Combining Locally Trained Neural Networks by Introducing a Reject Class

Suk-Joon Kim
Artificial Intelligence Lab (SCAI)
Dept. of Computer Engineering
Seoul National University
Seoul 151-742, Korea
sjkim@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
Dept. of Computer Engineering
Seoul National University
Seoul 151-742, Korea
btzhang@scai.snu.ac.kr

Abstract

This paper presents a new strategy for building and combining a local committee when a dataset is given. Training local committees is performed in two stages: active data partitioning and recombination by introducing an additional reject class. Active data partitioning is a preprocessing step that partitions the given dataset into several similar subsets using active learning. Additional reject class in this strategy plays an important role in assigning a focused area to each individual network of the committee. For combining the outputs of each individual network, we use a kind of sum rule criteria, assuming that the outputs of the individuals are equivalent to a posteriori Bayesian probabilities. All the learning procedures are based on the active learning paradigm. Experiments are performed on the two real-world datasets from the UCI machine learning database. The results show that the active data partitioning and recombining strategy is very successful for building a local committee and the combined result outperforms other algorithms, but the combined result can be affected by the training error level ϵ.

1. Introduction

Committee machines provide a method for improving classifier's generalization performance. One of the main motivations for building committee machines is the bias-variance dilemma [3, 13]. Let $F_i(\mathbf{x}, D)$ be the actual response of network i then, the committee output is $F(\mathbf{x}, D) = \frac{1}{M}\Sigma_{i=1}^{M} F_i(\mathbf{x}, D)$, where M is the number of networks and $\mathbf{x} \in R^n$ is the input pattern and D is training set. Assuming that d is the desired output, the expected mean squared error of the combined system can be written in terms of the individual network output [11]:

$$
\begin{aligned}
&E_D[(E[d|\mathrm{x}] - F(\mathrm{x}, \mathrm{D}))^2] \\
&\quad = (E_D[F(\mathrm{x}, \mathrm{D})] - E[d|\mathrm{x}])^2 \\
&\quad + E_D[\frac{1}{M^2}\sum_{i=1}^{M}(F_i(\mathrm{x}, \mathrm{D}) - E_D[F_i(\mathrm{x}, \mathrm{D})])^2] \\
&\quad + E_D[\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j\neq i}(F_i(\mathrm{x}, \mathrm{D}) - E_D[F_i(\mathrm{x}, \mathrm{D})]) \\
&\quad (F_j(\mathrm{x}, \mathrm{D}) - E_D[F_j(\mathrm{x}, \mathrm{D})])],
\end{aligned}
$$

where the first term is the bias of the combined system, the second and third terms are variance and covariance of the outputs of the individual networks, respectively. Since the bias and variance tradeoff problem is common to all learning algorithms, in view of the committee machine, the covariance comes to be a crucial factor for the generalization performance of the committee. This implies that the best generalization performance can be achieved by making each individual have no correlations or even negative correlations. It is shown that when each individuals of a committee have local characteristics, called 'local experts', each individuals comes to be uncorrelated or even negatively correlated for each other [1].

Two main issues on committee machines with local experts are how to make each individual network to have local characteristics and how to combine the outputs of the locally trained networks. Common approaches for building local committees estimate two parameters: standing point $\mathbf{V}$, and mapping function F. In the case where learning is *coupled*, there is a single cost function for supervised training of both sets of parameters. In the *uncoupled* case, training the position $\mathbf{V}$ is an unsupervised process which precedes the separate, supervised training of the mapping function F. For combining the individuals' outputs, two kinds of schemes have

0-7803-5529-6/99/$10.00 ©1999 IEEE

been used. In a *competitive* scheme, the architecture is designed such that the final output is equal to the output of one of the mappings. In a *cooperative* scheme, there is no such requirement and the final output is a blend of the outputs of separate mappings. All these schemes are essentially very sensitive to the estimation of $\mathbf{V}$. Table 1 summarizes the previous approaches for building a local committee machine [1].

In this paper, we propose a simple strategy for building a local committee without estimating $\mathbf{V}$. Training local experts are performed in two stages: active data partitioning and data recombination via introducing additional reject class. In the first stage, the given dataset is partitioned according to its characteristics through active learning neural networks. In the second stage, the partitioned subsets are recombined to be fed into each expert as a training dataset, where the target class of examples in the unfocused partitions are reassigned to the additional reject class. For combining the outputs of experts, we adopt a kind of cooperative strategy under the assumption that the outputs of the individuals are equivalent to posterior Bayesian probabilities. This method is applied to the problems from the UCI machine learning database [18]. Experimental results show that this method makes each experts have local characteristics, and when fully trained, the combination strategy provides better performances compared to previous results.

The organization of this paper is as follows. Section 2 describes a procedure for building local committees when a dataset is given. Section 3 describes the procedures for combining the outputs of local experts, which is based on the method by Kittler *et al.*[9]. Section 4 reports experimental results. Section 5 draws conclusions and discusses future work.

2. Training Local Experts

Table 1: Comparison of previous approaches according to the learning strategy and the type of combining the local mappings.

Methods	Training	Combining
Hampshire and Waibel [4]	uncoupled	cooperative
Stokbro et al. [15]	uncoupled	cooperative
Jacobs et al. [6]	coupled	competitive
Bottou and Vapnik [2]	uncoupled	competitive
Martinetz et al. [10]	uncoupled	competitive
Murray-Smith [12]	uncoupled	cooperative

Active learning is a learning method where the learner has control over the training data, *i.e.* it selects its own training data from its environment. The primary concern in active learning is how to choose a new example $\mathbf{x}$ to get more information from its environment.

In previous work [8], we proposed two kinds of data selection measures. One of them is to select the most critical examples incrementally under the current network. The most critical example is defined as an example causing maximum error for the current trained network (W, A).

$$m^* = \underset{m \in C}{\operatorname{argmax}}(e_m(s))$$

$$e_m(s) = \frac{1}{\dim(y_m)} \|y_m - f(x_m; W, A)\|$$

, where (x_m, y_m) is the mth training pattern, f is the output of the network with weights W and architecture A, s is the number of selection, and C is the candidate set. We showed that the resulting subsets approximate well the global distribution of the whole data. We called this selection measure as *critical data selection* (CDS). The other method is to select the least critical example incrementally under the current networks,

$$m^* = \underset{m \in C}{\operatorname{argmin}}(e_m(s)).$$

This method prefers to select examples that are similar to the already trained patterns, thus the selected subset shows the distribution in a local area. We called this selection measure as *redundant data selection* (RDS). In this paper we use the redundant data selection measure for partitioning the given dataset.

The algorithm for training local experts is described as follows.

0: Let C denote the candidate set and T the training set.

1: Move the selected examples from C into T using the RDS measure via an active learning neural network.

2: Partition the accumulated examples in T sequentially such that $\bigcup_i T_i = T$, and $T_i \cup T_j = \emptyset$ for $\forall i \neq j$,where T_i is the ith subset of T.

3: $\tilde{T}_i \leftarrow T_i \cup T_i^*$, where T_i^* is the transformed subsets in $\bigcup_{j \neq i} T_j$ such that

$$T_i^* = \{(\mathbf{x}, y) : y \leftarrow \mathrm{R}_c \ , \ \mathbf{x} \in \bigcup_{j \neq i} T_j\},$$

where R_c is the extra reject class.

4: Train ith expert with $\tilde{T}_i$ using CDS until the number of selection reaches to N_{max}.

When training each expert, the number of the hidden units is set large enough to reach the predefined error level ϵ. In the experiments, the total number of the training examples for ith expert , N_{max}, is limited to $2T_i$. This restriction allows the learning process to be accelerated as shown in [17].

3. Combining the Outputs of Local Experts

MLPs are known to be asymptotically equivalent to the optimal Bayesian classifier, given that sufficiently many hidden units are available and that training uses sufficient amounts of data [14, 5]. We use a multilayer neural network as member networks of a committee. The MLP has d input nodes and n $(= c + 1)$ output nodes where n nodes include c real classes and 1 reject class. The outputs of MLP with n output nodes are $y_i(\mathbf{x}) = (y_{i,1}(\mathbf{x}), ..., y_{i,c}(\mathbf{x}), y_{i,c+1}(\mathbf{x}))$ given input $\mathbf{x}$, When training the MLP, we allows enough hidden nodes for the given dataset. Thus following [14, 5], we assume that the output y_n converges to the empirical posterior probability $p(C_k|\mathbf{x})$.

Under this assumption, the combination rule for a local committee can be derived. The derived rule is a kind of a sum rule criterion [9] where the final output is produced by cooperation of each members. Assuming that m experts are trained with n output nodes. The classification is performed as follows:

assign $y(\mathbf{x}) \rightarrow C_j$ if

$$(1-m)p(C_j) + \sum_{i=1}^{m} p(C_j|y_{i,j}(\mathbf{x})) =$$

$$\max_k[((1-m)p(C_k) + \sum_{i=1}^{m} p(C_k|y_{i,k}(\mathbf{x})))\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x}))],$$

where k runs over 1 to $n-1$, and $y_{i,k}(\mathbf{x})$ means the probability that i-th expert classifies the given example, $\mathbf{x}$, into class k. Note that the possible number of classes for final decision is $n-1$, not n. For a given test example $\mathbf{x}$, most individuals are apt to estimate high probability to the reject class, R_c, which is assigned to the n-th node. This means that most of the individuals ignore the test example $\mathbf{x}$. Thus the combination of the individuals under any kind of sum rule criteria assigns the highest probability to the reject class, R_c, which is meaningless. The rule can be derived as follows.

Assuming that the individual networks are independent of each other, Bayes theorem can be used to show the posterior probability that the committee classifies the given example $\mathbf{x}$ into class k ,when the each networks' output, $y_{i,k}(\mathbf{x})$ is given.

$$p(C_k|y_{1,k}(\mathbf{x}),y_{2,k}(\mathbf{x}),...,y_{m,k}(\mathbf{x})) = \frac{p(C_k)\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})|C_k)}{\sum_{j=1}^{n-1} p(C_j)\prod_{i=1}^{m} p(y_{i,j}(\mathbf{x})|C_j)},$$

where the denominator is constant as prior knowledge, thus

$$\begin{aligned}
&p(C_k)\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})|C_k) \\
&= \max_k p(C_k)\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})|C_k) \\
&= \max_k p(C_k)\prod_{i=1}^{m} \frac{p(C_k|y_{i,k}(\mathbf{x}))p(y_{i,k}(\mathbf{x}))}{p(C_k)} \\
&= \max_k p^{1-m}(C_k)\prod_{i=1}^{m} p(C_k|y_{i,k}(\mathbf{x}))p(y_{i,k}(\mathbf{x})) \\
&= \max_k p^{1-m}(C_k)\prod_{i=1}^{m} p(C_k|y_{i,k}(\mathbf{x}))\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x}))
\end{aligned}$$

,where k runs over 1 to $n-1$. If we assume that posterior probabilities, $p(C_k|y_{i,k}(\mathbf{x}))$ computed by the respective experts will not deviate dramatically from the prior probabilities, the posterior probability can be described as

$$p(C_k|y_{i,k}(\mathbf{x})) = p(C_k)(1+\delta_{ki}), \text{ where } \delta_{ki} \ll 1.$$

Using this, we have

$$\begin{aligned}
&p(C_k)\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})|C_k) \\
&= \max_k p^{1-m}(C_k)\prod_{i=1}^{m} p(C_k)(1+\delta_{ki})\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})) \\
&= \max_k p(C_k)(1+\sum_i \delta_{ki} + \sum_{h,i} \delta_{kh}\delta_{ki} + ...)\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})) \\
&\cong \max_k (p(C_k) + p(C_k)\sum_i \delta_{ki})\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})) \\
&= \max_k (p(C_k) - mp(C_k) + p(C_k)\sum_{i=1}^{m}(1+\delta_{ki}))\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})) \\
&= \max_k ((1-m)p(C_k) + \sum_{i=1}^{m} p(C_k)(1+\delta_{ki}))\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x})) \\
&= \max_k ((1-m)p(C_k) + \sum_{i=1}^{m} p(C_k|y_{i,k}(\mathbf{x})))\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x}))
\end{aligned}$$

Table 2: Comparison of various algorithms in terms of generalization performance on the two problem domains. The performance values for other methods are from Yao [16].

diabetes problem					
Algorithm	CDS	RBF	BP	CART	EPNet
Error Rate	75.60	76.7	74.2	74.5	76.5
credit card problem					
Algorithm	CDS	RBF	BP	CART	EPNet
Error Rate	86.96	85.5	84.6	85.5	88.5

Table 3: Classification rate of the Combined Local Committee with varying training error level ϵ

	$\epsilon = 0.00$	$\epsilon = 0.05$	$\epsilon = 0.10$
Diabetes	78.33	80.56	76.39
Credit Card	66.24	91.62	88.83

Although the above assumption seems to be rather restrictive, the experimental results below show that the derived combination rule produces a superior consensus of the locally trained individuals. If the probability that the i-th classifier classifies a given input $\mathbf{x}$ to the class k is uniformly distributed, and then the term $\prod_{i=1}^{m} p(y_{i,k}(\mathbf{x}))$ can be disregarded, the derived sum rule will have a similar form with the results of Kittler *et al.* [9]. In this paper, the parameter value of $p(y_{i,k}(\mathbf{x}))$ is considered as the prior probability that the given input $\mathbf{x}$ is assigned to the class k.

4. Experimental Results

We performed experiments on two real-world data sets. One is the australian credit card assessement data set, the other is the diabetes data set. The australian credit card assessment problem contains 690 cases in total. The output has two classes. The 14 attributes include six numeric values and eight discrete ones. The diabetes data set contains 500 examples of class 1 and 268 of class 2. Each example contains eight attributes. Table 2 lists the reported classification performances of the various learning algorithms for these two problems. The values in the table is mainly cited from [16].

Three experiments were run for each data set varying the training error level, $\epsilon = 0.0, 0.05, 0.10$. The experimental results are shown in Table 3. In both of the cases, the best generalization performance was achieved when $\epsilon = 0.05$. This indicates that too tight or too loose training can fail to generalize the global distri-

Table 4: Sum of the individual's classification rate with varying training error level ϵ

	$\epsilon = 0.00$	$\epsilon = 0.05$	$\epsilon = 0.10$
Diabetes	125.25	134.34	96.08
Credit Card	109.33	102.75	93.82

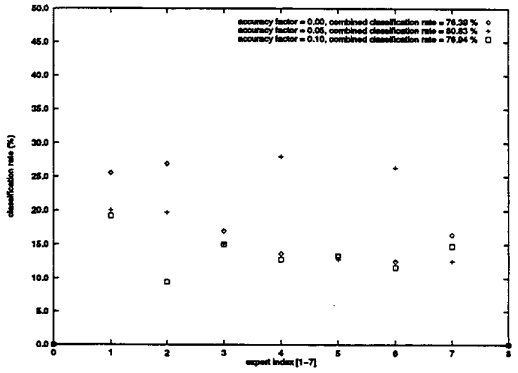

Figure 1: Generalization performance of each expert for the diabetes card problem.

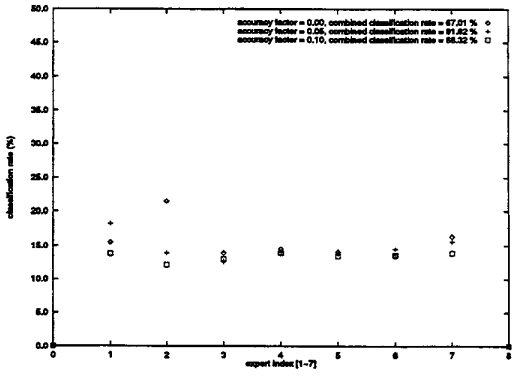

Figure 2: Generalization performance of each expert for the credit card problem.

bution well. Figure 2 and 1 show the generalization performance rate of each individuals on both problems. As shown in these two figures, each generalization performance is less than 30 %. Table 4 shows the total sum of classification rates of individual networks. From Table 4 and Figure 2, 1, we see that the highest total classification rate is not sufficient condition for the best performances. Rather, in the case of credit card problem, $\epsilon = 0.00$ shows the highest total classification rate, but the combined classification rate shows the worst performance among the algorithms. The fact that the best combined performance is achieved when $\epsilon = 0.05$ should be a lucky guess. But by now we can conclude that the training error level ϵ is relevant to the generalization performance with a chance that there can be other factors for determining the final generalization performance.

5. Conclusion

It has been proved that local committees are better than simple committee machines for improving generalization performance theoretically and experimentally [3, 13]. This can be explained by the 'bias-variance-covariance analysis' [11]. But the problems of how to training a local committee and how to combine the locally trained individual networks have been unclear. In this paper, we presented a simple strategy to build a local committee when a data set is given. This scheme does not require the estimation of the standing point $\mathbf{V}$ which is a critical factor for most other approaches. We showed that the active data partitioning and recombination strategy makes each individual network have local characteristics and that the sum rule criterion produces superior combined results. As shown in Table 3, the combined generalization performance is much dependent on the training error level ϵ. Future works include finding an optimal error level ϵ for training each individual networks in the local committee and finding another relevant factors.

Acknowledgements: This research was supported in part by the Korea Ministry of Science and Technology though KISTEP under Grant BR-2-1-G-06.

References

[1] E. Alpaydin, and M.I. Jordan, "Local linear perceptrons for classification," *IEEE Transactions on Neural Networks*, vol. 7, no. 3, pp. 788-792, 1996.

[2] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Computation*, Vol. 4., pp. 888-900, 1992.

[3] R.T. Clemen and R.L. Winkler, "Limits for the precision and value of information from dependent sources," *Operations Research*, vol.33, pp.427-442, 1985.

[4] J.B. Hampshire, and A. Waibel, "Connectionist architectures for multi-speaker phoneme recognition," *Advances in Neural Information Processing Systems*, 2, pp. 203-210, 1990.

[5] J.B. Hampshire and B.A. Pearlmutter, "Equivalence proofs for multilayer perceptron classifiers and the Bayesian discriminant function," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep., 1994.

[6] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79-87, 1991.

[7] R.A. Jacobs, "Bias/variance analysis of mixture-of-experts architectures," *Neural Computation*, vol. 9, pp. 363-383, 1997.

[8] S.J. Kim, "Active Data Partitioning for Building Mixture Models," *Proceedings of The International Conference of Neural Information Processing (ICONIP'98)*, vol. 2, pp. 854-857, 1998.

[9] J. Kittler, M. Hatef, R.P.W Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, 1998.

[10] T.M. Martinetz, S.G. Berkovich, and K.J. Schulten, "Neural-Gas network for vector quantization and its application to time-series prediction," *IEEE Transactions on Neural Networks*, vol. 4, pp. 558-569, 1993.

[11] R. Meir, "Bias, variance, and the combination of least squares estimators," *Advances in Neural Information Processing System*, 7, pp. 295-302, 1995.

[12] R. Murray-Smith, "A local model network approach to nonlinear modelling," PhD Thesis, Department of Computer Science, University of Strathclyde, 1994.

[13] M. Perrone and L.N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," *Neural Networks for Speech and Image Processing*, R.J. Mammone, Ed., Chapmann-Hall, 1993.

[14] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, and B.W. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminat function," *IEEE Transactions on Neural Networks*, vol. 1, pp. 296-298, 1990.

[15] K. Stokbro, D.K. Umberger, and J.A. Hertz, "Exploiting neurons with localized receptive fields to learn chaos," *Complex Systems*, 4, pp. 603-622, 1990.

[16] X. Yao, and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 694-713, 1997.

[17] B.-T. Zhang, "Accelerated learning by active example selection," *International Journal of Neural Systems*, vol. 5, no. 1, pp. 67-75, 1994.

[18] UCI machine learning database: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/