim is explorative but with time as the variance decreases, the search process becomes more exploitative. Gaussian mutation is also incorporated in the search process. The other difference in our approach is in the formulation of the multi-objective problem which will be considered in the following section in greater detail and which proves a constructive confluence of NCL and MPANN.

3 DIVACE: Algorithm Formulation

Our approach, called DIVACE (DIVerse and ACcurate Ensemble learning), takes in ideas from MPANN and NCL algorithms. For the evolutionary process, we use

MPANN, and for diversity, we use the negative correlation penalty function of NCL as one of our objectives for the multi-objective problem. The two objectives on which to optimise the performance of the ensemble are accuracy and diversity.

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

$$\textbf{(Minimise)}\ \text{Accuracy}_k = \frac{1}{N}\sum_{i=1}^{N}\left(f_k^i - o^i\right)^2, \tag{1}$$

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.

$$\textbf{(Minimise)}\ \text{Diversity}_k = \sum_{i=1}^{N}\left(f_k^i - f^i\right)\left[\sum_{j\neq k, j=1}^{M}\left(f_j^i - f^i\right)\right], \tag{2}$$

where f^i is the ensemble output for a training sample i. In the information theoritic sense, mutual information is a measure of the correlation between two random variables. A link between the diversity term used here (equation 2) and mutual information was shown in [9]. 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 [9, 12]. Hence the use of this penalty function as the diversity term in DIVACE.

It should be noted here 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 is to address the diversity-accuracy trade-off in a multiobjective evolutionary setup.

DIVACE. Following is the DIVACE algorithm:

Step 1: Create a random initial population[1] (size M) of networks, the weights for each are uniformly distributed random values in $U(0, 1)$.
Step 2: Apply BP to all individuals in the population.
Step 3: Repeat until termination conditions (a certain number of generations in our case) are met.

[1] For training, we take all the networks in the population as our ensemble but for testing, we only use the final pareto set as the ensemble.

a) Evaluate the individuals in accordance with the two objective functions (Accuracy/quadratic error and diversity/penalty function of NCL[8]) and label the non-dominated set – Non-dominated sorting procedure by Srinivas and Deb [10] used here.
b) If the number of non-dominated individuals is less than 3 then a repair rule similar to that used in MPANN [3] is used.
c) All dominated solutions are deleted from the population.
d) Repeat until population size is M
   – Variance update: updating the variance value for the Gaussian distribution used in crossover. We do it according to

$$\sigma^2 = 2 - \left(\frac{1}{1 + e^{(\text{anneal\_time} - \text{generation})}} \right), \tag{3}$$

   where anneal_time is a parameter signifying exploration time/number of generations for which the search process is to be explorative. In our experiments, we use a value of 50 for the anneal_time parameter.
   – Select 3 parents at random from the population. Let α_1 be the main parent and α_2 and α_3 be the supporting parents.
   – Perform crossover: Produce a child which has an architecture which is similar to the parents but weights given by,

$$w_{hi} = w_{hi}^{\alpha_1} + N\left(0, \sigma^2\right)\left(w_{hi}^{\alpha_2} - w_{hi}^{\alpha_3}\right) \tag{4}$$

$$w_{oh} = w_{oh}^{\alpha_1} + N\left(0, \sigma^2\right)\left(w_{oh}^{\alpha_2} - w_{oh}^{\alpha_3}\right) \tag{5}$$

   – Perform mutation: Mutate the child with probability $1/|population|$ according to,

$$w_{hi} = w_{hi} + N\left(0, 0.1\right) \tag{6}$$

$$w_{oh} = w_{oh} + N\left(0, 0.1\right) \tag{7}$$

   – Apply BP to the child and add it to the population.

4 Results

This section presents some results obtained on testing DIVACE on 2 data sets (Australian credit card assessment dataset and Diabetes dataset), available by anonymous ftp from ice.uci.edu in /pub/machine-learning-databases. The experimental setup is similar to that in [2, 3, 7]. Table 1 shows the performance accuracy (accuracy rates/percentage accuracy) of the formed ensemble on the Australian credit card assessment dataset as well as the Diabetes dataset. During the course of the evolutionary process, it was expected that each member in the Pareto front (after every generation) would perform well on different parts of the training set. Since the results we get are quite comparable with previous results and as the multi-objective problem is formulated to enforce diversity, we can say that DIVACE performed verly well in finding an appropriate trade-off between accuracy and diversity among members.

Table 1. Performance (accuracy rates) of the ensemble formed using DIVACE on the Australian credit card assessment and Diabetes datasets.

Australian credit card assessment dataset						
	Simple Averaging		Majority Voting		Winner-takes-all	
	Training	Testing	Training	Testing	Training	Testing
Mean	0.872	0.862	0.867	0.857	0.855	0.849
SD	0.007	0.049	0.007	0.049	0.007	0.053
Max	0.884	0.927	0.879	0.927	0.864	0.927
Min	0.859	0.753	0.856	0.768	0.842	0.753
Diabetes dataset						
	Simple Averaging		Majority Voting		Winner-takes-all	
	Training	Testing	Training	Testing	Training	Testing
Mean	0.780	0.773	0.783	0.766	0.766	0.766
SD	0.006	0.050	0.005	0.057	0.017	0.049
Max	0.791	0.859	0.791	0.875	0.796	0.843
Min	0.768	0.687	0.772	0.671	0.730	0.671

5 Conclusion

In this paper, the problem of creating a diverse and accurate set of networks for an ensemble was discussed. The DIVACE algorithm performs very well on the training front, which is expected as we take into account the whole training set with consistent class distributions for each network. The noteworthy aspect here is that our algorithm produces competitive results (table 2) on the testing front as well when compared with the second formulation of the MPANN algorithm [2] which has a similar training setup. The whole idea behind this paper mainly was to present an ensemble learning technique which combines good ideas from both MPANN and NCL, as a result, addressing the trade-off between diversity and accuracy within an evolutionary multi-objective framework. The new algorithm, DIVACE, can produce accurate and diverse ensembles automatically using the multi-objective evolutionary approach.

Table 2. Comparison of DIVACE with the second formulation of MPANN in [2]. Shown in the table are the accuracy rates (on the test set) for both the data sets using simple average, majority vote and winner-takes-all strategies respectively.

Algorithm	Australian	Diabetes
DIVACE	0.862,0.857,0.849	0.773,0.766,0.766
MPANN [2]	0.844,0.844,0.824	0.744,0.744,0.746

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-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.
3. H. A. Abbass. Pareto neuro-ensemble. In *16th Australian Joint Conference on Artificial Intelligence*, pages 554–566, Perth, Australia, 2003a. Springer.
4. 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.
5. Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. Diversity creation methods: A survey and categorisation. *Journal of Information Fusion (to appear)*, 2004.
6. S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.
7. Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380–387, November 2000.
8. Yong Liu and Xin Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404, 1999.
9. Yong Liu and Xin Yao. Learning and evolution by minimization of mutual information. In J. J. Merelo 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.
10. N. Srinivas and K. Deb. Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
11. R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, USA, 1995.
12. Xin Yao and Yong Liu. Evolving neural network ensembles by minimization of mutual information. *International Journal of Hybrid Intelligent Systems*, 1(1), January 2004.