Neural network rule extraction for credit scoring

Bart Baesens[a], Rudy Setiono[b], Valerina De Lille[a], Stijn Viaene[a], and Jan Vanthienen[a]

[a] *Department of Applied Economic Sciences*
K.U.Leuven, Naamsestraat 69, B - 3000 Leuven, Belgium
E-mail: {Bart.Baesens; Stijn.Viaene; Jan.Vanthienen}@econ.kuleuven.ac.be
Valerina.Delille@student.kuleuven.ac.be

[b] *Department of Information Systems*
National University of Singapore, Kent Ridge, Singapore 119260, Republic of Singapore
E-mail: Rudys@comp.nus.edu.sg

Abstract

In this paper, we evaluate and contrast four neural network rule extraction approaches for credit scoring. Experiments are carried out on three real life credit scoring data sets. Both the continuous and the discretised versions of all data sets are analysed. The rule extraction algorithms, Neurolinear, Neurorule, Trepan and Nefclass, have different characteristics with respect to their perception of the neural network and their way of representing the generated rules or knowledge. It is shown that Neurolinear, Neurorule and Trepan are able to extract very concise rule sets or trees with a high predictive accuracy when compared to classical decision tree (rule) induction algorithms like C4.5(rules). Especially Neurorule extracted easy to understand and powerful propositional if - then rules for all discretised data sets. Hence, the Neurorule algorithm may offer a viable alternative for rule generation and knowledge discovery in the domain of credit scoring.

Keywords:

Neural Networks, Rule Extraction, Credit Scoring

Introduction

Neural networks have shown to be very powerful pattern recognition techniques for classification in a variety of domains. Unfortunately, one of the most important drawbacks of using neural networks is their opacity. The latter refers to the fact that they do not allow formalisation of the relationship between the outputs and the inputs in a comprehensible way. Neural networks are then often described as black box techniques because they generate complex mathematical models which relate the outputs to the inputs using a set of weights, biases and activation functions which are hard for humans to interpret.

In the field of credit scoring, many successful neural network applications have already been reported in the literature e.g. [1]. Most of these applications primarily focus at developing models with high predictive accuracy without paying attention to explaining the classifications being made. Nevertheless, the latter is believed to play a pivotal role in the credit granting process. Besides having accurate neural network scoring models, the expert also wants to be able to explain why a credit applicant has been classified as either bad or good. This problem may be solved by using rule extraction techniques which try to open the neural network black box and extract symbolic rules (or trees) with the same predictive power as the neural network itself. The extracted rule sets may then be used to build advanced credit scoring expert systems which assist the expert in making his credit granting decisions.

In this paper, we will evaluate and contrast four popular neural network rule extraction techniques, Neurolinear, Neurorule, Trepan and Nefclass, for the domain of credit scoring. These rule extraction algorithms have different characteristics with respect to their perception of the neural network and their way of representing rules or trees. The experiments will be carried out on three real life financial credit scoring data sets. Both the continuous and the discretised versions of these data sets will be analysed. The results will be compared with C4.5 and C4.5rules in terms of predictive accuracy and conciseness of the generated rule sets or trees.

Neural Network rule extraction

Overview

Andrews, Diederich and Tickle [2] propose a classification scheme for neural network rule extraction techniques based

on various criteria. In this paper, we will mainly focus on two dimensions when discussing the examined algorithms: the translucency of the rule extraction algorithm and the expressive power of the extracted rules.

The translucency criterion considers the technique's perception of the neural network. A decompositional approach starts extracting rules at the level of the individual hidden and output units by analysing the activation values, weights and biases. Decompositional approaches then typically approximate the hidden units as threshold units. A pedagogical algorithm considers the trained neural network as a "black box". Instead of looking at the internal structure of the network, these algorithms directly extract rules which relate the inputs and outputs of the network. These techniques typically use the trained network to classify examples and to generate additional "artificial" examples which are then used by a symbolic learning algorithm to infer the rules.

The expressive power of the extracted rules depends on the language used to express the rules. Propositional if - then rules are implications of the form *if X=a and Y=b then Class=1*. An example of a fuzzy classification rule is: *if X is low and Y is medium then Class=1*, whereby low and medium are fuzzy sets with corresponding membership functions. The M- of - N rule *if 2 of (X=a,Y=b,Z=c) then Class=1* is logically equivalent to *if ((X=a and Y=b) or (X=a and Z=c) or (Y=b and Z=c)) then Class=1*. Finally, oblique rules represent piece - wise discriminant functions and are usually represented as follows: *if* $c_1X+c_2Y>c_3$ *then Class=1* whereby $c_1, c_2; c_3 \in R$.

Neurolinear and Neurorule

Neurolinear and Neurorule are algorithms that extract rules from trained 3 - layered feedforward neural networks. Both techniques share the following common steps [3, 4]:

1. Train a neural network to meet the prespecified accuracy requirement.

2. Remove the redundant connections in the network by pruning while maintaining its accuracy.

3. Discretise the hidden unit activation values of the pruned network by clustering.

4. Extract rules that describe the network outputs in terms of the discretised hidden unit activation values.

5. Generate rules that describe the discretised hidden unit activation values in terms of the network inputs.

6. Merge the two sets of rules generated in steps 4 and 5 to obtain a set of rules that relates the inputs and outputs of the network.

Both techniques differ in their way of preprocessing the data. Neurorule assumes the data are discretised and represented as binary inputs using the thermometer encoding [3]. The latter facilitates the process of generating propositional if - then rules. Neurolinear generates oblique rules representing piece - wise linear discriminant functions which are not necessarily axis - parallel as is the case for the rules induced by C4.5 [5]. It can work with continuous input data which are typically normalised to the interval [- 1,1].

Both techniques use an augmented regularised cross - entropy error function to train the neural network. The networks are trained using the BFGS method which is a modified version of the quasi - Newton algorithm. The trained networks are subsequently pruned by inspecting the magnitude of the weights. Using pruned neural networks to extract rules results in more concise rule sets which are easier to interpret.

Neurolinear and Neurorule are decompositional rule extraction techniques extracting oblique and propositional rules, respectively.

Trepan

Like many decision tree induction algorithms, Trepan considers neural network rule extraction as an inductive learning task [6]. It works by querying trained neural networks to induce a decision tree which represents the concept learned by the neural network. Trepan maintains a queue of leaves, which are expanded into sub - trees as they are removed from the queue. The trees are grown using a best - first expansion. The node at which there is greatest potential to increase the fidelity of the extracted tree with respect to the trained network will be the preferred node.

One major drawback of conventional decision tree induction algorithms like C4.5 is that the number of training examples available at a tree - node decreases with the depth of the tree. Hence, splits near the leaves of the tree may often be poorly chosen due to the insufficient number of training examples. Trepan overcomes this by enriching the original training data with additional, artificial training examples. In fact, these artificial training examples are generated for every tree node in which the number of original training examples is less then S_{min}, where S_{min} is a user - specified parameter of the algorithm. The artificial examples are generated taking into account all previously selected splits that lie on the path from the root of the tree to the current node. Furthermore, the generation process also takes into account the distributions of the individual features which are modeled using frequency counts for discrete - valued features and a kernel density estimation method for continuous features. The trained neural network is then used as an oracle which answers queries about class membership of the original and the artificial training examples.

Trepan induces both binary and M - of - N type of splits. The M - of - N splits are constructed using a hill - climbing search process. First, the best binary split is selected according to the gain ratio criterion which is also the criterion used in C4.5 [5]. This binary split is then used as a seed to construct M - of - N type of splits using again the gain ratio criterion and two operators discussed in [6]. Tree expansion is stopped using a statistical test to decide whether or not a node covers only instances of a single class or when a prespecified limit on the number of internal tree nodes is reached.

Trepan is to be considered as a pedagogical algorithm

extracting decision trees from trained neural networks with arbitrary architecture.

Nefclass

The Nefclass (Neuro Fuzzy Classification) algorithm developed by Nauck and his coworkers [7] aims at extracting interpretable fuzzy classifiers from data. Nefclass is based on a three - layer feedforward fuzzy perceptron whereby the first layer represents input variables, the hidden layer units represent fuzzy rules and the third layer represents (crisp) output classes. Nefclass starts with an empty network having as many input neurons as there are inputs in the data set, zero hidden neurons and as many output neurons as there are classes in the data set. For each input, fuzzy sets are defined modelling linguistic concepts e.g. small, medium and large. These fuzzy sets may have different types of membership functions (e.g. trapezoidal, triangular, bell - shaped).

A Nefclass system can be created with or without insertion of prior knowledge in the form of fuzzy rules. When the classifier is created from scratch, a three - phase learning mechanism is used. In the first step, an initial rule base is constructed whereby hidden units are added until all training patterns have been covered by at least one rule. The best rules (hidden units) are then retained using a heuristic (e.g. the best k created rules). In the second step, the membership functions are trained using a variant of the well - known error backpropagation algorithm. In the third step, Nefclass offers the possibility to prune the rule base by removing rules and variables, based on a simple greedy algorithm which uses several heuristics (e.g. correlation and redundancy). This pruning step is fully automated without the need for user interaction. The goal of this pruning is to improve the comprehensibility of the created classifier. Note that the second and third steps are typically executed iteratively i.e. after each pruning step, the membership functions are trained again.

Nefclass can be seen as a decompositional technique, because the rule base is built by examining the internal structure of the fuzzy perceptron. The output of Nefclass is a set of fuzzy rules, which can easily be interpreted since they can be expressed in linguistic terms.

Empirical Evaluation

Data sets and experimental set up

All neural network rule extraction techniques were applied to three real life credit scoring data sets. Two data sets were obtained from major Benelux financial institutions. The other data set is the Statlog German credit data set. The inputs include socio - demographic variables as well as loan specific information such as the purpose of the loan and its term. For the Benelux data sets, a bad loan was defined as a loan whereby the customer had missed three consecutive months of payments. Each data set was split into 2/3 training set and 1/3 test set. Table 1 displays the characteristics of the three data sets.

We conduct experiments using both the continuous and the discretised versions of all data sets. The discretisation is done using the discretisation algorithm of Fayyad and Irani [8]. We include C4.5 and C4.5rules as a performance benchmark [5]. Both the classification accuracy and the complexity of the generated trees or rules are compared and discussed. Complexity for the trees is measured by the number of leaf nodes and the total number of nodes. We also evaluate the fidelity of the rule extraction approaches defined as the percentage of test set examples which the rule extraction technique and the neural network classify in the same way.

We start with extracting rules using Neurolinear and Neurorule for the continuous and discretised data sets, respectively. All neural networks have hyperbolic tangent hidden units and linear output units. We use two output units and the class is assigned to the output unit with the highest activation value (winner take all learning). The neural networks are trained and pruned according to the set up discussed above. The same pruned networks are also used to extract trees using Trepan. Since Trepan is a pedagogical tree extraction algorithm, we can apply it to any trained neural network with arbitrary architecture. This allows us to make a fair comparison between a decompositional approach (Neurorule and Neurolinear) and a pedagogical approach (Trepan). Following Craven [6], we set the parameters for the Trepan analyses as follows: $S_{min} = 1000$ and the maximum tree size to 15 internal nodes. Finally, we also included Nefclass as an example of a neurofuzzy rule extraction system. We defined three fuzzy sets for each variable which were modelled using triangular or bell - shaped membership functions. We set the maximum number of fuzzy rules to 100.

Results for the continuous credit scoring data sets

Table 2 presents the results of applying C4.5, C4.5rules, Neurolinear, Trepan and Nefclass to the three continuous credit scoring data sets. Both the classification accuracy as measured by the percentage correctly classified (PCC) and the complexity of the generated rules or trees are depicted. Table 2 clearly indicates that the rules and trees extracted by Neurolinear and Trepan are both powerful and very concise when compared to C4.5rules and C4.5. Neurolinear yields the best absolute performance for all three data sets with a maximum of three oblique rules for the Bene1 data set. For the German credit data set, it performed significantly better than C4.5rules according to McNemar's test. Furthermore, Neurolinear obtained a significantly better performance than Nefclass on all three data sets. Nefclass was never able to extract compact and powerful fuzzy rule sets for any of the data sets.

The neural networks used by Neurolinear and Trepan have 1 hidden unit for the German credit and Bene2 data set and 2 hidden units for the Bene1 data set. After pruning, 16 inputs remained for the German credit data set, 17 inputs for the Bene1 data set and 23 inputs for the Bene2 data set. Neurolinear obtained 100% test set fidelity for all three data sets. The test set fidelity of Trepan with respect to the

Table 1 - Characteristics of credit scoring data sets

	Number of inputs	Data set size	Training set size	Test set size
German Credit	20	1000	666	334
Bene1	33	3123	2082	1041
Bene2	33	7190	4793	2397

neural networks which were also used by Neurolinear is 91,31%, 87,60% and 85,31% for the German credit, Bene1 and Bene2 data set, respectively. This clearly indicates that Neurolinear was able to extract rule sets which better reflect the decision process of the trained neural networks than the trees inferred by Trepan. Note that for all three data sets Trepan has a better performance than C4.5 with much fewer leaves and nodes.

In summary, Neurolinear was able to extract compact rule sets with a high predictive accuracy on all continuous data sets. The trees induced by Trepan are concise and also give satisfactory classification performance. It has to be noted that the rules extracted by Neurolinear are oblique rules which may still be hard for humans to interpret. Hence, in the following subsection we will discretise all data sets and use Neurorule to extract classical, propositional if - then rules which might be easier understood by the credit scoring expert.

Results for the discretised credit scoring data sets

Table 3 presents the results of applying C4.5, C4.5rules, Neurorule, Trepan and Nefclass to all discretised data sets. Note that, in general, the accuracy of C4.5, C4.5rules, Trepan and Nefclass increased on all discretised test sets. Neurolinear obtained a higher accuracy on the continuous German credit and Bene1 data sets than Neurorule on their discretised counterparts. Furthermore, it may be concluded from Table 3 that Neurorule is able to extract very concise rule sets with a high classification accuracy on the test set for all three discretised data sets. Again, Nefclass was never able to infer compact and powerful fuzzy rule sets. Also notice that for all three data sets, Trepan has a better performance than C4.5 with much fewer nodes and leaves.

All the neural networks that were used by both Neurorule and Trepan had one hidden unit. After binarisation using the thermometer encoding and pruning, 6 inputs remained for the German credit data set, 5 inputs for the Bene1 data set and 11 inputs for the Bene2 data set. Neurorule obtained a test set fidelity of 100% on all discretised data sets. For the German credit and Bene1 data set, the test set fidelity for Trepan was 100% whereas for the Bene2 data set it was 97,7%. These are clearly higher fidelity rates than for the continuous data sets and indicate that both Neurorule and Trepan are able to accurately approximate the decision process of the neural networks trained on the discretised data sets.

In summary, Neurorule extracted concise rule sets with a very good predictive accuracy on all discretised data sets. Also Trepan gave good results in terms of classification accuracy and tree complexity. The rules extracted by Neurorule are classical propositional if - then rules which might be easily understood and used by the credit scoring expert. These rules may then be used to build expert systems to aid the credit scoring expert in the credit granting process. Decision tables may offer a viable and interesting alternative to represent these rules in a user - friendly expert system [9].

Conclusion

In this paper, we evaluated and contrasted four popular neural network rule extraction techniques, Neurolinear, Neurorule, Trepan and Nefclass for credit scoring purposes. Experiments were conducted on three real life credit scoring data sets. Both the continuous and the discretised versions of all three data sets were analysed. The results were compared with the decision trees and rules induced by the popular C4.5 algorithm. Both the conciseness and the classification accuracy of the generated rules or trees were investigated. Our experimental results have shown that Neurolinear, Neurorule and Trepan are able to extract compact and comprehensible rules and trees with a high predictive accuracy. Especially Neurorule extracted easy to understand and powerful propositional if - then rules for all discretised data sets. Hence, the Neurorule algorithm may offer an interesting and viable alternative to classical decision tree and rule induction algorithms for credit scoring. Future research is needed to further investigate the generated rule sets and build expert systems using e.g. decision tables.

References

[1] D. West 2000. Neural network credit scoring models. *Computers & Operations Research* 27(11 - 12):1131-1152.

[2] R. Andrews, J. Diederich, and A.B. Tickle 1995. A survey and critique of techniques for extracting rules from trained neural networks. *Knowledge Based Systems* 8(6):373-389.

[3] R. Setiono and H. Liu 1996. Symbolic representation of neural networks. *IEEE Computer* 29(3):71-77.

[4] R. Setiono and H. Liu 1997. Neurolinear: from neural networks to oblique decision rules. *Neurocomputing* 17(1):1-24.

[5] J.R. Quinlan 1993. *C4.5: programs for machine learning* Morgan Kaufmann Publishers.

[6] M.W. Craven and J.W. Shavlik 1996. Extracting tree - structured representations of trained networks. In Advances in Neural Information Processing Systems, volume 8, Cambridge, MA, MIT Press.

[7] D. Nauck, F. Klawonn, and R. Kruse 1997. *Foundations of Neuro - Fuzzy Systems*. Wiley, Chichester.

[8] U.M. Fayyad and K.B. Irani 1993. Multi - interval discretization of continuous - valued attributes for classification learning. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, 1022-1029. San Francisco, CA, Morgan Kaufmann.

[9] J. Vanthienen and G. Wets 1994. From decision tables to expert system shells. *Data and Knowledge Engineering* 13(3):265-282.

Table 2 – Results for continuous credit scoring data sets

Data set		PCC_{train}	PCC_{test}	Complexity
German credit	C4.5	82.58	70.96	37 leaves, 59 nodes
	C4.5rules	81.53	70.66	13 propositional rules
	Neurolinear	80.93	77.25	2 oblique rules
	Trepan	75.97	73.35	6 leaves, 11 nodes
	Nefclass	71.32	70.36	16 fuzzy rules
Bene1	C4.5	89.91	68.68	168 leaves, 335 nodes
	C4.5rules	78.63	70.80	21 propositional rules
	Neurolinear	77.43	72.72	3 oblique rules
	Trepan	73.29	70.60	12 leaves, 23 nodes
	Nefclass	67.53	66.19	8 fuzzy rules
Bene2	C4.5	90.24	70.09	849 leaves, 1161 nodes
	C4.5rules	77.61	73.00	30 propositional rules
	Neurolinear	76.05	73.51	2 oblique rules
	Trepan	73.36	71.84	4 leaves, 7 nodes
	Nefclass	69.43	69.25	2 fuzzy rules

Table 3 – Results for discretised credit scoring data sets

Data set		PCC_{train}	PCC_{test}	Complexity
German credit	C4.5	80.63	71.56	38 leaves, 54 nodes
	C4.5rules	81.38	74.25	17 propositional rules
	Neurorule	76.13	75.15	7 propositional rules
	Trepan	75.38	73.95	3 leaves, 5 nodes
	Nefclass	73.57	73.65	14 fuzzy rules
Bene1	C4.5	78.29	68.68	57 leaves, 98 nodes
	C4.5rules	77.23	70.12	19 propositional rules
	Neurorule	72.38	71.47	7 propositional rules
	Trepan	72.38	71.47	6 leaves, 11 nodes
	Nefclass	69.84	69.07	3 fuzzy rules
Bene2	C4.5	82.64	73.22	438 leaves, 578 nodes
	C4.5rules	77.76	73.51	27 propositional rules
	Neurorule	75.82	74.30	17 propositional rules
	Trepan	74.98	73.46	11 leaves, 21 nodes
	Nefclass	71.60	71.17	5 fuzzy rules