A New Neural Data Analysis Approach Using Ensemble Neural Network Rule Extraction

Atsushi Hara[1] and Yoichi Hayashi[2]

[1] Fujitsu Social Science Laboratory Ltd.
Nakahara-ku, Kawasaki 211-0063, Japan
toohsk@gmail.com
[2] Department of Computer Science
Meiji University
Tama-ku, Kawasaki 214-8571, Japan
hayashiy@cs.meiji.ac.jp

Abstract. In this paper, we propose the Ensemble-Recursive-Rule eXtraction (E-Re-RX) algorithm, which is a rule extraction method from ensemble neural networks. We demonstrate that the use of ensemble neural networks produces higher recognition accuracy than individual neural networks and the extracted rules are more comprehensible. E-Re-RX algorithm is an effective rule extraction algorithm for dealing with data sets that mix discrete and continuous attributes. In this algorithm, primary rules are generated as well as secondary rules to handleonlythoseinstances that do not satisfy the primary rules, and then these rules are integrated. We show that this reduces the complexity of using multiple neural networks. This method achieves extremely high recognition rates, even with multiclass problems.

Keywords: Ensemble neural network rule extraction, Re-Rx Algorithm, Ensemble method, Recursive neural network rule extraction.

1 Introduction

In recent years, neural networks have received increasing attention as a data mining technology, because they are known to be an effective method for real-world classification problems. To increase recognition accuracy, a number of proposals for ensemble neural networks, which involve multiple individual neural networks, have been published [3], [4], [9], [10]. Though neural network ensembles are multiple individual neural networks, they present their own problems: their complexity is greater, rule extraction is more difficult, and they use more computing resources than are necessary. To be sure, algorithms that use bagging [1] or boosting [2] in the C4.5 algorithm have been presented, but these do not directly address the problem.

In this paper, we address the problems of extracting rules from ensemble neural networks, and reduce the number of neural networks to avoid wasteful use of computer resources. Toward that goal, we suggest the Ensemble-Recursive-Rule Extraction (E-Re-RX) algorithm [12]. The extracted rules maintain the high

A.E.P. Villa et al. (Eds.): ICANN 2012, Part I, LNCS 7552, pp. 515–522, 2012.

recognition capabilities of a neural network while expressing highly comprehensible rules. To test these results, we conducted experiments using the CARD data set, the German Credit data set, the Thyroid data set, and the Page Block data set, which can be obtained from the UCI repository [11].

2 Structure of the E-Re-RX Algorithm [12]

2.1 Neural Network Ensembles

The proposed ensemble neural network is constructed from two neural networks. The first obtains the "primary rules" from the learning data set LD' extracted at random from the full learning data set. The second network relearns from learning data set LDf which instances are unclassified by the primary rules. The composition of the two neural networks is shown schematically in Fig.1.

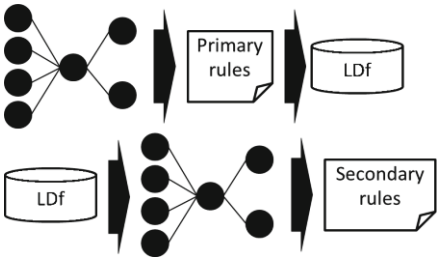

Fig. 1. Basic process for extraction of primary and secondary rules of high accuracy from two neural networks

2.2 Re-RX Algorithm

The Re-RX algorithm [5], [8] is designed to generate classification rules from data sets that have both discrete and continuous attributes. The algorithm is recursive in nature and generates hierarchical rules. The rule conditions for the discrete attributes are disjointed from those for the continuous attributes. The continuous attributes only appear in the conditions of the rules lowest in the hierarchy.

The outline of the algorithm is as follows:

Algorithm Re-RX(S, D, C)
Input: A set of data samples S having discrete attributes D and continuous attributes C.
Output: A set of classification rules.

1. Train and prune a neural network using the data set S and all its D and C attributes.
2. Let D' and C' be the sets of discrete and continuous attributes, respectively, still present in the network, and let S' be the set of data samples correctly classified by the pruned network.

3. If D' has associated continuous attributes C', generate a hyperplane to split the samples in S' according to the values of the continuous attributes C', then stop. Otherwise, using only the discrete attributes D', generate the set of classification rules R for data set S'.

   For each rule Ri generated:

   If support(Ri)>δ_1 and error(Ri)>δ_2, then

   - Let Si be the set of data samples that satisfy the condition of rule Riand let Di be the set of discrete attributes that do not appear in the rule condition of Ri.
   - If D' has associated continuous attributes C', generate a hyperplane to split the samples in Si according to values of the continuous attributes Ci, then stop. Otherwise, call Re-RX(Si, Di, Ci).

2.3 Integration of Rules

Extracting rules from multiple neural networks is assumed to increase the number of rules, and some of these extracted rules may be redundant or irrelevant as classification rules. The accuracy of the rules is maintained and their number is reduced by integrating the primary and secondary rules in accordance with the attributes.

In this paper, the rules extracted from the Re-RX algorithm use the decision tree formed by the J4.8.

First, compare generated rules in R and Rf, if the type and value of the multiple attribute are same, then those two rules should be integrated as one rule.

Second, the multiple attributes of the same type and value are integrated into rules having more attribute types; that is, primary and secondary rules are integrated into finer rules. In judgments made with a decision tree, rules having a larger number of attribute types are considered to be more accurately classified. In this paper, when rules are integrated, a reduced rule number is achieved by integration into finer rules.

That said, in some instances a primary rule will contradict a secondary rule. Here, the instance that generates the secondary rule is judged to differ depending on the hyperplane and the associated rule generated during primary rule training. Making this distinction can adequately explain cases in which contradictory rules are extracted. In contradictory rules, the attributes and values are exactly the same but the class labels differ, or the class labels and attributes that appear in the rules are the same, but the attribute values conflict. In these cases, the rules that appear in the secondary rules are integrated into the primary rules. The decision is made based on the number of samples in the running data set. More specifically, the running data set LD' has a greater number of samples than the running data set used to generate the secondary rules for the LDf. So, the primary rules encompass more samples. This means that when comparing primary rules and secondary rules on the basis of reliability, the primary rules can be regarded as having greater reliability. This is why the secondary rules are integrated into the primary rules.

2.4 Ensemble-Re-RX (E-Re-RX) Algorithm

The essentials of the E-Re-RX algorithm are outlined as follows.

Algorithm E-Re-RX
Input: Training sets LD' and LDf
Output: Integrated rule set obtained by integration of R and Rf.

1. Extract at random an arbitrary proportion of instances from the learning set LD and designate this randomly extracted learning set LD'.
2. Perform training and pruning of LD' with the first neural network.
3. Apply the Re-RX algorithm to the output of Step 2 to obtain the R.
4. Based on these primary rules, generate the set LDf consisting of instances that do not satisfy these rules.
5. Train and prune LDf with the second neural network.
6. Apply the Re-RX algorithm to the output of Step 5 in order to output the Rf.
7. Integrate the primary and secondary rules.

In the proposed E-Re-RX algorithm, we first produce the learning data set LD', which is necessary for training the first neural network. LD' is the set of instances extracted at random in an arbitrary proportion from the full learning data set LD. LD' is input into a neural network having one node in its hidden layer, the neural network is trained and pruned [6], and the rules are extracted. In this paper, we restrict ourselves to back propagation neural networks with one hidden layer because such networks have been shown to possess a universal approximation property [7]. An effective neural network pruning algorithm is a crucial component of any neural network rule extraction algorithm. From these extracted rules, we re-extract rules in accordance with the Re-RX algorithm, and take the final rule set as the primary rules R.

We divide LD into data sets that do and do not conform to these primary rules. The set of instances that do not conform are taken as LDf, which is input into the second neural network. The second neural network is similarly trained on LDf and pruned, rules are extracted, and rules are then re-extracted in accordance with the Re-RX algorithm. These extracted rules are the secondary rules Rf.

Integrated rules are obtained from extracted rules R and Rf as explained in 2.3. With the neural network ensemble, it is possible to determine the final output by integrating the outputs of the multiple neural networks. With the E-Re-RX algorithm, it is possible to determine the overall final output by integrating the rules. This rule integration enables the reduction of the number of neural networks and irrelevant rules.

3 Results

We performed all experiments on the CARD3, German Credit, Thyroid, and Pageblock data using the same methods. All of this data is publicly available from the UCI repository.

Our experimental results are shown in Table 1, which shows the recognition rates for the data sets, and Table 2, which shows the number of rules. Our E-Re-RX results are presented with reference to Hara et al. [12]. Our Re-RX results are presented with reference to Setiono et al. [5], [8]. Our results for MNNEX, PITS, Neural Network Bagging, and Neural Network AdaBoosting are presented with reference to Akhand et al. [9].We conducted each of the E-Re-RX experiments using $\delta_1 = \delta_2 = 0.05$, except for the German Credit experiments, where we used $\delta_1 = \delta_2 = 0.09$. Finally, we would show the rule set which after integrated what we get in thyroid and page block data set are followed.

Rule set in thyroid data set:

R1: if D10=1 then,
  R1a: if D3=0 then predict Class 2.
  R1b: if D3=1 then predict Class 3.
R2: if D3=0 and D10=0 and D2=0 then,
  R2a: if C17≦0.005 then predict Class 3.
  R2b: if C17>0.005 then predict Class 2.
R3: if D3=0 and D10=0 and D2=1 then,
  R3a: if C17≦0.004 then predict Class 3.
  R3b: if C17>0.004 then predict Class 2.
R4: if C21≦0.064 then predict Class 1.
R5: if C21>0.064 then predict Class 3.

Rule set in page block data set:

R1: if D1≦4 then predict Class 2.
R2: if D1>4 and D1≦6 and D10≦3 then predict Class 2.
R3: if D1>4 and D1≦6 and D10>3 and D2≦138 then predict Class 1.
R4: if D1>4 and D1≦6 and D10>3 and D2>138 then predict Class 2.
R5: if D1>6 then predict Class 1.
R6: if D2≦3 then predict Class 4.
R7: if D2>3 and D1≦85 then predict Class5.

Table 1. Comparison of recognition accuracy levels achieved by various methods with each data set

	E-Re-RX	Re-RX	MNNEC	PITS	NN Bagging	NN AdaBoosting
Card1	94.14%	89.53%	-	-	-	-
Card3	94.77%	88.95%	-	-	-	-
German Credit	82.20%	80.48%	76.48%	77.52%	75.60%	75.12%
Thyroid	99.64%	-	-	94.20%	94.11%	96.72%
Pageblock	95.25%	-	-	-	92.58%	96.30%

Table 2. Number of rules resulting from the E-Re-RX experiment compared to the that resulting from the Re-RX experiment

	Card1	Card3	German Credit	Thyroid	Pageblock
E-Re-RX	7	14	17	8	9
Re-RX	13	7	41	-	-

R8: if D2>3 and D1>85 and D2≦34 then predict Class 4.
R9: if D2>3 and D1>85 and D2>34 then predict Class 5.

4 Discussion

In these experiments, we found the recognition accuracy offered by the Re-RX algorithm [5], [8] to be more than sufficient for Card1 and Card3. We believe this can be explained by the fact that we partitioned the learning data set and worked with only part of it. In short, by working with a partial data set, we reduced the number of instances that lead to overfitting, while keeping a number just high enough for the primary rule set. Next, using only the data that did not satisfy the primary rule set, we performed another round of rule extraction, which was able to extract rules that could not have been extracted from the learning data set and produced a high recognition rate. A comparison of the number of rules showed that using an ensemble neural network sometimes increases the number of rules. However, by integrating rules, we were able to eliminate redundant rules, which seem to hold the level of redundancy to a minimum.

Next, when applied to the German Credit data, we were able to obtain the highest level of recognition accuracy with E-Re-RX. In particular, we exceeded the Re-RX algorithm by 2%, although it has a high recognition accuracy, and reduced the number of rules by a significant 40%. Here again, we attribute the difference to working with a partial learning data set. Using few instances, as in LD' made it possible to classify those instances with few rules. Including a few instances in LDf likewise resulted in fewer extracted rules. Accordingly, it can be seen that even after rule integration, our method resulted in fewer rules extracted than did the Re-RX algorithm.

Likewise, regarding the Thyroid data set, we achieved results better than the existing methods: 5% better Neural Network Bagging, and 3% better than Neural Network Adaboosting. We attribute this to the fact that we extracted Class 2 and Class 3 rules in the primary rule set, and Class 1 and Class 3 rules in the secondary rule set. The Thyroid data set consists almost entirely of instances that are Class 3, so in ordinary fitting, a bias will exist toward fitting to Class 3, and so it is assumed to sometimes be impossible to fit to Class 1 and Class 2 correctly. Akhand et al. [9] reported a maximum value of 94.81%, but found that in the absence of a bias toward Class 3, changes in accuracy improvements were dictated by the extent to which Class 1 and Class 2 were fitted accurately. In our research, we were able to extract Class 1 and Class 3 rules from partial data, and the instances that did not satisfy these rules were in many cases Class 2. We were able to fit to Class 2 efficiently this way

and to extract rules with high recognition accuracy, even in a multiclass problem, resulting in accuracy that is much higher than that of previous methods.

Finally, regarding the Pageblock data set, our results were poorer than Neural Network Adaboosting, but 3% better than Neural Network Bagging. Here, for reasons that are the same as in the Thyroid data set, dealing with a partial data set allowed us to extract Class 1 and Class 2 rules in the primary rule set, and in the secondary rule set, we were able to extract Class 4 and Class 5 rules, which can be seen to have high recognition accuracy. However, we were unable to extract Class 3 rules in this experiment. This can be attributed to the fact that Class 3 instances made up no more than 0.5% of the total data set. In other words, because Class 3 exerted little influence on the weight correction, it was not possible to classify by Class 3. It is thought this can be improved by preparing a data set consisting of instances that do not satisfy the primary or secondary rule sets and performing fitting and rule extraction to extract rules related to Class 3. This suggests that an effective approach to multiclass problems would be an iterative method in which instances that do not satisfy rules are gathered into a new data set, on which fitting and rule extraction are performed iteratively until rules that identify all of the classes have been extracted. This implies a more intensive use of computing resources, but because it avoids the problem of setting up an unknown number of neural networks, we still consider it to be an effective method.

5 Conclusion

In this study, we set out to address three problems in ensemble neural networks: to increase recognition rates, to extract rules from ensemble neural networks, and to minimize the use of computing resources. We used a minimal ensemble neural network consisting of two neural networks, which enabled high recognition accuracy and the extraction of comprehensible rules. Furthermore, this enabled rule extraction resulting in fewer rules than previously proposed methods. The results make it possible for the output from an ensemble neural network to be in the form of rules, thus breaking open the black box of ensemble neural networks. Ensemble neural networks promise a new approach to neural data analysis, and we are confident that these results will make data mining more useful, and will increase the opportunities to use data mining with high recognition accuracy.

References

1. Breiman, L.: Bagging predictors. Mach. Learn. 24, 123–140 (1996)
2. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proc. of the Thirteenth International Conference on Machine Learning, Bari, Italy, pp. 148–156 (1996)
3. Hayashi, Y., Setiono, R.: Combining neural network predictions for medical diagnosis. Comput. Biol. Med. 32, 237–246 (2002)
4. Hartono, P.: Ensemble of linear experts as an interpretable piecewise linear classifier. ICIC Exp. Lett. 2, 295–303 (2008)

5. Setiono, R., Baesens, B., Mues, C.: A note on knowledge discovery using neural networks and its application to credit card screening. Eur. J. Oper. Res. 192, 326–332 (2009)
6. Setiono, R.: A penalty-function approach for pruning feedforward neural networks. Neural Comp. 9(1), 185–204 (1997)
7. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
8. Setiono, R., Baesens, B., Mues, C.: Recursive neural network rule extraction for data with mixed attributes. IEEE Trans. Neural Netw. 19, 299–307 (2008)
9. Akhand, M.A.H., Murase, K.: Neural Network Ensembles. Lambert Academic Publishing (LAP) (2010)
10. Alpaydin, E.: Multiple Neural Networks and Weighted Voting. IEEE Trans. on Pattern Recognition 2, 29–32 (1992)
11. University of California, Irvine Machine Learning Repository, `http://archive.ics.uci.edu/ml/`
12. Hara, A., Hayashi, Y.: Ensemble neural network rule extraction using Re-RX algorithm. In: Proc. of IJCNN 2012 (2012) (under review)