Classification Performance of Committee Networks Improvement under Sparse Data Conditions

การเพิ่มประสิทธิภาพของแบบจำลองโครงข่ายคอมมิตตีในการแยกประเภทของข้อมูล สำหรับข้อมูลที่มีจำนวนจำกัด

Rujapa Nanthapodej (รุจาภา นันทโพธิ์เดช)* Dr.Danaipong Chetchotsak (ดร.ดนัยพงศ์ เชษฐโชติศักดิ์)**

ABSTRACT

In most real world applications, the data for modeling is normally sparse. This makes it difficult for modelers to construct a neural network model. Eventually the training process may cause overfitting. This paper proposes committee network methodology to deal with sparse data for a classification problem. The committees are developed based on bootstrapped training sets and are called adjusted pair-wise committee and adjusted random-mix committee. We test the committees' performance against that of the bootstrap committee and single neural network using the selected data sets from UCI Machine Learning Repository, Center for Machine Learning and Intelligent System. The results reveal that the proposed models perform as well as or better than the baseline models.

บทคัดย่อ

ในกรณีที่ข้อมูลมีจำนวนจำกัด การกำหนดค่าพารามิเตอร์ที่เหมาะสมในการสร้างแบบจำลองสำหรับแบ่งกลุ่มข้อมูลนั้นทำได้ค่อนข้างยาก ทำให้แบบจำลองที่ได้เกิดการจดจำข้อมูล ไม่เกิดการเรียนรู้ ผลลัพธ์ที่ได้มีความผิดพลาด และไม่เหมาะสมต่อการนำไปประยุกต์ใช้ ดังนั้นงานวิจัยนี้จึงเสนอแนวทางการสร้างแบบจำลองโครงข่ายคอมมิตตีสำหรับการแบ่งกลุ่มข้อมูล ด้วยวิธีอะจัสทิทแพร์ไวส์และวิธีอะจัสทิทแรนดอมมิกส์ ซึ่งเป็นแบบจำลองที่ประกอบด้วยโครงข่ายประสาทเทียมหลากหลายโครงข่าย และใช้วิธีบูทสแทรปเพื่อเตรียมชุดข้อมูลในการเรียนรู้ของแบบจำลอง ผลที่ได้พบว่าแบบจำลองที่นำเสนอมีประสิทธิภาพในการแบ่งกลุ่มข้อมูลได้เทียบเท่าหรือสูงกว่าแบบจำลองบูทสแทรปคอมมิตตีและแบบจำลองโครงข่ายประสาทเทียมเดี่ยว

Key Words : Committee network, Sparse data, Classifications

คำสำคัญ : โครงข่ายคอมมิตตี ข้อมูลจำนวนจำกัด การแยกประเภทข้อมูล

** Student, Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University*

*** Assistant Professor, Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University*

Introduction

Under sparse data conditions where most problem space is usually unknown, constructing neural network models is very difficult. This is because the amount of data is not enough to determine neural networks' topologies, e.g., number of hidden units, number of learning cycles, and so on. In this matter, a particular neural network usually learns the available data too well but fails to generalize to the "unseen" data. Such occurrence is referred to as "overfitting".

There have been a number of published papers attempting to improve neural networks' performance under sparse data conditions. Many of those use the committee network approach (Parmanto *et al.*, 1996; Lam, 1999; Siriphala, 2000; Chetchotsak and Twomey, 2007). Here, the most promising but simple approaches are proposed by Siriphala (2000). To be specific, the committee networks in his work are constructed based on one of the resampling method known as the bootstrap. The rationale of using the bootstrap method is to encourage each neural network in the committee to learn different parts of data and thus they all would have different expertises. If all of them make a mistake, they would make a mistake at different places, and eventually the errors would be cancelled out. Such an approach is known as the error decorrelation (Krogh and Vedelsby, 1995). In addition to the use of the bootstrap method to diversify each network in the committee, Siriphala (2000)'s work also promotes and escalates even more diversity among the networks in the committee. His approaches are named as "pair-wise" and "random-mix" algorithms and are to deal with function approximation problems. However, there is no report on performance of the "pair-wise" and "random-mix" algorithms when applied to a classification problem.

The objective of this paper is to report our attempt to improve classification performance of committee networks under sparse data conditions. Our proposed methods are developed based on the "pair-wise" and "random-mix" algorithms. Then the proposed methods are evaluated using a simulation through different selected classification problems. These methods may be applied to any classification problems such as product classification, yield improvement, and so on.

Background

Committee networks are constructed based on the concept that "many heads are better than one". They consist of several neural networks called committee members. According to Figure 1, each neural network learns the data and helps one another to solve or predict the same problem. Each neural network's output is combined through a fusion rule to produce a committee output. The most common fuser is known as the majority voting scheme.

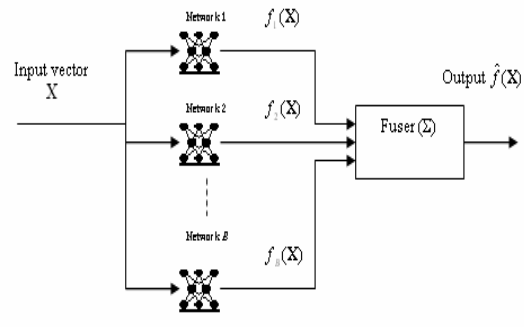


Figure 1 Block diagram of a committee network (Chetchotsak, 2004)

Several papers have proposed a method to train a committee network. Many of those use the error decorrelation approach. In particular, each network in the committee is trained using different parts of data so as to have different expertises. When those networks make a mistake, they would make a mistake at different places and eventually each mistake from the committee member would be cancelled out. The most well known method to decorrelate the training data is the bootstrap method (Table 1).

Adjusted pair-wise committee (APW)

APW consists of seven network architectures[1] and is based on the bootstrap algorithm. The use of seven architectures attempts to escalate the degree of diversity among the committee members so as to encourage each network to have different expertise and help each other to predict the output. Figure 2 depicts the diagram of APW algorithm. In this case, the fusion rule (majority voting scheme) is used at two layers. In the first layer, the fusion rule combines the output of the neural networks with the same architecture but different bootstrapped training sets. In the second layer, the majority voting scheme combines the final output to produce the committee output. This decision mechanism helps to fitter out the decisions made by each committee member. The APW algorithm is presented in Table 2.

[1] Such a number is a proposed number in this experiment. According to Parmanto *et al.* (1996), Siriphala (2000), Chetchotsak and Twomey (2007) that the number of committee members should be more than 20. For a classification problem, the number of committee members should be an odd number for consensus of decision.

Table 1 The bootstrap algorithm.

Step i)	Let $\hat{F}$ be the empirical probability distribution where T with n observation is drawn.
Step ii)	Let $t_1, t_{2,...}, t_n$ be a collection of training set T where $t_i = (x_i, y_i)$.
Step iii)	Specify the number of bootstrap samples, B to produces $T^{*1}, T^{*2}, ..., T^{*B}$.
Step iv)	Randomly choose t_i from T for $i = 1,...,n$ with replacement and equal probability mass $\frac{1}{n}$ to produces each T^{*i}.
Step v)	Repeat *Step iv)* B times to produce $T^{*1}, T^{*2}, ..., T^{*B}$.
Step vi)	Train Network 1, Network 2,…, Networks B using $T^{*1}, T^{*2}, ..., T^{*B}$.
Step vii)	Given the input x_i for $i = 1,...,n, \hat{f}(T^{*1}, x_i), ..., \hat{f}_B(T^{*B}, x_i)$ are the outputs of these networks.
Step viii)	Majority voting is used as a fuser of this committee network, as shown in Figure 1.

Adjusted random-mix committee (ARM)

ARM also consists of seven network topologies. In this case, all the committee members' outputs are combined through only one fuser. This encourages each committee member to have the same rank to vote for the output. Table 3 shows the ARM algorithm.

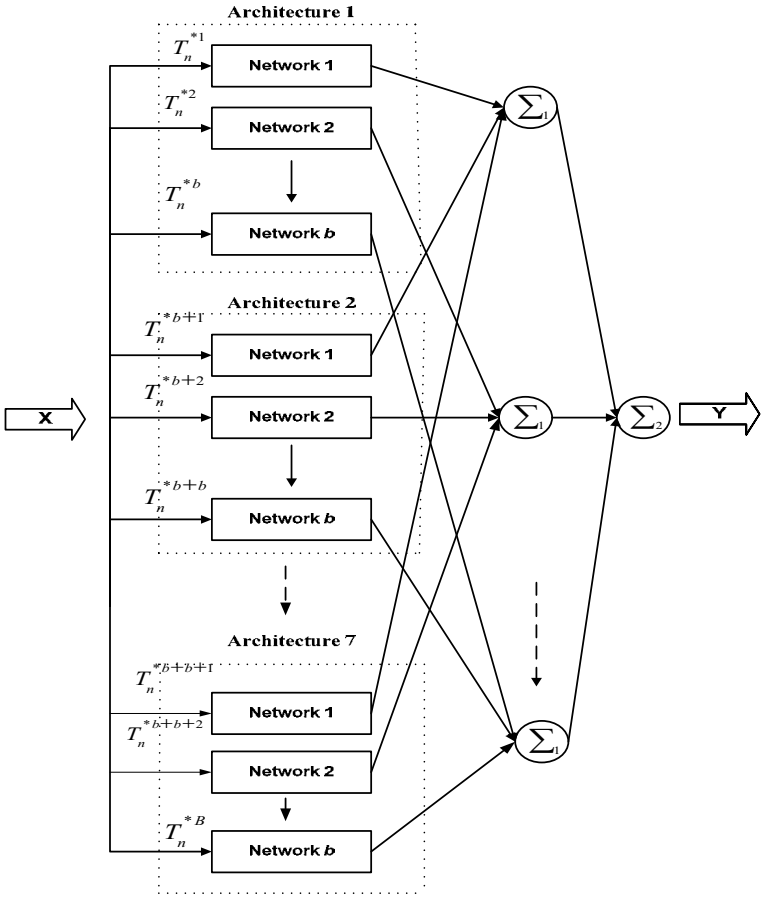


Figure 2 Adjusted pair-wise committee.

Table 2 Adjusted pair-wise algorithm

Step i)	Let $T_n^{*1}, T_n^{*2}, ..., T_n^{*B}$ represent the bootstrap samples of n observations randomly generated, where B is the number of bootstrap samples and total number of APW members.
Step ii)	Train neural networks using the bootstrap samples in *Step i)* with 7 different architectures. Thus, the total number of neural networks with the same architecture is $N = B/7$.
Step iii)	Use the majority voting scheme as a fuser to produce the committee's output as described in Figure 2.

Table 3 Adjusted random-mix algorithm

Step i)	Let $T_n^{*1}, T_n^{*2}, ..., T_n^{*B}$ represent the bootstrap samples of n observations randomly generated, where B is the number of bootstrap samples and total number of ARM members.
Step ii)	Train neural networks using the bootstrap samples in *Step i)* with 7 different architectures. Thus, the total number of neural networks with the same architecture is $N = B/7$.
Step iii)	Use the majority voting scheme as a fuser to produce the committee's output as described in Figure 3.

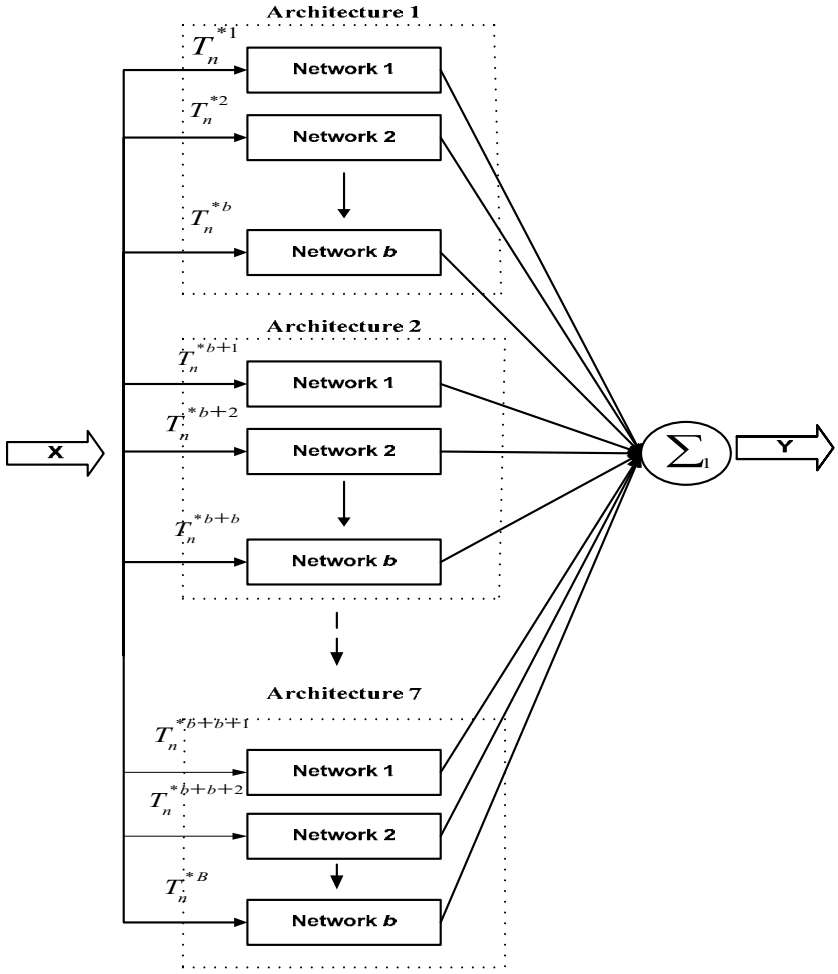


Figure 3 Adjusted random-mix committee.

Method

Performance of APW and ARM is evaluated through an empirical study (computer simulation) which includes three data problems. The training and test data sets are selected from the synthetic data sets used by Parmanto *et al.* (1996) and the UCI Machine Learning Repository, Center for Machine Learning and Intelligent System. To determine sparse data sets for training, we apply the rule of thumb used by Chetchotsak and Twomey (2007) in this experiment; that is the lower bound of a sparse sample size is equal to five times the number of input variables (attributes). Described below are the data sets used in this experiment.

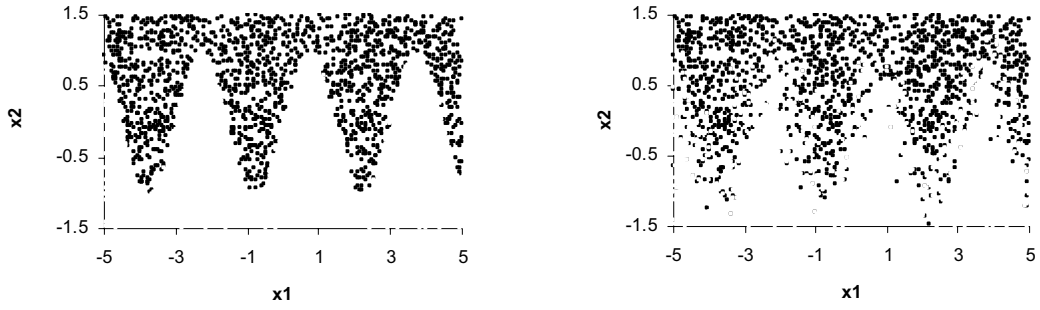
The sine wave problem

In this problem, the task is to separate the data set into two regions according to the sine curve. The functional form of the data is written as

$$y = \begin{cases} 1, & \text{if } x_2 + z_1 \geq \sin(\frac{2\pi}{3}(x_1 + z_2)) \\ 0, & \text{if } x_2 + z_1 < \sin(\frac{2\pi}{3}(x_1 + z_2)) \end{cases}, \quad (1)$$

where z_1 and z_2 - $N(0,\sigma)$. In this experiment, we use two levels of noise to test the committee's performance: first noise-free and second noisy

(σ=0.3) as shown in Figure 4. Each training set consists of 10 records (from the rule of thumb). Then another separated data set of 3,000 records is used as a test set to evaluate the classification performance.



(a) noise-free (b) noisy

Figure 4 Sine wave data sets with different levels of noise

Pima Indians diabetes problem

In this problem, we are to diagnose whether the patients show signs of diabetes, given that the patients' clinical and history records such as diastolic blood pressure, 2-hr serum insulin, and so on are known. Thus, the prediction output is either positive (patients show signs of diabetes) or negative (patients show no sign of diabetes). Totally, the number of instances is 768 with 8 attributes. From the rule of thumb, the size of sparse training data in this experiment is set to be 5 x 8 = 40 instances.

BUPA liver disorders problem

In this problem, we are to classify patients who have and do not have liver disease. The data set consist of six factors that may contribute to liver disease and has 345 records. Hence, the classification output is either positive (patients have liver disease) or negative (patients not have liver disease). In this experiment, the sparse training set size is set to be 6 x 5 = 30 records.

Experimental design

Performance of APW and ARM is evaluated under varying factors of network complexity levels and number of learning cycles through a Monte Carlo simulation. Table 4 summarizes the experimental design. This helps to investigate how each algorithm performs under various conditions.

Network complexity levels: We use the number of hidden units to represent levels of complexity. Here SN stands for the single neural network while BTC is for the bootstrap committee. In Table 4 APW and ARM have 5, 10, 15, 20, 25, 30, and 35 hidden units. The subscripts "min", "avg", and "max" symbolize the minimum, average, and maximum of the set {5, 10, 15, 20, 25, 30, 35}, respectively.

Number of learning cycles: This factor is important for network construction. Choosing the number of learning cycles to be too large may lead to overfitting. In this experiment we will test how each method performs when trained using different numbers of learning cycles.

Table 4 Experimental design

Factors	Levels
1. Network complexity (number of hidden units)	
Single neural networks (SNs) : SNmin	1. Minimum (5 hidden units)
: SNavg	2. Average (20 hidden units)
: SNmax	3. Maximum (35 hidden units)
Bootstrap committees (BTCs) : BTCmin	1. Minimum (5 hidden units)
: BTCavg	2. Average (20 hidden units)
: BTCmax	3. Maximum (35 hidden units)
Adjusted pair-wise committee : APW	(5, 10, 15, 20, 25, 30, 35 hidden units)
Adjusted random-mix committee : ARM	(5, 10, 15, 20, 25, 30, 35 hidden units)
2. Number of learning cycles	1. 20,000 cycles
	2. 50,000 cycles
	3. 150,000 cycles

Here, the single neural network and bootstrap committee are used as a baseline method to compare against APW and ARM. Moreover, this experiment is replicated for say 20 times to remove dependency on sampling of training data. This is done by re-sampling the training data sets of each problem 20 times and each time all the networks are trained and tested according to the experimental design. Furthermore, performance of each method is evaluated using a set of the remaining data, exclusively separated from the training set.

Classification model construction

All the committee models in this experiment are constructed based on the bootstrapped data. The number of bootstrap networks or committee members is chosen to be 21. This number is chosen based on the experimental results in Parmanto et al. (1996), Lam (1999), and Siriphala (2000) and the number must be an odd number in order to make a consensus decision for the majority voting scheme. In this experiment, all the neural networks are multilayer perceptrons trained with the backpropagation algorithm and the sigmoid is used as an activation function.

Performance assessments

Performance measures are defined in eqn (2)-(4). Percentage of classification errors (error rate) measures how well a particular classification model is in classifying the data. Fault positive rate reflects how well the model classifies the data labeled as "positive". Finally, the fault negative rate indicates the model's accuracy in classifying the data labeled as "negative". In this matter, a model that has a small error rate does not necessarily have small fault positive or fault negative rates. This is especially true when the ratios between records labeled by "positive" and "negative" are quite different. A model that is said to be robust should have small rates in all measures.

$$\%\text{Accuracy} = \frac{\text{number of correct outputs}}{\text{number of total samples}} \times 100 \quad (2)$$

$$\%\text{ Fault Positive} = \frac{\text{number of correct outputs} \mid \text{the outputs are positive}}{\text{number of tatal samples} \mid \text{the targets are positive}} \times 100 \quad (3)$$

$$\%\text{ Fault Negative} = \frac{\text{number of corect outputs} \mid \text{the outputs are negative}}{\text{number of total samples} \mid \text{the targets are negative}} \times 100 \quad (4)$$

Results

Experimental results demonstrate each model performance at different conditions and classification problems. The followings present results in the form of a 95% confidence interval (C.I.) of the performance measures described above.

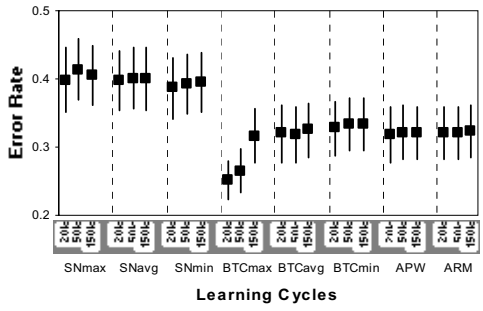
The sine wave problem

Simulation results for the noise-free data are shown in Figure 5 and the results for the noisy data are depicted in Figure 6. In both data cases, all the committee networks outperform all SN types. For the clean data set, using high network complexity or a large number of learning cycles does not help to improve SN's performance. For the committee types, BTC_{max} shows the best performance only at small learning cycles. When the data is noisy, performance difference among the algorithms is quite clear. All SN types perform much worse than the committee. In this case, increasing the number of hidden units does not improve the algorithms' performance. It would rather degrades the classification ability. Performance difference is quite clear in the noisy data case. Here, the committee types perform much better than the single neural network. Among the committee types, BTC_{max} is the best, regarding all measures. BTC_{max} shows the best performance when they trained using 20,000 cycles.

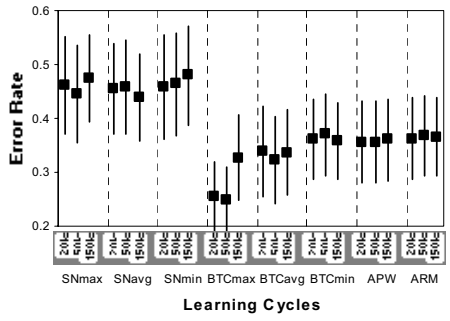
The Pima Indian diabetes problem

The Pima Indian diabetes problem's results are shown in Figure 7. In general, the committee types perform better than all the single neural networks for all measures at all conditions. Among the committee group, BTC_{max} seems to show less performance variation. Other committee networks' performance fluctuates along the learning cycles. The result also shows that using either large number of hidden units or large number of learning cycles does not improve the algorithms' performance.

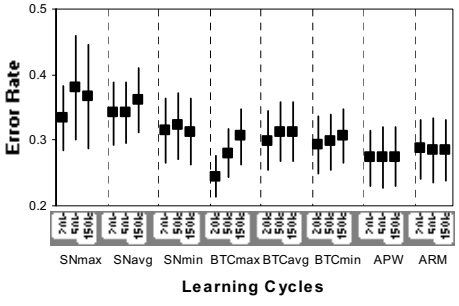
In this data set, it can be noticed that the rate of fault negative is greater than the rate of fault positive. This implies that the number of records labeled as "negative" (patients show no sign of diabetes) is much smaller than that labeled by "positive" (patients show sign of diabetes). The data labeled as "negative" may be considered as a very sparse data set and thus percentage of fault negative would be the most appropriate performance measure in this case. According to Figure 7(c), BTC_{min} does not perform well compared to the group.



(a) % accuracy

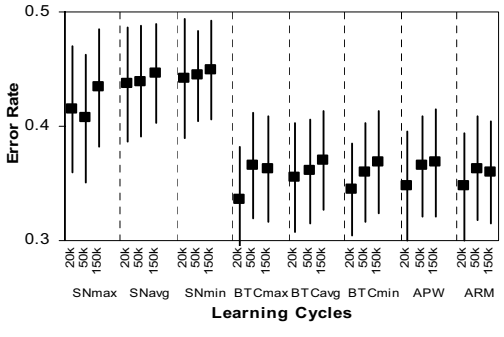


(b) % fault positive

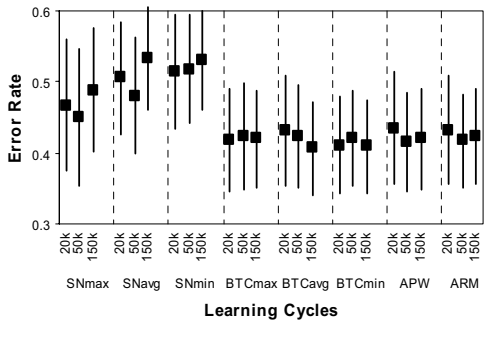


(c) %fault negative

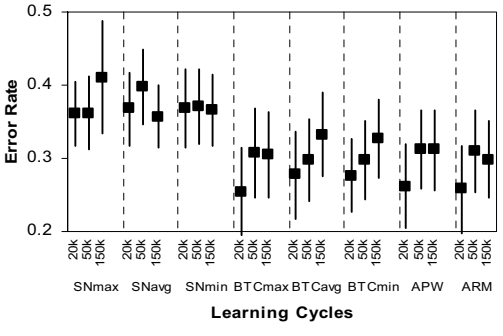
Figure 5 Simulation results for the sine wave problem: noise-free data



(a) % accuracy



(b) % fault positive



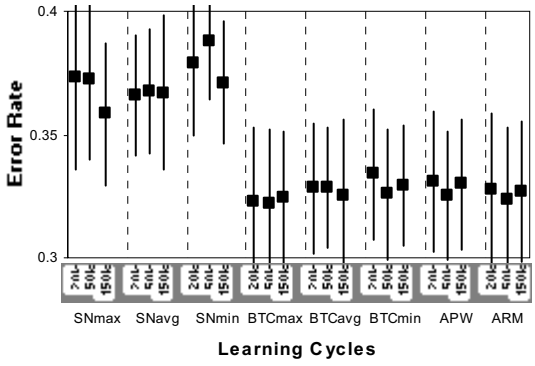
(c) %fault negative

Figure 6 Simulation results for the sine wave problem: noisy data

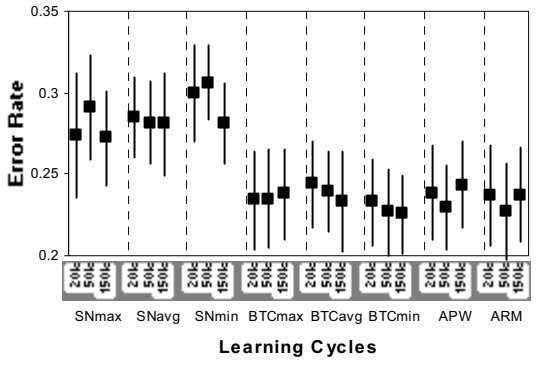
BUPA liver disorders problem

Figure 8 demonstrates the experimental results for the BUPA liver disorder data set. The figure reveals that the results for this data set and those for the Pima Indian data set follow the same trend. In particular, the committee group still outperforms all the single neural networks. Additionally, BTC_{max} shows less performance variation among the group. Like the Pima Indian diabetes problem, there is no evidence showing that a large number of hidden units or number of learning cycles helps to improve the classification performance.

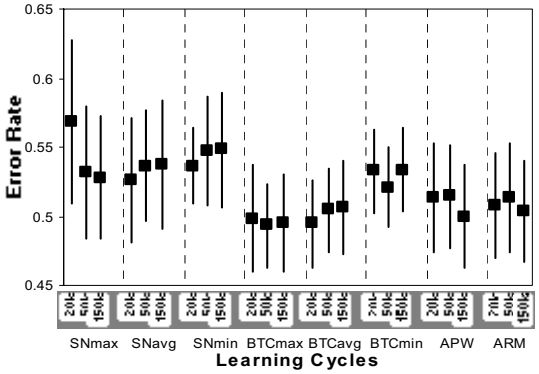
Furthermore, the rate of fault negative is generally larger than that of the fault positive, like the case of the Pima Indian diabetes problem. This also indicates that the number of negative data (patients not have liver disease) is greater than the number of positive data (patients have liver disease). Hence, the percentage of fault negative will also play an important role in monitoring classification performance. In this case, BTC_{max} and BTC_{avg} seem to perform better than other methods in the group.



(a) % accuracy

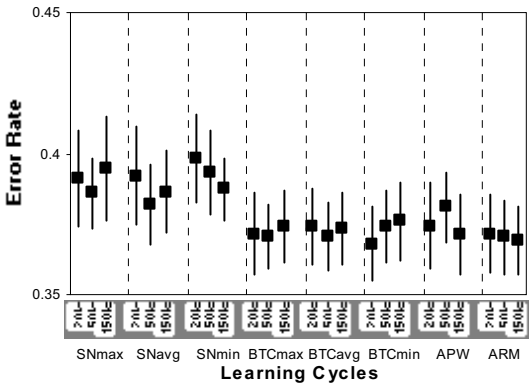


(b) % fault positive

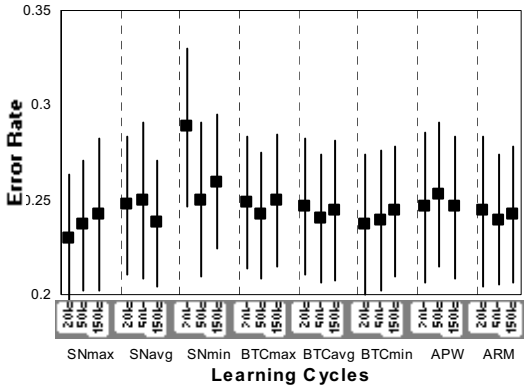


(c) %fault negative

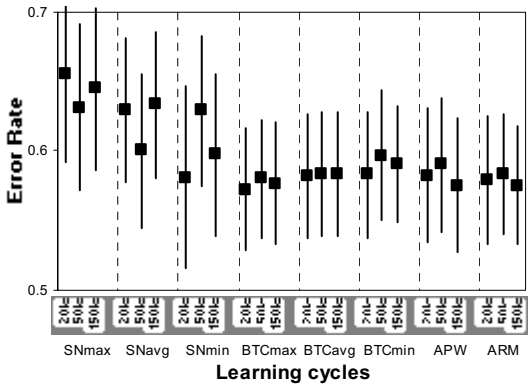
Figure 7 Simulation results for the Pima Indian diabetes problem



(a) % accuracy



(b) % fault positive



(c) %fault negative

Figure 8 Simulation results for the BUPA liver disorder problem

Discussion and Conclusion

As mentioned earlier, the rationale of using APW and ARM algorithms is to promote and escalate diversity among the networks in the committee in order to have each network help one another to predict a reasonable classification, according to the concept of the error decorrelation. The simulation results from all data problems in this paper however, appeared to contradict our hypothesis on the error decorrelation approach and even Siriphala (2000)'s results. In fact, the pair-wise and random-mix algorithms in this paper do not show any performance improvement compared to the bootstrap committees, like BTC_{max}, BTC_{min}, and BTCavg. On the other hand, the pair-wise and random-mix algorithms in Siriphala (2000)'s work outperform the simple bootstrap committee in most cases. It should be noted that Siriphala (2000) uses the pair-wise and random-mix committees with a function approximation problem while we use these algorithms with a classification problem. Such difference may be the cause of the contradicted results.

The rationale behind the APW and ARM algorithms is to encourage each neural network in the committees to have different expertise through

the use of different network architectures in addition to the use of the bootstrap algorithm. We hypothesize that if levels of diversity among the neural network increase, the total error would cancel out since each network makes a mistake at different places as mentioned before. For a function approximation problem, Siriphala (2000)'s results with the simple average as a fusion scheme confirm such a hypothesis. However, the APW and ARM algorithms have taught a lesson that increasing levels of disagreement among the neural networks may not help to reduce the error when using the APW or ARM algorithm in a classification problem. Indeed, the committees make a decision in terms of classes, e.g., "positive" or "negative", via the majority voting scheme. Here, promoting levels of disagreement among the committee members would rather confuse the committees' decision than improve the classification ability.

Our results from the simulation reveal that all the committee networks outperform the single networks in all conditions. However, the APW and ARM algorithms perform as well as or worse than the bootstrap committees. Among the group, BTC_{max} appears to be the most robust model. To be specific, BTC_{max} performs better than other methods in all conditions. Its robustness enable modelers to construct a committee models without having difficulty of choosing network topologies, particularly for the number of hidden units and learning cycles. This is true especially under sparse data conditions where the problem space is mostly unknown and network topologies are very difficult to determine.

References

Chetchotsak, D. 2004. Lecture Notes in 164785 Intelligent Computing for Industrial and Manufacturing Applications. Khon Kaen: Department of Industrial Engineering, Faculty of Engineering, Khon Kaen University.

Chetchotsak, D., and Twomey, JM. 2007. Combining Neural Networks for Function Approximation under Conditions of Sparse Data: The Biased Regression Approach. International Journal of General Systems, 36, 479-499.

Krogh, A., and Vedelsby, J. 1995. Neural Network Ensembles, Cross Validation, and Active Learning. Advance in neural Information Processing System, 7, 1-8.

Lam, SSY. 1999. Improved Prediction and Validation Using Resampled Neural Networks: Committee Networks and Hybrid Validation, Ph.D. Thesis, Department of Industrial Engineering, University of Pittsburgh.

Parmanto, B., Munro, PW., and Doyle, HR. 1996. Reducing Variance of Committee Prediction with Resampling Techniques. Connection Science, 8, 3 and 4, 405-425.

Siriphala, P. 2000. Controlling Artificial Neural Networks Overtraining when Data is Scarce, Ph.D. Thesis, Department of Industrial and Manufacturing Engineering, Wichita State University.