

Evolutionary Random Neural Ensembles Based on Negative Correlation Learning

Huanhuan Chen and Xin Yao

Abstract—This paper proposes to incorporate bootstrap of data, random feature subspace and evolutionary algorithm with negative correlation learning to automatically design accurate and diverse ensembles. The algorithm utilizes both bootstrap of training data and random feature subspace techniques to generate an initial and diverse ensemble and evolves the ensemble with negative correlation learning. The idea of generating ensemble by simultaneous randomization of data and feature is to promote the diversity within the ensemble and encourage different individual NNs in the ensemble to learn different parts or aspects of the training data so that the ensemble can learn better the entire training data. Evolving the ensemble with negative correlation learning emphasizes not only the accuracy of individual NNs but also the cooperation among different individual NNs and thus improves the generalization. As a byproduct of bootstrap, out-of-bag (OOB) estimation, which can estimate the generalization performance without any extra data points, serves another benefit of this algorithm. The proposed algorithm is evaluated by several benchmark problems and in these cases the performance of our algorithm is better than the performance of other ensemble algorithms.

I. INTRODUCTION

Ensemble of multiple learning machines, i.e. a group of learners that work together as a committee, has received a lot of research interests in the machine learning community because it is thought as a good approach to improve the generalization ability [1]. The term “ensemble” can be used to describe the paradigm that brings together a number of learning machines to provide a single output. This technique originates from Hansen and Salamon’s work [1], which shows that the generalization ability of a neural network can be significantly improved through ensembling a number of neural networks. Because of the simple and effective properties, neural network ensemble has become a hot topic in machine learning communities and has already been successfully applied to many areas, for example face recognition [2], character recognition [3], image analysis [4], etc.

It is demonstrated by both theoretical [1][5] and empirical studies [6] that the generalization ability of ensemble depends greatly on both accuracy and diversity among individual classifiers in the ensemble. However, it is difficult to design an accurate yet diverse ensemble since there is a trade-off between accuracy and diversity in the ensemble. The

existing ensemble algorithms pay more attention to either accuracy or diversity. For example, Bagging, Bagging of feature and random forests focus on diversity by randomization of data or/and feature and Adaboost concentrates on accuracy by changing the sampling weight of each sample to reduce the training error.

Bagging [7] and Random feature subspace [8], the well-known ensemble algorithms which have attracted an extensive research interest, employ bootstrap sampling [9] of training data and random selection of feature subsets to promote the diversity and thus improve the performance of ensemble. Random forests [10], a successful example that combines bootstrap sampling and random subspace method, holds an excellent generalization ability in many areas.

However, these ensemble algorithms (Bagging, Random feature subspace and Random Forests which rely on randomization of data or/and feature) pay more attention to diversity but ignore the importance of accuracy of individual classifiers in the ensemble. This may degrade the performance of ensemble. Take random forest for example, though the generalization error converges to a limit as the number of trees in the forest becomes large, the predictions may fluctuate greatly because simultaneous randomization of data and features may lead to weak accuracy of individual classifiers in the ensemble. The disadvantage of these algorithms inspires us to develop an algorithm which pays attention to both accuracy and diversity.

This paper proposes to incorporate evolutionary algorithm and negative correlation learning to automatically design and train random neural network ensembles, i.e. simultaneous sampling of data and feature to train individual neural networks to constitute ensemble. In order to promote the diversity among individual classifiers, bootstrap sampling and random subspace method are employed together to generate the initial neural ensemble. This will encourage different individual NNs in the ensemble to learn different parts or aspects of the training data so that the ensemble can learn better the entire training data. Then an evolutionary algorithm with negative correlation learning has been employed to search for a population of diverse individual NNs that together solve a problem. Negative correlation learning [11], which is a successful neural network ensemble learning technique, creates negatively correlated NNs using a correlation penalty term in the error function to encourage different individual networks in the ensemble to cooperate with each other. In negative correlation learning, the individual networks are trained simultaneously, rather than independently or sequentially. Evolving the ensemble

Huanhuan Chen is with The Centre of Excellence for Research in Computational Intelligence and Applications (Cercia), School of Computer Science, University of Birmingham, Birmingham B15 2TT, United Kingdom (Phone: +44-121-4143736; email: H.Chen@cs.bham.ac.uk).

Xin Yao is with The Centre of Excellence for Research in Computational Intelligence and Applications (Cercia), School of Computer Science, University of Birmingham, Birmingham B15 2TT, United Kingdom (Phone: +44-121-4143747; email: X.Yao@cs.bham.ac.uk).

with negative correlation learning emphasizes not only the accuracy of individual NNs but also the cooperation among different individual NNs and thus improve the generalization. In the evolving processing, the algorithm will keep the randomization of data and feature to maintain the diversity in the ensemble.

As each member in the ensemble is learned from bootstrap sample of the training examples, which typically omits $1/e \approx 37\%$ of the training examples, out-of-bag (OOB) estimation, based on recording the votes of each member on those training examples omitted from its bootstrap sample and aggregating the votes for each training example for an estimation of the generalization error, serves another benefit of this algorithm.

The rest of this paper is organized as follows. After the background description in Section II, the proposed algorithms are described in Section III. Experimental results and discussion are presented in Section IV. Finally, Section V will conclude the paper and discuss future work.

II. BACKGROUND

Neural network ensembles [1] is a learning paradigm where a collection of neural networks are trained for the same task. There have been many ensemble methods studied in the literatures, such as Bagging [7], Boosting [12], ensemble of features and so on.

Bagging is proposed in [7] based on bootstrap sampling [9]. In a Bagging ensemble, each base learner is trained on a set of n training samples, drawn uniformly at random with replacement from the original training set of size n . Predictions on new samples are made by simple averaging. For unstable learners such as neural networks or decision trees Bagging works very well. Generally speaking, Bagging can reduce the error due to deduction of variance of the base learner [13].

AdaBoost is one of the best known variations of Boosting [12]. Its main idea is to introduce weights on the training set \mathcal{D} and pay more attention to those training samples that are misclassified by former classifier in the training of next classifier. It is widely believed that AdaBoost approximately maximizes the margins of the training samples [14]. AdaBoost often gives a satisfying result for many different applications. However, it seems to be very sensitive to noise [15].

Apart from randomly sampling the training set, random feature subspace [8] method generates ensemble by adopting different feature subsets for different ensemble members to promote the diversity [8] [16] and most of the existing ensemble feature methods claim better results than traditional methods [17][18], especially when the data set has a large number of features and not too few samples [8].

Random forests [10] combines Bootstrap sampling and random subspace method to generate decision forests. It is consisted of a number of decision trees which grow with the examples bootstrap sampled from the training set and randomly employ different feature subsets for different decision nodes. Random forests performs the same as Adaboost in

terms of error rate, but are more robust with respect to noise. Though the generalization error converges to a limit as the number of trees in the forest becomes large, the predictions may fluctuate because of simultaneous randomization of data and features.

In order to construct more flexible and efficient ensembles, evolutionary algorithms have been employed to train neural network ensemble. Negative correlation learning [19] [20] is a successful neural network ensemble learning algorithm originated from the evolutionary computation literature. It is different from previous work such as bagging or boosting and it emphasizes interaction and cooperation among the individual base learners in the ensemble, and uses an unsupervised penalty term in the error function to produce biased individual base learners whose error tend to be negatively correlated. Islam et al. [21] take a constructive approach to building the ensemble, starting from a small group of networks with minimal architecture. The networks are all partially trained using negative correlation learning. To our knowledge, the approach can automatically determine weights, network topologies and ensemble membership. However, the process of constructing ensemble is complicated and there are many parameters needed to be tuned in practice.

Diverse and Accurate Ensemble Learning Algorithm [22] is an approach that combines evolving neural network and multiobjective algorithm. In this paper, adaptive Gaussian variance is developed for generating the offsprings and Mimetic Pareto Neural Network Algorithm [23] is used for evolving neural network. Finally, diverse and accurate classifier can be achieved through these procedures.

Oliveira et al.'s work [17] incorporates ensemble of feature selection and multiobjective algorithm to solve handwritten word recognition. This algorithm produces a set of classifiers which have small number of features and low error rate, which is achieved by evolving these classifiers with different *randomly* chosen features. The combination weight of ensemble is obtained by multiobjective algorithm with two different objectives: diversity and accuracy.

Cooperative Coevolution of Artificial Neural Network Ensembles [24] combines the coevolution of different subpopulations of diverse networks and the evolution of the combination weights of these networks. In this algorithm, the cooperation with the rest of the networks is defined as one objective, each network is evaluated in the evolutionary process using multiobjective method. Thus, the algorithm encourages the collaboration among ensemble and improves the combination schemes for ensemble.

Chen et al. [25] proposed to incorporate evolutionary multiobjective algorithm and Bayesian Automatic Relevance Determination (ARD) to automatically design and train ensemble. The algorithm adopts different feature subsets, selected by Bayesian ARD, to maintain accuracy and promote diversity among individual NNs in an ensemble and employs multiobjective algorithms to encourage the networks with lower error rate and fewer features.

III. EVOLVING RANDOM NEURAL ENSEMBLES WITH NEGATIVE CORRELATION LEARNING (ERNE)

It is widely believed that the success of ensemble algorithms depends on the accuracy and diversity among these base classifiers [26]. In general, the individual classifiers in ensemble are designed to be accurate and diverse. For example, Bagging relies on bootstrap that produces different subsets of the training data; Ensemble of features employs different features instead of training data to generate diverse ensemble [8]. Random forests [10] combines bootstrap sampling and random subspace method to generate more diverse ensembles. However, the predictions of random forests may fluctuate because of simultaneous randomization of data and features. Although the disadvantage of this could be slightly offset by including more and more decision trees in the ensemble, this of course leads to extended training times and more resources consumed.

The existing methods, random sampling of data and features, may promote the diversity but degrade the accuracy. How to improve the accuracy and simultaneously maintain the diversity to make sure that the obtained ensemble is both accurate and diverse is a key factor for ensemble algorithms. ERNE offers a natural way to optimize accuracy and simultaneously maintain the diversity among the individuals in the ensemble. In the algorithm, randomization of both data and features have been adopted/kept to generate/maintain the diversity in the ensemble. Evolutionary ensemble with negative correlation learning provides the opportunities for these individual NNs to negatively correlated with each other and thus improves the accuracy of these individual NNs.

In this paper, we are interested in evolutionary algorithms since combining individual NNs from a population into a NNs ensemble has a close relationship with the design of NN ensembles. Though the individual in the evolution population is not an appropriate solution for machine learning problems, since the individual with the lowest training error might not be the individual with the best generalization error, combining the entire population would be expected to have better generalization than any single individual. Previous works have acknowledged the successful applications of evolutionary learning for generating neural network ensembles [19], [11].

Our method for designing neural network ensemble has made full use of the feature of diversity generation techniques: bootstrap of data and random feature subspace, and evolutionary ensemble with negative correlation learning to construct the ensemble, which make the individual neural networks accurate and diverse.

A. Evolutionary Random Ensemble

In this subsection, we will present an ensemble training algorithm based on evolving random neural ensemble with negative correlation learning.

The algorithm firstly generates an initial population of Neural Networks (NNs), each of which is trained on bootstrap of training data and random feature subspace. Then

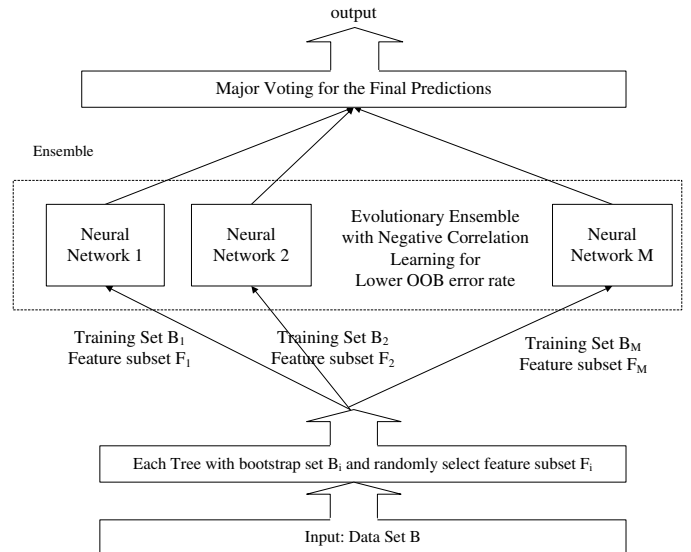


Fig. 1. The architecture of ERNE

the diverse population is evolved to improve the accuracy of individual NN.

The whole process could be illustrated as following.

- 1) Sample the original training set and obtain M replications of training set $\{B_i\}_{i=1}^M$.
- 2) Generate an initial population of M Neural Networks (NNs), the number of hidden nodes for each NN, $n_i (i = 1, \dots, M)$ is specified randomly restricted by the maximal number of hidden nodes. The random initial weights are distributed uniformly inside a small range.
- 3) Train each NN on each bootstrap set B_i with randomly selected feature subset $\{F_i\}_{i=1}^M$ for a certain number of epochs that is proportional to the number of hidden nodes using negative correlation learning and calculate the out-of-bag estimation as the ensemble fitness function.
- 4) In each generation, randomly choose s NNs to create offspring NNs¹. For each offspring s_i , evolve each s NNs with Gauss mutation², and train it with its corresponding parent's bootstrap set B_i and feature subset F_i . s is specified by the user.
- 5) Compare the fitness of each s_i NN with their respective parents and include the better one in the population and recalculate the out-of-bag error as the fitness.
- 6) Go to the next step if the maximum number of generations has been reached. Otherwise, and go to Step 3.
- 7) Combining the population to form the ensemble.

The architecture of our algorithm can be shown in Figure 1. There are four advantages of this algorithm: (1) Ensemble of different data subset and feature subset promote the

¹Each individual, selected to be mutated with equal probability, reflects the emphasis on evolving a diverse set of individuals.

²Add Gauss noise to the weight vector of neural network. The parameter of Gauss noise is: $mean = 0$ and $variance = \mu$, will be specified manually.

diversity among individual classifiers in the ensemble. (2) Evolving the individual NN in the ensemble helps to improve the accuracy. (3) Negative correlation learning enables these individual NNs in the ensemble correlated with each other and improves the generalization performance. (4) It generates an internal unbiased estimate of the generalization error, OOB, as the NN ensemble building progresses.

B. Negative Correlation Learning

Negative Correlation Learning (NCL), a successful neural network ensemble learning technique developed in the evolutionary computation literature, has shown a number of empirical successes and varied applications, including regression problems [27] and classification problems [11]. It has consistently showed very competitive results with other techniques like Mixtures of Experts, Bagging, and Boosting [19] [28] [29].

NCL introduces a correlation penalty term into the error function of each individual network in the ensemble so that all the networks can be trained simultaneously and interactively on the same training data set. The error function e_i for network i is defined by

$$e_i = \frac{1}{2}(f_i - d)^2 + \lambda p_i \quad (1)$$

where d is the true value of the training point, λ is a weighting parameter on the penalty term p_i :

$$p_i = (f_i - \bar{f}) \sum_{j \neq i} (f_j - \bar{f}) \quad (2)$$

and \bar{f} is defined by $\bar{f} = \frac{1}{M} \sum_{i=1}^M f_i$.

The first term in the right side of (1) is the empirical risk function of network i . The second term, p_i , is a correlation penalty function. The purpose of minimizing p_i is to negatively correlate each network's error with errors for the rest of the ensemble. The λ parameter controls a trade-off between the objective and penalty terms. With $\lambda = 0$ we would have an ensemble with each network training with plain back propagation, exactly equivalent to training a set of networks independently of one another. If λ is increased, more and more emphasis would be placed on minimizing the penalty.

Liu et al. [11] implemented NCL by gradient descent method for training neural network. From equation (1), the minimization of the empirical risk function of the ensemble is achieved by minimizing the error functions of the individual networks. In fact, negative correlation learning provides a novel way to decompose the learning task of the ensemble into a number of subtasks for different individual networks.

C. Out-of-Bag Fitness Evaluation

In ERNE, out-of-bag (OOB) estimation error is taken as the objective to be optimized. As each member in the ensemble is learned from bootstrap sample of the training examples, which typically omits $1/e \approx 37\%$ of the training examples. The out-of-bag estimate is based on recording the votes of each member on those training examples omitted

from its bootstrap sample and aggregating the votes for each training examples for an estimation of the generalization error. Out-of-bag estimates [30] is proposed as an ingredient in estimates of generalization error, which has been empirically supported by [31] that the out-of-bag estimate is as accurate as using a test set of the same size as the training set. ByLander [32] examined the out-of-bag estimate and found that out-of-bag estimate and 10-fold cross-validation have similar performance. Because no additional predictors are generated, out-of-bag estimate requires considerably less time than 10-fold cross-validation.

IV. EXPERIMENTS

This section will evaluate performance of ERNE on six well known benchmark problems, which cover a wide variety of problems including different numbers of patterns, different kinds of input types and different application areas. The results of these problems can reveal the performance of our algorithm clearly. In the following, experimental details and results are described.

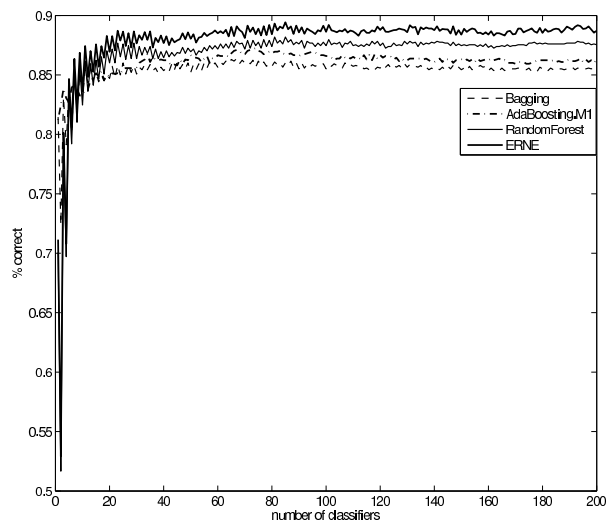
TABLE I
SUMMARY OF DATA SETS

Data Set	Examples	No. Features	Output Classes
Card	690	14	2
Diabetics	768	8	2
Cancer	277	9	2
Sonar	208	60	2
Ionosphere	351	34	2
Splice	3175	60	2

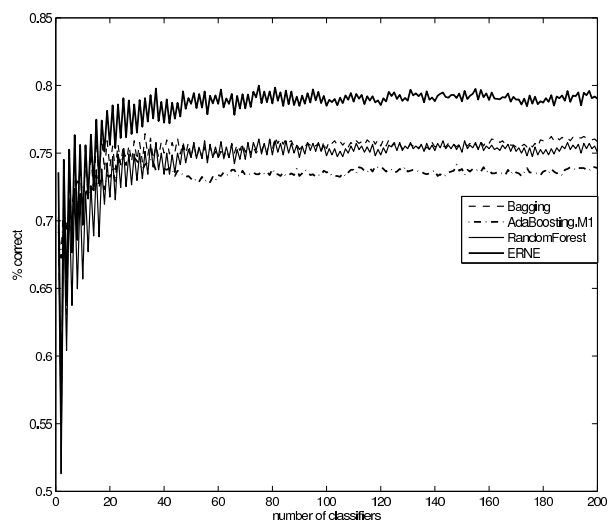
A. Experimental Setup

In this experiments, six benchmark problems are used to evaluate the performance of ERNE: they are Australian credit card assessment problem, the breast cancer problem, the diabetics problem, sonar signal classification, ionosphere and splice problems, which are obtained from the UCI machine learning benchmark repository [33].

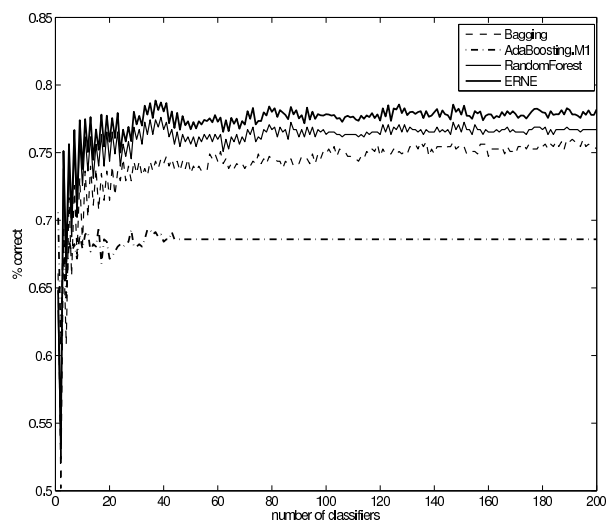
As we know, the experimental results depend on the partitions of data set. It may vary significantly for different partitions of the same data set. The old benchmark methodology has been criticized for its strategy to fix the training, validation and test data because the strategy will lead to poor estimation of generalization error and the situation will be especially worst when the test data becomes smaller. In this paper, cross validation is employed as the criteria to evaluate the performance. In k -fold cross-validation [34], the data is divided into k subsets of equal size. Then train the classifier k times, each time leaving out one of the subsets from training, but using only the omitted subset to compute error rate. If k equals the sample size, it is called leave-one-out cross-validation which gives an almost unbiased estimate of expected generalization error [35]. This paper employs two-fold cross validation to allow a sufficient test set to estimate the generalization error.



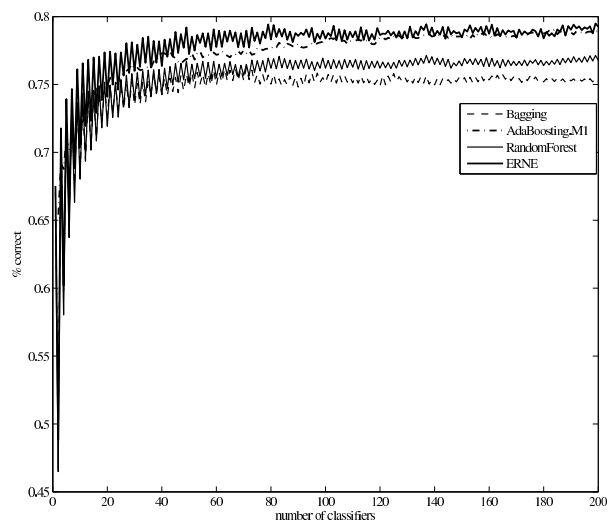
(a) Card



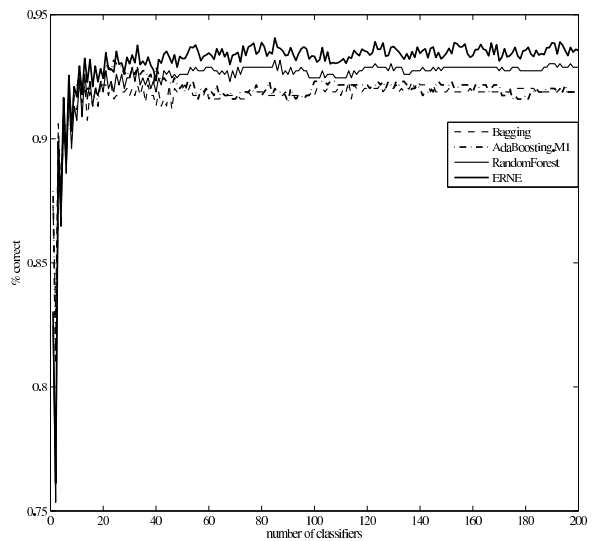
(b) Diabetics



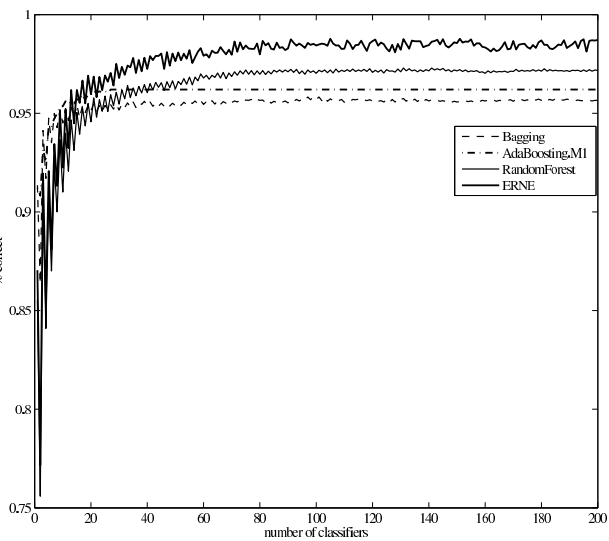
(c) Cancer



(d) Sonar



(e) Ionosphere



(f) Splice

Fig. 2. Performance comparison of ERNE against three tree-based ensemble approaches: Bagging, Adaboosting and random forests, in predicting the six problems based on two-fold cross validation. The differences between ERNE and other classifiers are significant at the 5% significance level, see Table III

TABLE II

COMPARISON AMONG ERNE WITH OTHER WORKS IN TERMS OF AVERAGE CROSS VALIDATION ERROR FOR SIX DATA SETS. THE RESULTS ARE AVERAGED ON 20 RUNS OF 2-FOLD CROSS-VALIDATION, RESPECTIVELY

% error	ERNE	Random Forests	Bagging	Adaboosting
Card	11.28	12.46	14.64	13.70
Diabetics	20.97	24.93	24.15	26.11
Cancer	22.16	23.29	24.73	31.41
Sonar	20.74	23.22	24.76	21.08
Ionosphere	6.44	7.12	8.11	8.12
Splice	1.28	2.83	4.35	3.80

TABLE III

RESULTS (P VALUE) OF TWO-TAIL PARED T-TEST IN TERMS OF PREDICTION ERROR BETWEEN ERNE AND RANDOM FORESTS, BAGGING, ADABOOST ON CARD, DIABETICS, CANCER, SONAR, IONOSPHERE AND SPLICE PROBLEMS

Method	Card	Diab.	Cancer	Sonar	Iono.	Splice
R. F.	0.0187	0.0101	0.0287	0.0349	0.0426	0.0278
Bagging	0.0062	0.0152	0.0124	0.0008	0.0112	0.0016
Adaboost	0.0127	0.0057	0.0000	0.2431	0.0198	0.0093

The input attributes are rescaled to between 0.0 and 1.0 by a linear function as the preprocessing procedure. The network we used in this paper is three-layer feed-forward NN. The number of hidden nodes will be initialized randomly but restricted in the range 3 to 8. Initial connection weights for individual NNs in an ensemble are randomly chosen. The parameter λ is set to 0.8 and the variance of Gaussian mutation is 1.

The parameters in use are set to: the population size M (200), the number of offspring s ($\max[20, M]$), the number of generations (100). These parameters are chosen after some preliminary experiments. They are not meant to be optimal.

B. Experimental Results

In this experiment, we have compared ERNE with other ensemble methods: Bagging of Decision Trees, Adaboosting of Decision Trees, i.e. Adaboost.M1 algorithm in [12], and Random Forests. In these ensemble approaches, regression and classification tree (CART) is used as base classifiers. In order to allow a sufficient test set to estimate the generalization error, two-fold cross validation is employed to evaluate these methods.

Figure 2 shows the results of ERNE over 20 independent cross validation on six different problems. In each sub-figure, we record the performance of these four algorithms with respect to the scale of the ensemble, i.e. the number of classifiers in this ensemble. Table II lists the comparisons. In the table, we show the average error of classification over the 20 runs and Table III gives the result of two-tail pared t-test in terms of prediction error between ERNE and other classifiers.

From Figure 2, ERNE consistently outperforms other

algorithms in terms of cross validation error. In [36], we also evaluated ERNE on Ames test mutagenicity problem and ERNE exhibits good performance as well. This is understandable since the performance of random forests is better than or similar as the other ensemble algorithms in most of the cases, and ERNE maintains good diversity by adopting bootstrap and random feature subspace, which is similar as random forests, and evolves the ensemble to optimize the accuracy and cooperation. Generally speaking, ERNE will perform no worse than random forests.

The success of random forest, which is the state-of-the-art algorithm [10], is largely due to that it uniquely adopts both bootstrap of data and feature subset selection, which is particularly of value for handling a data set with a large amount of feature variables [8]. ERNE not only keeps the benefits of random forests but also improves its performance by optimizing the accuracy. The superiority of ERNE over random forests can be observed in the experimental results. In our experiments, we also find that Adaboost.M1 of trees sometimes overfits when adding more and more trees in the ensemble.

There are two points to explain the reason why the performance of our algorithm is better than the performance of others.

- ERNE generates a diverse ensemble. Firstly, bootstrap sampling of data and random feature subsets generate a diverse ensemble in the initial population, which inherits the merits of random forests. In the evolving stage, the diversity is maintained by only mutating the weight of individual NN but not changing the bootstrap of data and the feature subset used by this individual NN.
- Evolving ensemble with negative correlation learning optimizes the accuracy and cooperation of the existing individual NNs in the ensemble. The combination of the two points, optimization of accuracy and maintenance of diversity, reduces the generalization error and makes ERNE powerful.

V. CONCLUSIONS

This paper proposes an efficient ensemble algorithm that combines bootstrap of data, random feature subspace and evolutionary algorithm with negative correlation learning. ERNE adopts bootstrap sampling of training points and random feature subsets to promote and maintains diversity among individual NNs in the ensemble. In the following, evolving random neural ensemble with negative correlation learning provides the opportunities for these individual NNs to negatively correlated with each other and thus improves the accuracy of these individual NNs. Out-of-bag (OOB) estimation, which requires considerably less time than 10-fold cross-validation and saves the data points for validation, facilitates fitness evaluation in the evolutionary algorithm.

Several experiments have been carried out in this paper to evaluate how does ERNE perform on different problems in comparison with other ensemble algorithms. ERNE has shown an excellent performance in the solution of these data

sets. In the experimental discussion, we conclude two reasons why the performance of ERNE outperforms others.

The following work is to analyze ERNE further. It is also interesting to implement ensemble selection algorithms to select the effective combination of NNs to constitute a small ensemble instead of ensembling all of them, whose aims are to improve the generalization performance and to save the computational resources.

ACKNOWLEDGMENT

This work is partially supported by a Dorothy Hodgkin Postgraduate Scholarship to the first author.

REFERENCES

- [1] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
- [2] F. J. Huang, T. Chen, Z. Zhou, and H. Zhang. Pose invariant face recognition. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, pages 245–250, Washington, DC, USA, 2000.
- [3] L. K. Hansen, L. Liisberg, and P. Salamon. Ensemble methods for handwritten digit recognition. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 333–342, Helsingoer, Denmark, 1992.
- [4] K. J. Cherkauer. Human expert level performance on a scientific image analysis task by a system using combined artificial neural networks. In *Proceedings of AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, pages 15–21, Menlo Park, CA, USA, 1996.
- [5] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems 7*, pages 231–238, Denver, Colorado, USA, 1995.
- [6] S. Hashem. Optimal linear combinations of neural networks. *Neural Networks*, 10(4):599–614, 1997.
- [7] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [8] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
- [9] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, London, U.K., 1993.
- [10] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [11] Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Transaction on Evolutionary Computation*, 4(4):380–387, 2000.
- [12] R. E. Schapire. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406, 1999.
- [13] J. H. Friedman. Bias, variance, 0-1 loss and the curse of dimensionality. Technical report, Stanford University, 1996.
- [14] G. Rätsch, T. Onoda, and K. R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.
- [15] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2003.
- [16] D. W. Optiz. Feature selection for ensembles. In *Proceedings of the 16th International Conference on Artificial Intelligence*, pages 379–384, 1999.
- [17] L. S. Oliveira, M. Morita, R. Sabourin, and F. Bortolozzi. Multi-objective genetic algorithms to create ensemble of classifiers. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, volume 87, pages 592–606, 2005.
- [18] J. O’Sullivan, J. Langford, R. Caruana, and A. Blum. Featureboost: A meta-learning algorithm that improves model robustness. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 703–710, San Francisco, CA, USA, 2000.
- [19] Y. Liu and X. Yao. Ensemble learning via negative correlation. *Neural Networks*, 12(10):1399–1404, 1999.
- [20] Y. Liu and X. Yao. Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 29(6):716–725, 1999.
- [21] M. M. Islam, X. Yao, and K. Murase. A constructive algorithm for training cooperative neural network ensembles. *IEEE Transaction on Neural Networks*, 14(4):820–834, 2003.
- [22] A. Chandra and X. Yao. Divace: Diverse and accurate ensemble learning algorithm. In *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning*, volume 3177, pages 619–625, 2004.
- [23] H. A. Abbass. A memetic pareto evolutionary approach to artificial neural networks. In *Proceedings of the fourteenth Australian Joint Conference on Artificial Intelligence*, volume 2256, pages 1–12, 2000.
- [24] N. García, C. Hervás, and D. Ortiz. Cooperative coevolution of artificial neural network ensembles for pattern classification. *IEEE Transactions on Evolutionary Computation*, 9(3):271–302, 2005.
- [25] H. Chen and X. Yao. Evolutionary multiobjective ensemble learning based on bayesian feature selection. In *Proceedings of IEEE Congress on Evolutionary Computation*, volume 1141, pages 267–274, 2006.
- [26] G. Brown, J. Wyatt, and P. Tino. Managing diversity in regression ensembles. *Journal of Machine Learning Research*, 6:1621–1650, 2005.
- [27] X. Yao, M. Fischer, and G. Brown. Neural network ensembles and their application to traffic flow prediction in telecommunications networks. In *Proceedings of International Joint Conference on Neural Networks*, pages 693–698, 2001.
- [28] 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.
- [29] R. McKay and H. Abbass. Analyzing anticorrelation in ensemble learning. In *Proceedings of 2001 Conference on Australian Artificial Neural Networks and Expert Systems*, pages 22–27, 2001.
- [30] D. H. Wolpert and W. G. Macready. An efficient method to estimate bagging’s generalization error. *Machine Learning*, 35:41–55, 1999.
- [31] L. Breiman. Out-of-bag estimation. Technical report, Stanford University, 1996.
- [32] T. Bylander. Estimating generalization error on two-class datasets using out-of-bag estimates. *Machine Learning*, 48(1-3):287–297, 2002.
- [33] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.
- [34] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the fifteenth International Conference on Artificial Intelligence*, pages 1137–1143, Qubec, Canada, 1995.
- [35] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
- [36] H. Chen and X. Yao. Evolutionary ensemble for in silico prediction of ames test mutagenicity. In *Proceedings of 2007 International Conference on Intelligent Computing*, 2007.