ESANN'2005 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 27-29 April 2005, d-side publi., ISBN 2-930307-05-6.

Evolutionary Framework for the Construction of Diverse Hybrid Ensembles

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, UK

Abstract. Enforcing diversity explicitly in ensembles while at the same time making individual predictors accurate as well has been shown to be promising. This idea was recently taken into account in the algorithm DIVACE. There have been a multitude of theories on how one can enforce diversity within a combined predictor setup. This paper aims to bring these theories together in an attempt to synthesise a framework that can be used to engender new evolutionary ensemble learning algorithms. The framework treats diversity and accuracy as evolutionary pressures that can be exerted at multiple levels of abstraction and is shown to be effective.

1 Introduction

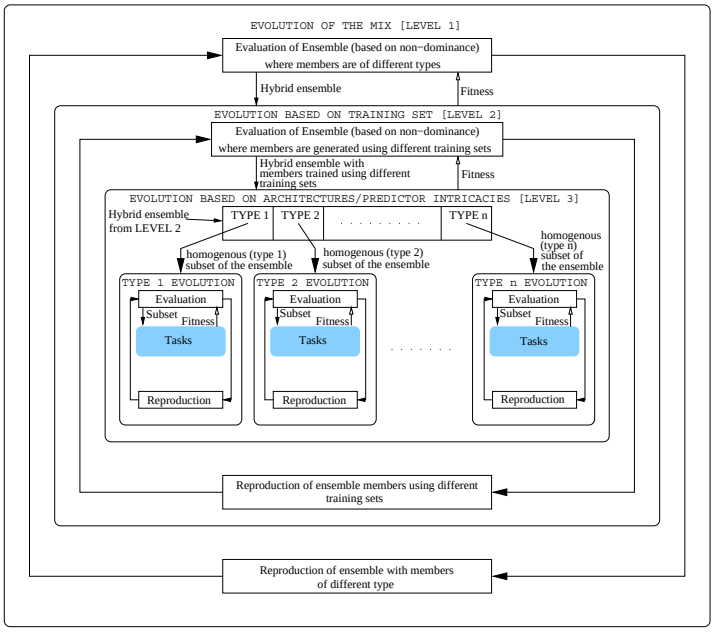
There have been many studies into developing holistic classification schemes for ensemble methods. We are mainly concerned with ways in which diversity can been enforced in various ensemble learning algorithms. A very appropriate classification scheme was recently presented by Brown et al. [3]. According to them [3] this scheme encapsulates a majority of the proposed ensemble methods. We are essentially concerned with this classification scheme as it includes, according to our knowledge, more ensemble methods in it than any other classification scheme, which is one reason we have tried to develop an ensemble constructing framework that revolves around it as will be seen shortly.

What we wanted in our framework was to make it flexible enough such that while creating an ensemble, diversity could be enforced in as many ways as possible (exploiting the classification scheme of Brown et al. [3]). Moreover, Yates and Partridge [7] came up with a scheme which puts diversity generating methods, as alluded to above, into various levels depending on the efficacy of each in enforcing diversity within an ensemble. According to them, methodological diversity is the most effective diversity generating method. This is followed by training set structure, architecture of the learning machine and initial conditions for the base learners in this particular order. Consolidating these ideas into an evolutionary scheme leads us to propose a hierarchical framework which can be used for synthesising new ensemble learning algorithms. It should be noted here that DIVACE [4], with its explicit treatment of diversity and accuracy, forms a embryonic part of this framework and can be said to be one of the motivations behind proposing it. As will be seen, the framework models a generic scheme from which new ensemble learning algorithms can be instantiated. We present

this framework together with an algorithm or instance resulting from it, followed by some empirical results to validate its promise shortly.

2 A Proposed Merger

A possible evolutionary framework for the construction of diverse hybrid ensembles can be described by Figure 1. As can be seen, there are three levels of evolution present. First is the evolution of the mix i.e. evolving the mixture of the various types of predictors. Second, we consider evolution of the ensemble based on the structure of the training set (given the mix). A process similar to the original DIVACE forms the third and final evolutionary level.



NOTE: Fitness consists of 2 values for each member in the ensemble/subset viz. Diversity and Accuracy

Fig. 1: The proposed framework.

The framework shows that the hybrid ensemble at Level 2 will have subsets of different types of predictors. These subsets can, in themselves, be considered as homogeneous ensembles and evolved in accordance with DIVACE, keeping the other subsets fixed. Level 3 can be called as the DIVACE stage where reproduction depends on the evolutionary factor(s) chosen for a given predictor type. The factors could be architectures or weights in case of NNs, kernel function in case of SVMs, architectures of RBFNs etc.

Level 3, due to its subset evolution process, *enforces competition* between the various subsets. This competition makes the framework as a whole model a co-evolutionary learning strategy where subspecies (subsets) compete with each

other to stay in the ensemble. Additionally, these very species cooperate at the immediate higher level (Level 2) and compete again at Level 1. *The framework therefore embodies both competitive and cooperative co-evolution within a multi-objective and multi-level evolutionary setup.* The choice on the placement of these levels essentially depends on the prior knowledge available. However, the above mentioned co-evolutionary theme would be most effective if we keep this very ordering in the levels or at least let the innermost level stay where it is as it makes more sense to have a hybrid set of predictors and then let the subsets compete with each other than to enforce competition without having a mix.

3 An Instance of the Framework: DIVACE-II

Here we presents our algorithm i.e. DIVACE-II which can be thought of as being one instance of the framework. In DIVACE-II, we try to incorporate all the levels mentioned in the framework presented in the previous section. However, one should limit the ensemble construction approach to as fewer levels as possible depending on domain knowledge due to the computationally intensive nature of evolutionary methods. We model all three levels in our algorithm mainly to test the effectiveness of the framework. We use a technique similar to AdaBoost [2] while initialising the predictor population as well as generating offspring. The idea is to generate a new training set, $1/4^{th}$ of which is generated in a manner similar to AdaBoost and the rest of the instances are chosen randomly from the original training set. Lets call this as the *train_generate* procedure.Following is the DIVACE-II algorithm:

Step I: Initialise the population of predictors [1] using Bagging [2] and the variant of AdaBoost [2] discussed above.
Step II: Perform k-means clustering [5] using the Euclidean distance (with respect to the failure patterns [2] of the predictors on the original training set) to form M clusters and select the best [3] predictor from each cluster to form the initial ensemble.
Step III: Repeat until termination conditions (a certain number of generations in our case) are met.

1. Preserve the elite: archive the current ensemble if it dominates the previous best ensemble (based on training and test accuracies).

2. Evaluate the individuals in accordance with the two objective functions (accuracy and diversity) [4] and label the non-dominated set as in [4].

[1] Population contains p number of NNs, p number of SVMs and p number of RBFNs, where $p = 20$.

[2] 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.

[3] Best/worst individual/predictor wherever mentioned is in terms of accuracy on the original training set

[4] Multi-objective formulation similar to that in DIVACE [4] where accuracy was formulated as the mean squared error and diversity as the correlation penalty function in [6].

3. All non-dominated individuals are selected as parents. Misclassified training examples have their probability values increased as in AdaBoost and then *train_generate* applied.

4. If non-dominated individuals = total number of individuals in the current ensemble then goto 5.

   - Generate an offspring pool [5] using the training set generated in the previous step and applying *train_generate* for all the types of predictors being used.
   - Cluster the offspring pool using k-means clustering where the number of clusters is decided by the number of dominated individuals.
   - Replace the individuals that are dominated with the best individual from each cluster while making sure that only the individuals which are common with respect to types are replaced. This replacement strategy ensures equal representation of each type of predictor.
   - Goto 6.

5. Generate an offspring pool [6] using the training set generated in 3 and applying *train_generate*, the size of which is equal to the number of types of predictors being used. Replace the worst individual (in the population and one which is not in the ensemble) of the same type (as the best individual in this pool) with the best from this pool.

6. Re-cluster the population with the number of clusters equal to the size of the ensemble. This is done to ensure that there is only 1 member from each cluster present in the ensemble at all times. The best member is selected to be included in the new ensemble.

7. *(Optional)* Level 3 evolution. Perform DIVACE for the various subsets in the new ensemble which subsequently gives rise to a newer ensemble.

Step IV: Use the archived ensemble as the final hybrid ensemble.

4 Results and Comparison

DIVACE-II was 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 compare it with MPANN (both variants from [1] - we refer to these as MPANN1 and MPANN2 here), DIVACE [4] and EENCL [6] due to the experimental setup [7] in all these being similar.

[5] Pool contains q number of NNs, q number of SVMs and q number of RBFNs, where $q = 15$.
[6] Pool contains 1 NN, 1 SVM and 1 RBFN.
[7] 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 [1, 4, 6] 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 [1, 4, 6].

Table 1 shows interesting properties of the algorithm 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.

Taking the case of training for both datasets (from Table 1), as compared to ± 0.005 for DIVACE-II, other approaches show slightly more variable characteristics in having confidence intervals of ± 0.011, ± 0.009, ± 0.004 and ± 0.006 respectively. On an average (averaging out the confidence intervals established to illustrate the difference and calling the result as 'average variability'), the four approaches considered for comparison have confidence intervals of the order ± 0.0075 whereas for DIVACE-II this is ± 0.005. A similar situation can be seen for the Diabetes dataset where we have ± 0.004 for DIVACE-II as opposed to ± 0.007 for others. So, we can say that DIVACE-II is less variable i.e. performs well on the training front. On the testing front, for the Australian credit dataset, the confidence intervals established for DIVACE-II can be given by ± 0.0223 whereas these are ± 0.0298 for other approaches. The latter is higher, signifying more variability on an average. Same is true for the Diabetes dataset where the interval established by DIVACE-II is ± 0.0146 as opposed to ± 0.0234 for others. Generally speaking, DIVACE-II does compare well with previously studied approaches. Also, the stability (low values for standard deviation) of DIVACE-II over multiple repetitions of cross validation is depicted in Table 2.

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 rates obtained from 10 and 12 folds for the Australian and Diabetes datasets respectively.

	Training		Testing	
Algorithm	Australian	Diabetes	Australian	Diabetes
DIVACE-II	$.877 \pm .005$	$.771 \pm .004$	$.895 \pm .0223$	$.789 \pm .0146$
MPANN1	$.854 \pm .011$	$.771 \pm .013$	$.862 \pm .0303$	$.779 \pm .0186$
MPANN2	$.852 \pm .009$	$.755 \pm .011$	$.844 \pm .0347$	$.744 \pm .0192$
DIVACE	$.867 \pm .004$	$.783 \pm .003$	$.857 \pm .0303$	$.766 \pm .0322$
EENCL	$.891 \pm .006$	$.802 \pm .004$	$.857 \pm .0241$	$.764 \pm .0237$
average variability	$\pm .0075$	$\pm .007$	$\pm .0298$	$\pm .0234$

Table 2: Average performance (training and testing accuracy rates) of DIVACE-II on both datasets. Results averaged on 10 cross validation repetitions.

	Australian		Diabetes	
	Training	Testing	Training	Testing
Mean (SD)	0.875 (0.003)	0.897 (0.005)	0.768 (0.004)	0.781 (0.009)

5 Conclusion

The main idea behind pursuing this research was to come up with a generic ensemble construction model that could be used to generate new ensemble learning algorithms. Bringing together diversity enforcement mechanisms with DIVACE at the backdrop essentially results in an evolutionary framework that rolls these diversity enforcement ideas into one multi-level ensemble learning strategy where individual predictors are generated automatically by successively competing and co-operating with each other. An algorithm resulting from the framework was presented as well to prove its effectiveness/validity. DIVACE-II is generally seen to outperform most of the algorithms it is compared with rendering the framework valid. This establishes our idea of enforcing diversity at multiple levels (which is modelled by our framework) as being plausible. To conclude, the framework proposed here looks promising but much work still remains to be done in order to establish it as/or come up with a truly generic model for ensemble construction from which new ensemble learning algorithms can be synthesised.

References

[1] 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.

[2] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1,2), 1999.

[3] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: A survey and categorisation. *Journal of Information Fusion*, 6(1):5–20, 2005.

[4] 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.

[5] V. Faber. Clustering and the continuous k-means algorithm. In *Los Alamos Science: High Performance Computing*, volume 22, pages 138–144. Los Alamos National Laboratory, 1994.

[6] Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE-EC*, 4(4):380, November 2000.

[7] W. Yates and D. Partridge. Use of methodological diversity to improve neural network generalization. *Neural Computing and Applications*, 4(2):114–128, 1996.