2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

Neural Network Rule Extraction by using the Genetic Programming and its Applications

Yoshikazu Ikeda† , Shozo Tokinaga‡ and Jianjung Lu‡

†Faculty of Economics
Shinshu University
3-1-1 Asahi, Matsumoto-shi, Nagano 390-8621 Japan
‡Graduate School of Economics
Kyushu University
6-19-1 Hakozaki, Higashi-ku, Fukuoka 812-8581 Japan
Email: ikeda@econ.shinshu-u.ac.jp, tokinaga@en.kyushu-u.ac.jp

Abstract—This paper deals with the use of neural network rule extraction techniques based on the Genetic Programming (GP). Even though, recent methods of neural networks rule extraction enable us to generate classification rules, but the comprehensive mathematical method which relate the output to the inputs using parameters are complicated. Then, in our paper, we utilized the GP to automatize the rule extraction process in the trained neural networks where the statements changed into a binary classification. At first, the pruning process of weight among neurons is applied to obtain simple but substantial binary expressions which are used as statements is classification rules. Then, the GP is applied to generate ultimate rules. As an applications, we generate rules to prediction of bankruptcy and the classification of corporate bonds (rating) by using the financial indicators.

1. Introduction

There exist methods to develop decision making and decision tables such as multivariate discriminant function, logistic regression. There are also methods categorized to classification tree such as the entropy-based decision tree (for example, ID3) and inductive learning [3]. However, most of these studies focus primarily on developing classification models with high perspective accuracy without any attention to explaining how the classifications are being made.

This paper deals with the use of neural network rule extraction techniques based on the Genetic Programming (GP) to build intelligent and explanatory evaluation systems. Though neural networks have their universal approximation property that seems to be attractive at first sight, their intrinsically black-box nature prevent them being successfully applied in various field. Fortunately, recent development in algorithms that extract rules from trained neural networks enable us to generate classification rules [1][2]. However, the comprehensive mathematical method which relate the output to the inputs using the set of weight, bias and nonlinear activation functions are hard for human to trace. Then, in our paper, we utilized the GP to automatize the rule extraction process in the trained neural networks where the decision basically boils down to a binary classification problems.

Even though the production (classification) rule generation used in these models are applicable straightforward to the underlying problems for decision making, but in the original GP method production rules include many statements described by arithmetic expressions as well as basic logical expressions [5]-[7]. The fact makes the rule generation process very complicated. Therefore, we utilize the neural network and binary classification to obtain simple and relevant classification rules in real applications. At first, the pruning process of weight among neurons is applied to obtain simple but substantial binary expressions which are used as statements is classification rules. Then, the GP is applied to generate ultimate rules.

As an applications, we generate rules to bankruptcy prediction and to classification of corporate bonds (rating) by using the financial indicators.

2. Neural Networks and Rule Extraction

2.1. Algorithm of neural networks

Neural networks are mathematical representations inspired by the functioning of the human brain. Especially, the multilayer neural network is typically composed of an input layer, one or more hidden layers, and an output layer, each consisting of several neurons. Each neuron processes its input and generates one output value which is transmitted to the neurons in the subsequent layer. All neurons and layers are arranged in a feed-forward manner, and no feedback connections are allowed. The output of the neuron i is computed by processing the weighted inputs and its bias term as follows.

In the formula, the $w_{i,j}^{n,n-1}$ denotes the weight connecting the j th unit in layer n with the i th unit in layer $n-1$. The value x_j^{n-1} is the output of jth unit in layer $n-1$. Then, the

input to i th neuron in layer n is obtained by

$$u_i^n = \sum w_{i,j}^{n,n-1} x_j^{n-1}. \quad (1)$$

The output of neurons is calculated by using the transfer function emulating threshold logic which is called sigmoid function.

The signal which is back-propagated from i th neuron in output layer N is obtained by using the difference between the output of output layer N and the prescribed observation d_i^N as follows.

$$\delta_i^N = (d_i^N - x_i^N) g(u_i^N), g(x) = f(x)(1 - f(x)). \quad (2)$$

Similarly, the signal back-propagated from the i th neuron in layer n to neurons in layer $n-1$ is given as

$$\delta_i^n = df(u_i^n)/du \sum_k \delta_k^{n+1} w_{k,i}^{n+1,n}. \quad (3)$$

In summary, the update for the weight and the threshold value is given as follows where $t, t+1$ are the time step in the update.

$$\Delta w_{i,j}^{n,n-1}(t) = \eta \delta_i^n x_j^{n-1} \alpha \Delta w_{i,j}^{n,n-1}(t-1), \quad (4)$$

$$\Delta h_i^n(t) = \eta \delta_i^n + \alpha \Delta h_i^n(t-1). \quad (5)$$

2.2. Neural network rule extraction

The neural network rule extraction techniques attempt to open the neural network black box and generate symbolic rules with the same predictive power as the neural network itself. In the decomposition algorithm such as the Neurorule, starts extracting rules at the level of the individual hidden and output units by analyzing the activation values, weights, and biases. Decompositional approaches then typically treat the hidden units as threshold units.

For example, the algorithm of the Neurorule is summarized as follows [1].
Step 1.Train a neural network to meet the prespecified accuracy requirement.
step 2. Remove the redundant connections in the network by pruning while maintaining its accuracy.
Step 3. Discretize the hidden unit activation values of the pruned network by clustering.
Step 4. Extract rules that describe the discretized hidden unit activation values in terms of the networks inputs.
Step 5. Generate rules describe the discretized hidden unit activation values in terms of the network inputs.
Step 6. Merge the two sets of rules generated in Step4 and 5 to obtain a set of rules that relates the inputs and outputs of the network.

2.3. Problems in neural network rule extraction

Even though the decompositional approach play a role to assess the relevant rules using the hidden layer on the basis of activation values, the procedure included in Step 5 through 6 is not straight forward.

In Neurorules, we must at first descretize the level of signal in hidden layers, and then also examine the relation between the input signal and discretized level in hidden layers. Even more, in the extraction process of rules, we must employ the conventional method presented by Quinlann in a multiplicative way. The overall algorithm become to be complicated to obtain the expression for rules.

If the rule extraction after discretizing the signal is automatized, then it is expected that we can easily obtain relevant rules on the basis of routine works. Therefore, we introduce the GP procedure in place of Step 4 through Step 6 in Neurorule algorithm.

The overview of the GP procedure for generating rules is summarized as follows.
Step 1.Train a neural network to meet the prespecified accuracy requirement.
step 2. Remove the redundant connections in the network by pruning while maintaining its accuracy.
Step 3. Discretize the hidden unit activation values of the pruned network by clustering.
Step 4. Generate rules based on the GP using the binary representation as the terminal variables (statements) in the logical expression.

3. Production rules and the GP

3.1. Tree representation of production rules

For simplicity, we start with the GP operations on the arithmetic expressions [4]-[7]. The prefix representation is equivalent to the tree representation where the external points(leaves) of the tree is labeled with terminals (i.e., constants and variables), and the root of the tree is labeled with the primitive function such as binomial operation $+, -, \times, /$, and the operation taking the square root of variable. For example, we have the next prefix representation.

$$(6.43 \times x_1 - x_2) \times (x_3 - 3.54) \rightarrow - \times 6.43 x_1 x_2 - x_3 3.54. \quad (6)$$

The equation represented by using the prefix are interpreted based upon the stack operation.

We must ensure that after initialization, crossover operation and mutation, we have a valid representation of a tree. For this purpose, the so-called stack count (denoted as $StackCount$)is useful [5]-[7]. The $StackCount$ is the number of arguments it places on minus the number of arguments it takes off the stack. The cumulative $StackCount$ never becomes positive until we reach the end at which point the overall sum still needs to be 1. Any two loci on the two parents' genomes can serve as crossover points as long as the ongoing $StackCount$ just before those points is the same. The crossover operation creates new offspring by exchanging sub-trees between two parents.

Before applying the genetic operation, we must evaluate the ability of each individual (tree structure). The ability is called as the fitness. Usually, we calculate the root mean square error between prediction and observation, and the fitness S_i of i th individual is defined as the inversion of the error.

We iteratively perform the following steps until the termination criterion has been satisfied.
(Step 1)

Generate an initial population of random composition of the functions and the terminals of the problem (constants and variables).
(Step 2)

Execute each program (evaluation of system equation) in the population and assign it a fitness value using the fitness measure. Then, sort the individuals according to the fitness S_i.
(Step 3)

The operations are applied to the individuals chosen with a probability $p_i$based on the fitness. The probability p_i is defined for ith individual as follows.

$$p_i = (S_i - S_{min}) / \sum_{i=1}^{N} (S_i - S_{min}). \quad (7)$$

where S_{min} is the minimum value of S_i, and N is the population size.
(Step 4)

Create new individuals (offsprings) from two existing ones by genetically recombining randomly chosen parts of two existing individuals using the crossover operation applied at a randomly chosen crossover point.
(Step 5)

If the result designation is obtained by the GP (the maximum value of the fitness become larger than the prescribed value), then terminate the algorithm, otherwise go to Step 2.

We apply the global and local mutation operations defined in [5]-[7].

3.2. Applying the GP to rule generation

We can define the condition part of a rule as a logical expression which is represented as a connection of statements with logical operators including AND,OR,NOT. The expression is the same as the arithmetic expression using prefix representation by replacing the operands by the statements, and the arithmetic operators by the logical operators.

Statement can be defined as a connection of two arithmetic expressions (equations) with comparative operators, including $>, \geq, <, \leq, =, \neq$. Then, the rule extraction using the neural networks helps us to avoid the statement generation. By pruning the connections (weights) in neural networks, we have finally several binary expressions used for the input to the simplified neural networks. Then, these binary expressions are used as the statements included in the logical expressions in the GP.

We also define the fitness of individuals as the accuracy of rule generated by the rule corresponds to the underlying individual. To improve the fitness of individuals in the arithmetic expressions (individuals in second level pool of individuals), we apply the GP operations to the logical expressions.

Finally, we can know the logical value of the whole logical formula (individual) by applying the logical operations among statements.

4. Application

4.1. Prediction of bankruptcy

There have been a fair number of previous studies for predicting corporate failure (bankruptcy) using probabilistic estimate such as logit model, multi-variate discriminant analysis (MDA), and neural networks [8]. In this section, we apply the neural network rule extraction based on the GP method proposed in the paper to derive the linguistic rules to predict the bankruptcy.

The population are restricted by (i) the period from 1956 to 1986 in Japan, (ii)the equity of the company had to have been traded on some stock market to exclude small firms, (iii) the company must be classified as an industrial.

In the next step, we must also gather the data for non-bankruptcy firms to specify the binary sample space as well as in ordinary MDA. Our two samples of firms consist of 13 bankrupt firms and a matched samples of 13 non-bankruptcy entities. The latter are matched to the failed group by industry and year of the data. The pair of firms should exist in the fiscal year just before the bankruptcy.

The next phase is one of actually collecting financial data for the bankruptcy firms. Each report has to include the balance sheet, income statement, fund statement. We have at first prepared 29 financial ratios (vector of input variables) for the prediction of bankruptcy. However, it is noted that a number of variables obey very disformed distribution among firms, and are seemed to be irrelevant for statistical inference. Then, these variable are removed from the pool of input variables.

Then, we have 13 variables to predict the bankruptcy, but the number is relatively large to derive concise prediction rule in an linguistic manner. Therefore, we applied the principal component analysis to these 13 variables, and determined 5 variables for prediction. Then, we apply the binary prediction for learning using the back-propagation to derive the meaningful statement included in the production rules.

Table 1 summarizes the (linguistic) rules to predict the bankruptcy compared with conventional MDA and neural network method. As is seen from these tables, the prediction of bankruptcy using the production rules is almost the same as another methods.

Table 1: Prediction of bankruptcy (true classification%

category	GP method	NN	MDA
failed	11(84%)	12(92%)	12(92%)
sound	11(84%)	11(84%)	11(84%)

Table 2: Result of inference (% means true classification)

rating	GP-method	NN	MDA
A	22(73%)	24(80%)	23(76%)
B	24(75%)	27(84%)	27(84%)
C	19(70%)	21(77%)	21(77%)

4.2. Application to bond rating

Industrial corporate bonds have been assigned quality rating since the early 1990s. Then, the neural network rule extraction based on the GP proposed in the paper is applied to derive linguistic rules so that the inference using the input variables is available to discuss the credit worthiness of firms.

At first, we select three groups of Japanese companies[14][15]. 1) 35 companies from electric machinery industry (group 1) 2)30 companies from mechanical machinery industry (group 2) 3)55 companies from several industries. (Group 3) 4) all 120 companies included in Group 1, 2 and 3 (group 4)

We select these companies from three industries so that we can prove the robustness of the rule. It is observed that the distribution of the financial ratio between another industries are usually different. Before utilizing these financial ratios, we must examine the statistical proportions of the financial ratios.

The financial data are screened by factoring analyzing data to identify basically independent dimensions of variables. Factor analysis is associated with numerical weights (factor loadings) which show the degree of involvement of variables. The final set of variable for the analysis contains variables associated only five of these dimensions.

The information of rating for the companies are obtained from the published data of Japan Credit Rating Agency. Even though, the range of rating varies from AAA to CCC, usually it is very rare to categorize the company to the lower ranks such as CCC or BBB. The Japanese companies treated here are relatively good, and the range of rating is included in AAA, AAA, A and BBB. The output is categorized into these three categories.

Table 2 show the result of simulation. As is seen from the result, the correct recognition for rating is around 70 %.

5. Conclusion

This paper treated the use of neural network rule extraction techniques based on the Genetic Programming (GP). We used the neural network and binary classification to obtain simple and relevant classification rules by pruning weight among neurons to obtain simple but substantial binary expressions which are used as statements is classification rules generated by the GP. The method was applied to extract rules to prediction of bankruptcy and the classification of corporate bonds (rating) by using the financial indicators.

For further works, there exist the extension of the method to various rule extraction problems and to slightly complicated functional forms rather than fundamental arithmetic. Further works will be done by the authors.

References

[1] B.Baesens, R.Setino,C.Mues and J.Vanthienen, “Using neural netowrk rule extraction and decision tables for credit-risk evaluation”, Management Science, vol.49,no.3, pp.312–329,2003.

[2] R.Setino and H.Liu, “Symbolic representation for neural networks”, IEEE Computer, vol.23,no.3,pp.71–77,1996.

[3] J.R.Quinlan, “Induction of Decision Tree”, Machine Learning,1,pp.81–106,1986.

[4] J.R.Koza, Genetic Programming, MIT Press, 1992.

[5] Y. Ikeda and S.Tokinaga, “Approximation of chaotic dynamics by using smaller number of data based upon the genetic programming”, Trans. IEICE, vol.E83-A,no.8, pp.1599–1607, 2000.

[6] Y. Ikeda and S.Tokinaga, “Controlling the chaotic dynamics by using approximated system equations obtained by the genetic programming”, Trans. IEICE, vol.E84-A,no.9, pp.2118–2127, 2001.

[7] X.Chen and S.Tokinaga, “Synthesis of multi-agent systems based on the co-evolutionary genetic Programming and its applications to the analysis of artificial markets”, Trans. IEICE,vol.E86-A,vol.10,pp.1038–1048,2003.

[8] E.I.Altman,Corporate Bankrapcy in America, D.C.Health and Company ,1971.