Building Optimal Committees of Genetic Programs

Byoung-Tak Zhang and Je-Gun Joung

Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
{btzhang,jgjoung}@scai.snu.ac.kr
http://scai.snu.ac.kr/

Abstract. Committee machines are known to improve the performance of individual learners. Evolutionary algorithms generate multiple individuals that can be combined to build committee machines. However, it is not easy to decide how big the committee should be and what members constitute the best committee. In this paper, we present a probabilistic search method for determining the size and members of the committees of individuals that are evolved by a standard GP engine. Applied to a suite of benchmark learning tasks, the GP committees achieved significant improvement in prediction accuracy.

1 Introduction

Committee machines are learning methods that make decisions by combining the results of several individual learners. The learners may be neural networks, decision trees, or any other algorithms. Recently, several authors have presented methods for combining models learned by means of evolutionary algorithms [1–4]. The basic idea behind this approach is that evolutionary algorithms are population-based distributed search methods that produce a variety of individuals. The diversity of the individuals generated during the evolution provide a rich resource for building committee machines.

Most research on evolutionary methods for committee machines has focused on two main issues: finding combination weights and creating diverse individuals. Examples of combining methods include weighted averaging [5] and majority voting [6]. The diversity of structures of committee members is essential in building effective committee machines. However, little research has been done to determine the committee size and to choose the members of the committee.

In this paper, we present a probabilistic method that finds an optimal committee through evolution of individuals that represent committee members. We assume Poisson distribution on the number of committee members whose mean is adapted during the search. An advantage of this approach is that the search can be fast since it focuses on the promising regions of the best committees. The method is applied to genetic programming (GP) to solve classification problems.

The results of GP committees are compared with those obtained by the standard GP and decision tree algorithms. The committee selection method achieved significant improvement in predictive accuracy of the standard GP engine.

The paper is organized as follows. In Section 2, we review previous work on evolutionary methods for building committee machines. Section 3 presents the probabilistic method for building the best committee from individuals evolved by a standard GP engine. Section 4 reports experimental results on the problems taken from the UCI-repository data sets. Section 5 contains conclusions.

2 Previous Work on Building Committee Machines

The basic idea behind the committee machine approach is to fuse knowledge acquired by individual experts to arrive at an overall decision that is supposedly superior to that attainable by any one of them acting alone [7]. Committee machines can be built in two different ways. One is to use a static structure. This is known generally as an ensemble method. Here, the input is not involved in combining committee members. Examples include ensemble averaging [8] and boosting [9]. The other method for building committees is to use a dynamic structure. This includes combining local experts such as mixtures of experts [10]. Here input is directly involved in the combining mechanism that uses an integrating unit such as a gating network adjusting the weights of committee members according to input. Most studies on committee machines have been based on neural networks [8, 11], decision trees [12], and statistical methods [13].

Some authors have used evolutionary algorithms for committee machines. Evolutionary algorithms generate a number of individuals during evolution. Most methods select the single best solution, discarding all remaining individuals generated during the evolution. However, the individuals evolved can be better utilized if they are combined to build committees. Opitz and Shavlik [1] presented the ADDEMUP method that uses genetic algorithms to search for a correct and diverse population of neural networks to be used in the ensemble. It has an objective function that measures both the accuracy of the network and the disagreement of that network with respect to the other members of the set. Yao and Liu [3] experimented with a variety of combination methods to integrate multilayer perceptrons that were evolved by evolutionary programming and a modified back-propagation algorithm. They also try to find a committee of variable size using a genetic algorithm. However, only the individuals in the final generation were used as candidates for the committee members. Neri and Giordana [14] introduced universal suffrage selection that chooses a suitable concept description to evolve partial concept descriptions as a whole. Here the concept description can be viewed as a committee member and universal suffrage selection may be regarded as a method for selecting committee members.

Zhang and Joung [2] presented the mixing GP (MGP) method that uses weighted majority voting to combine individuals evolved by genetic programming. It regards genetic programs as cooperating experts and makes the final decision on the basis of a pool of the best n experts from the population. Experimental results show that MGP improves the robustness of genetic programming. It also has the effect of accelerating the evolution speed since MGP requires smaller number of generations than a standard GP to obtain a specified level of performance. MGP has recently been extended to CENTs [4], the committees of evolutionary neural trees, to improve predictive accuracy of individual neural trees evolved. Both in MGP and CENTs, the size of committees was fixed. However, the committee size influences the performance of committee machines as well as the members of the committee. In the next section, we present an evolutionary search method that determines the appropriate size and members of committees.

3 Building Committees by Probabilistic Evolution

We consider an evolutionary method that builds committee machines in two separate stages. In stage one, given an evolutionary engine, a pool of diverse and fit individuals are evolved and kept. In stage two, these individuals evolved are used to find the best size and members of the committees.

3.1 Evolving Individuals

The goal of stage 1 is to generate good candidates for building committees. The candidates should be as diverse as possible. Let A_i denote an individual. We use symbol $A(g)$ to denote the population at generation g:

$$A(g) = \{A_i(g)\}_{i=1}^{M}, \tag{1}$$

where M is the size of the population. The populations generated during the maximum number G of generations constitute the space of candidate individuals for committees:

$$\mathcal{A} = A(0) \cup ... \cup A(G). \tag{2}$$

Starting from a random population, individuals are evolved by using genetic operators such as crossover and mutation. New generations of populations are produced until some termination criterion is met.

To measure the fitness of individuals at each generation, a set D of N data items are given:

$$D = \{(\mathbf{x}_c, y_c)\}_{c=1}^{N}, \tag{3}$$

where $\mathbf{x}_c$ is the input vector and y_c is the desired output. N is the total number of data items. The fitness of the ith individual is measured as the sum of errors and complexity of individual A_i:

$$F(A_i) = E(A_i) + \alpha C(A_i) \tag{4}$$

$E(A_i)$ is usually measured by the misclassification rate or the sum of squared errors on the data set D. The parameter α is the Occam factor that controls the accuracy and complexity of individuals [15]. $C(A_i)$ is based on the number of nodes and depth.

3.2 Evolving Committees

The goal of stage 2 is to find an optimal committee, optimal in the sense of the size and members. We use the symbol $V_k(g)$ to denote the kth committee at generation g. The population $V(g)$ at generation g of the evolutionary algorithm for evolving committees consists of

$$V(g) = \{V_k(g)\}_{k=1}^{K}, \tag{5}$$

where committee size m_k is variable for each committee $V_k(g)$. In this paper, we limit the maximum size K to 100 since the best individuals for each generation are considered as candidates and the typical number of maximum generation is 100. The fitness of committee V_k is defined similar to the fitness of individuals as follows:

$$R(V_k) = E(V_k) + \beta C(V_k), \tag{6}$$

where β controls the tradeoff between accuracy and complexity of committees. The error $E(V_k)$ of committee k is the total error on D made by the weighted average of committee members.

$$E(V_k) = \frac{1}{N}\sum_{j=1}^{m_k}\sum_{c=1}^{N}\left(v_{kj}f_j(\mathbf{x}_c) - y_c\right)^2. \tag{7}$$

Here v_{kj} is the jth component of the combining weight vector $\mathbf{v}_k$ of the kth committee and satisfies the condition $\sum v_{kj} = 1$. $f_j(\mathbf{x}_c)$ denotes the output value of the committee machine for input vector $\mathbf{x}_c$. If the type of the problem is classification, an indicator function can be used to count the number of misclassifications. In our approach, we use the generalized ensemble method (GEM) as the combining method [16]. The complexity of committee k is defined as the committee size m_k divided by training set size N:

$$C(V_k) = \frac{m_k}{N}. \tag{8}$$

If all the individuals over the whole generations are considered as candidate members for committees, the number of combinations for the committee is very large. We consider only the best individual $A_{best}(g)$ at each generation as the candidates. This is reasonable since a good fitness is a minimum requirement for good committee members. To maintain the diversity of individuals, we do not use elitist selection. Let B be the set of the best individuals:

$$B = \{B_g\}_{g=1}^{S}, \; B_g = A_{best}(g). \tag{9}$$

Here S is the maximum size of the pool. Each committee member B_i is selected with the probability

$$P(B_i) = \frac{\exp\{F(B_i)/T\}}{\sum_{j=1}^{S}\exp\{F(B_j)/T\}}, \tag{10}$$

where T is the constant temperature that determines the randomness of the selection. Equation (10) says that the candidate with a high fitness is selected with a higher probability. In fact, the useful individuals for building committees may be the individuals found in early generations. This selection scheme does not exclude the possibility of selecting the candidates with low fitness.

3.3 Probabilistic Aspects of Committee Selection

The size of the search space for the optimal committee V^* is $(2^S - 1)$ for pool size S. This is a large number. The search time can be reduced by using a strategy that concentrates on the search space of high performance. Equation (6) expresses that the complexity term $C(V_k)$ prevents the committee size from growing unnecessarily.

The search for optimal committees can be formulated as a Bayesian inference problem. Here, we show how the committee selection can be guided by probabilistic models. Let $P(V_k)$ denote the prior probability of committee V_k. Once we observe the data D, the likelihood $P(D|V_k)$ of the committee can be computed. Bayes rule provides a method for combining the prior and likelihood to obtain the posterior probability $P(V_k|D)$ of the committees:

$$P(V_k|D) = \frac{P(D|V_k)P(V_k)}{P(D)}. \tag{11}$$

According to the Bayesian framework for evolutionary computation [17], this problem can be solved by an evolutionary algorithm. The objective here is to find the best commitee V^* that maximizes $P(V_k|D)$. Note that maximizing (11) is equivalent to maximizing its numerator since $P(D)$ does not depend on V_k. Note also that a committee machine can be parameterized as $V_k = (\mathbf{v}, m)$, where m is the number of committee members and $\mathbf{v}$ is the values for mixing parameters for the members. Thus, we have

$$\begin{aligned} P(V_k|D) &\propto P(D|V_k)P(V_k) & (12)\\ &= P(D|\mathbf{v}, m)P(\mathbf{v}, m) & (13)\\ &= P(D|\mathbf{v}, m)P(\mathbf{v}|m)P(m). & (14) \end{aligned}$$

In the experiments, we assume the normality of distributions for $P(D|\mathbf{v}, m)$ and $P(\mathbf{v}|m)$. Under these assumptions, it can be shown that minimizing the sum of squared errors (7) maximizes the posterior probability. We also assume that committee size m is distributed according to the following Poisson distribution

$$P(m-2) = \frac{\lambda^{m-2}\exp(-\lambda)}{(m-2)!}, \quad m = 2, 3, 4, \ldots, \tag{15}$$

where λ is the average size of committees for the generation. The Poisson distribution turned out to be useful since it tends to sample smaller sizes of committees. The mean value λ was chosen for each generation with respect to the fitness for the best committee with size m:

$$P(\lambda) = \frac{R(V_{best}^{\lambda}(g-1))}{\sum_{m=2}^{K} R(V_{best}^{m}(g-1))}, \tag{16}$$

where $V_{best}^{\lambda}(g-1)$ is the best committee with size λ at generation $g-1$. Through this adaptation, the search for an optimal size focuses more on the regions where the fitter committees were frequently generated. Here $R(V)$ is the fitness of committee V measured on the validation-set which is distinct from the training set used to determine the weight of each member.

4 Empirical Results

4.1 Experimental Setup

The performance of the committee selection method was evaluated on the UCI data [18]. Four different problems were chosen, i.e. breast cancer, Australian credit card, heart disease, Pima-indians diabetes. These have different characteristics in several aspects, including the number of attributes, the number of examples, the distribution of positive and negative examples, and mixed numeric/discrete attributes. We used genetic programming to evolve solutions to these problems. The performance was measured by a 10-fold cross-validation for each data set.

The parameters for genetic programming were: maximum number of generations = 100, population size = 200, selection method = ranking selection, reproduction rate = 0.01, and mutation rate = 0.01. In this experiment, we did not use elitism. If it is used, the same best individuals tend to appear every generation, reducing the diversity of the genetic pool for building committee members. We used the GP function set consisting of arithmetic operators, such as $+, -, \times, \div$, and $\geq$. These functions have also been used in [19] for solving classification problems.

The parameter values for building committees in the second stage were: maximum number of generations = 50, population size = 25, pool size = 100, temperature for selection probability $T = 0.01$, initial mean of Poisson distribution $\lambda = 10$, maximum number of committees = 20, Occam factor for fitness of committees $\beta = 0.1$. Here the pool size is the same as the number of generations in genetic programming.

4.2 Experimental Results

Figures 1 to 3 show the results for 100 runs on the diabetes data. Figure 1 shows the evolution of value λ of the Poisson distribution as a function of generation. Figure 1(a) shows that λ changes adaptively with respect to the distribution of the size of best committees. Here, the committee V_{best}^{m}, the best committee with size m over the generations, does not mean the best committee in the final generation. The best committee appeared mostly in the middle of the evolution.

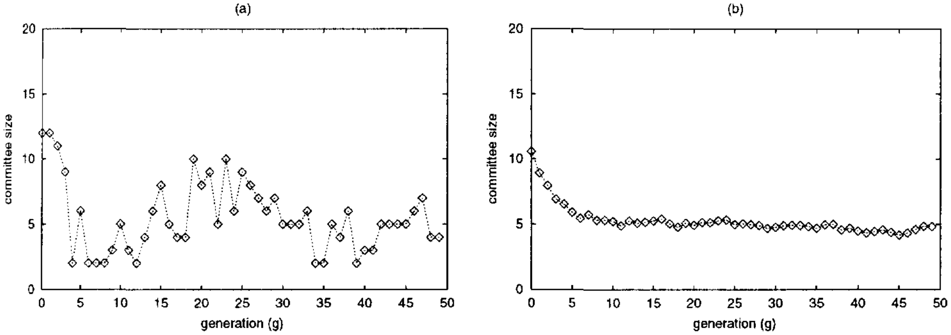

Fig. 1. Evolution of the best committee size (used as the mean value λ of the Poisson distribution) as a function of generation: (a) result for one run, (b) result averaged over 100 runs.

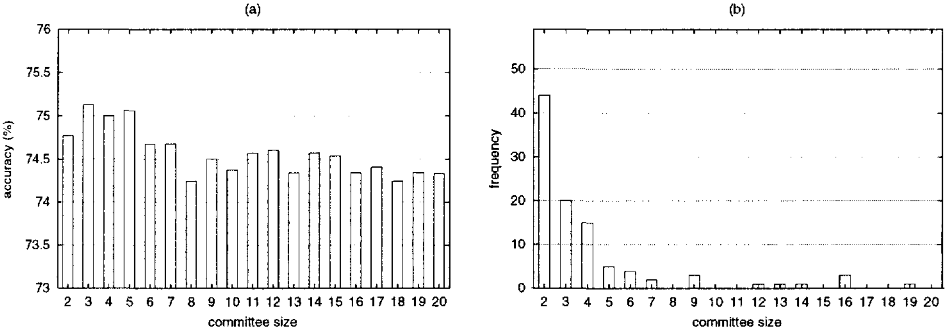

Fig. 2. Accuracy and frequency as a function of committee size: (a) Accuracy of the best committee for a fixed size. (b) Frequency of committees when the best committee is selected by the proposed committee selection method.

In this example, V^m_{best} appeared in generation 31 and m was 5. In Figure 1(b), the curve shows a convergence of λ to 5.

Figure 2 compares the performance of the presented method with that for not using committee selection. The left figure shows the performance by hill-climbing search with a fixed size. In the experiment, the number of iterations of hill-climbing was the same as the total number of generations multiplied by the size of the population in the step for evolving committees. In this particular experiment, the result shown in Figure 2(b) indicates that committee sizes of 2 to 7 have a higher accuracy than others.

Figure 3 plots the best fitness and the frequency for each size of the committee as generation goes on. The left figure shows the general tendency that the accuracy of each size gets higher as generation goes on. The right figure shows that in this problem the best committees concentrate around the small sizes for all generations. Figure 4 shows the average size of the best committee. Larger

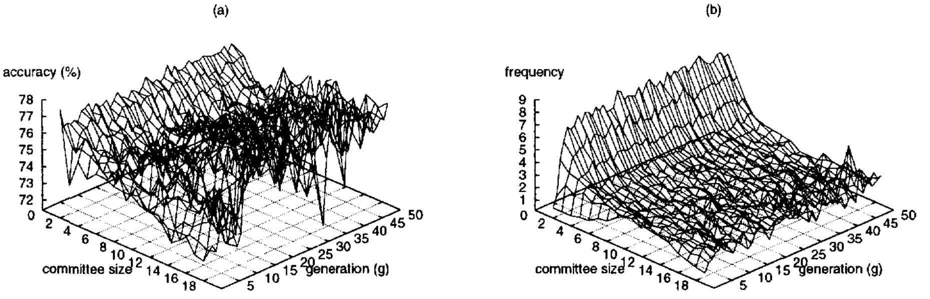

Fig. 3. Evolution of committees: (a) Accuracy of the best committee vs. generation vs. committee size. (b) Frequency vs. generation vs. committee size.

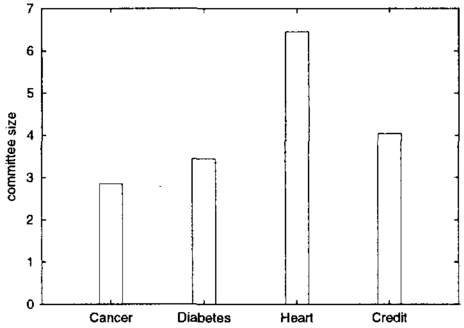

Fig. 4. Average size of the best committee for different problems.

Table 1. Comparison of misclassification rates for C5.0, C5.0 Boosting, the standard GP, and the GP committees on the four UCI data sets. The values are averages of ten standard 10-fold cross-validations.

		Average error rate			
Data set	Data size	C5.0	Standard GP	C5.0Boosting	GP Committee
Cancer	699 10-fold	5.42± 4.1	4.05± 2.8	3.14± 3.2	3.55± 2.1
Credit	690 10-fold	13.63 ± 4.4	14.63± 4.7	13.34± 3.3	12.02± 4.0
Heart	270 10-fold	22.96± 8.7	19.25± 7.7	19.63± 9.3	17.30± 7.2
Diabetes	768 10-fold	23.69± 6.5	25.50± 5.1	24.73± 5.5	24.90± 4.7

committees were more effective for the heart problem, while smaller committees were effective for the cancer and diabetes problems. The size of the best committee was less than 7 on average. Table 1 compares the average generalization error for the decision tree learner C5.0, C5.0Boosting, standard genetic programming (GP), and the GP commiittee using the presented committee selection method. Here the baseline results by C5.0 are the values reported in [20].

As in [20], we evaluated the results by using ten standard 10-fold cross-validations. The standard GP achieved better performances for C5.0 in the two problems (breast cancer and heart disease) out of four. The GP committee improved the performance of the standard GP and achieved better results than C5.0 in three out of four. GP committee outperformed even C5.0Boosting in two cases out of four.

5 Conclusions

This paper describes a new approach to searching an optimal committee. This approach includes the characteristic that searches the optimal committee size probabilistically. It also has a mechanism for penalizing large committees to promote compact committees. Therefore, redundant members tend to be removed from the committee. Experiments have been performed on a suite of four problems from the UCI machine learning database. Our experimental results show significant improvement in generalization accuracy by selecting committee size and members. Compared to simple genetic algorithms or other evolutionary algorithms, a distinguishing feature of the probabilistic evolutionary search is the sequential sampling of the size and the members of the committee, which can be naturally implemented in the Bayesian framework for evolutionary computation. The scaling properties of the presented method for committee selection for even larger committee sizes remain to be studied.

Acknowledgments

This research was supported in part by the Korea Ministry of Science and Technology through KISTEP under grant BR-2-1-G-06, by KOSEF Grant 981-0920-107-2 and by BK21 Program.

References

1. Opitz, W., Shavlik, J.W.: Actively Searching for an Effective Neural-Network Ensemble. *Connection Science*, **8** (1996) 337–353.
2. Zhang, B.-T., Joung, J.-G.: Enhancing Robustness of Genetic Programming at the Species Level. *Genetic Programming Conference (GP-97)*, Morgan Kaufmann, (1997) 336–342.
3. Yao, X., Liu, Y.: Making Use of Population Information in Evolutionary Artificial Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics*, **28B**(2) (1998) 417–425.

4. Zhang, B.-T., Joung, J.-G.: Time Series Prediction Using Committee Machines of Evolutionary Neural Trees. *Proceedings of the 1999 Congress on Evolutionary Computation*, **1** (1999) 281–286.
5. Perron, M.P.: *Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization.* PhD thesis, Department of Physics, Brown University, (1993).
6. Littlestone, N., Warmuth, M.K.: The Weighted Majority Algorithm. *Information and Computation*, **108** (1994) 212–261.
7. Haykin, S.: *Neural Networks, a Comprehensive Foundation.* Prentice Hall. (1994).
8. Hansen, L., Salamon, P.: Neural Network Ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12** (1990) 993–1001.
9. Drucker, H., Cortes, C., Jackel, L.D., LeCun, Y., Vapnik, V.: Boosting and Other Ensemble Methods. *Neural Computation*, **6**(6) (1994) 1289-1301.
10. Jacobs, R.A.: Bias/variance Analyses of Mixture-of-Experts Architectures. *Neural computation*, **9** (1997) 369–383.
11. Hashem, S.: Optimal Linear Combinations of Neural Networks. *Neural Networks*, **10**(4) (1997) 599–614.
12. Opitz, D., Maclin, R.: Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, **11** (1999) 169–198.
13. Lemm, J.C.: Mixtures of Gaussian Process Priors. *The Ninth International Conference on Artificial Neural Networks (ICANN 99)*, (1999) 7–10.
14. Neri, F., Giordana, A.: A Parallel Genetic Algorithm for Concept Learning. *The Sixth international Conference on Genetic Algorithms*, (1995) 436–443.
15. Zhang, B.-T., Ohm, P. and Müehlenbein, H.: Evolutionary Induction of Sparse Neural Trees. *Evolutionary Computation*, **5**(2) (1997) 213–236.
16. Perron, M.P., Cooper, L.N.: When Networks Disagree: Ensemble Methods for Hybrid Neural Networks. *Artificial Neural Networks for Speech and Vision*, Chapman & Hall, (1994) 126–142.
17. Zhang, B.-T.: A Bayesian Framework for Evolutionary Computation, *Proceedings of the 1999 Congress on Evolutionary Computation*, **1** (1999) 722–728.
18. Murphy, P.M., Aha, D.W.: *UCI Repository of Machine Learning Datasets (machine-readable data repository).* University of California-Irvine, Department of Information and Computer Science, (1994).
19. Walter, A.T.: Genetic Programming for Feature Discovery and Image Discrimination. In *Proceedings of the Fifth Conference on Genetic Algorithms*, (1993) 303–309.
20. Gama, J.: Local Cascade Generalization. In *Proceedings of the Fifth International Conference (ICML'98)*, (1998) 206–214.