Hybrid Neural Solutions for Automatic Knowledge Discovery from Databases

Bruno P. Amorim[1], Germano C. Vasconcelos[1], Lourdes M. Brasil[2]

[1] *Center of Informatics (CIn), Federal University of Pernambuco (UFPE)*
Recife, PE, Brazil
Email: {bpa, gcv}@cin.ufpe.br

[2] *Post-Graduation Program in Knowledge Management and Information Technology,*
Catholic University of Brasilia (UCB), Brasília, DF, Brazil
lmb@pos.ucb.br

Abstract

Artificial Neural Networks (ANN) have been successfully used in a wide variety of real-world applications. However, ANN alone have not been fully employed in KDD (Knowledge Discovery in Databases) applications because they often produce incomprehensible models. Neuro-fuzzy systems and techniques for symbolic knowledge extraction have been increasingly used to represent the knowledge acquired by ANNs in a comprehensible form. This paper presents hybrid neural solutions for the KDD process, resulting from a detailed experimental investigation of three neural models (MLP, FuNN and FWD), four symbolic knowledge extraction techniques (AREFuNN, REFuNN, TREPAN and FWD) and two feature selection algorithms (FWD and the decision tree extracted by TREPAN). A large scale credit assessment application in a real-world situation was used as the test bed for the experimental investigations carried out. The results demonstrate that the benefits obtained from hybrid neural solutions are actual.

1. Introduction

In the last decades, institutions have experienced a great growth in their capacity of generating and collecting data from their daily operations. However, the traditional methods used to manipulate these data can generate only informative reports. Such methods are unable to analyze the data and automatically extract strategic knowledge. These difficulties contributed to the arising of the field known as Knowledge Discovery in Databases (KDD) [1]. KDD is a discovery process of previously unknown, valid, novel, potentially useful and understandable knowledge from databases [1]. KDD consists of an iterative sequence of data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation and knowledge presentation. A wide variety of algorithms have been proposed for solving each one of these steps. Some of these techniques have been shown to efficiently solve some of the tasks involved but fail to accomplish some of the others. For example, Artificial Neural Networks (ANN) have been successfully used to perform data mining tasks but generally cannot easily present knowledge extracted in a comprehensible way to humans [1].

Intelligent Hybrid Systems (IHS) is an Artificial Intelligence approach based on the idea of integrating different techniques into a whole structure, thus overcoming their limitations [2]. The dissemination of IHS has contributed to the emergence of Hybrid Neural Systems (HNS) whose main research focus has been the integration of ANN, a strongly data based technique, with symbolic techniques, such as Fuzzy Logic and conventional symbolic algorithms. Neuro-fuzzy systems are HNS that combine ANN with fuzzy systems [3]. Other approaches for extracting symbolic knowledge from ANN have also been proposed [4][5].

The goal of this paper is to investigate hybrid neural solutions for the KDD process that are capable not only of performing data mining operations but also have the capacity to explain the knowledge embedded in the network. The proposed solutions resulted from an extensive experimental investigation of the Multi-Layer Perceptron network (MLP) [6]; neuro-fuzzy models FWD (Feature-Weighted Detector) [7] and FuNN (Fuzzy Neural Network) [8], together with their rule extraction techniques; and the TREPAN (Trees Parroting Networks) algorithm [4]. Two feature selection techniques (FWD and the decision tree extracted by TREPAN) were also investigated. The experiments were carried out in a large scale real-world credit risk assessment problem and the results demonstrate that the benefits obtained from hybrid neural solutions are actual.

2. Data mining algorithms

The FWD network [7] was proposed in order to solve simultaneously two major problems in pattern recognition: pattern classification and feature selection. This network attempts to select important features, while maintaining the maximum recognition rate. Moreover, the knowledge acquired by the network can be described as a set of fuzzy rules. Therefore, the FWD model comprises, within the same structure, three steps of the KDD process (data selection, data mining e knowledge presentation).

The FuNN network [8] uses a MLP network and a modified backpropagation training algorithm. Its architecture facilitates the learning from data and approximate reasoning, as well as the fuzzy rule extraction and insertion. It allows for the combination of data and rules into one system, and several methods of adaptation from which the membership functions of the fuzzy predicates and the initial fuzzy rules may adapt and change according to new data.

The MLP network was selected to be employed in the investigation as a reference model to be compared with the models FWD and FuNN and as the oracle for the TREPAN algorithm.

3. Knowledge presentation techniques

The rule extraction technique of the FWD model is performed in a very simple way through the memory connections m_{ji} [7]. Each feature is associated with n memory connections (n is equal to the total of classes) which are used in the fuzzy predicates of the rules (e.g., *nearly* m_{ij}). The rule extraction process produces one rule for each class and all the relevant features must be present in the rules.

REFuNN (Rule Extraction from a FuNN) and AREFuNN (Aggregated Rule Extraction from a FuNN) are two techniques proposed for extraction of fuzzy rules from the FuNN model [9]. These techniques use the architecture and weights of the FuNN network to extract the rules.

The TREPAN technique [4] is an algorithm that makes queries, during the learning process, to an "oracle" using training and artificial examples. The oracle can be any classification algorithm and the answers to the queries are used to build a decision tree that approximates the knowledge represented by the oracle. TREPAN represents the knowledge in a comprehensible way; can be used in applications that have discrete and continuous features; is able to produce succinct decision trees from large networks; is able to produce decision trees that maintain a high level of fidelity with their respective oracles, while being comprehensible and accurate; is scalable with respect to the database size, model complexity and execution time; and does not impose any requirements on either the network architecture or its training method.

4. Experimental investigation

4.1. Problem domain and database

The application of financial credit-risk evaluation was used as the test environment for the experiments carried out. This problem was chosen because it constitutes a large scale, real-life and complex application that provides a means for robust comparison between the algorithms studied. The situation consists of a classification problem that defines whether a credit will be given or not to an applicant. The database used was obtained from a Brazilian financial company and is composed of 27 input features (the database coded to the neural models has 68 input features), 2 classes (good and bad payers) and 60,141 records (48,218 good cases and 11,923 bad cases). The database contains personal and financial data about credit applications and the history of defaulting on the credit approval. The database was divided into three sets (training, validation and test) with sizes of 50%, 25% and 25% of the records, respectively.

4.2. Experimental methodology

Initially, several neural model configurations were analyzed using the same initial weights and the best configuration found (lowest validation MSE – Mean Square Error) was used to perform 30 runs with different initial weights. The stopping criteria used considered the generalization loss (5% - training is stopped when the validation error increases 5% with respect to the smallest error up to the current epoch) and the maximum number of epochs (3000).

In the experiments performed with the FWD model, the memory connections were initialized with random values and the weight connections were set to the unity (1). This model was composed of 68 input nodes, corresponding to the input features, and 2 output nodes, representing the good and bad payer classes.

The topology used for the FuNN model consisted of 68 input nodes (input features), 3 condition nodes (small, medium and large) for each continuous feature and 2 condition nodes (false and true) for each Boolean feature, 2 action nodes, representing the classes, and 1 output node. The weights between the condition layer and rule layer, and the rule layer and action layer were initialized with random values. The weights between the input layer and condition layer were initialized with the values 0, 0.5 and 1 for continuous inputs and 0 and 1 for Boolean inputs. The weights between the action layer and output layer were initialized with the values 0 and 1.

The MLP training was performed using backpropagation with momentum [6]. All the weights were randomly initialized in a network composed of 68 input nodes, 2 output nodes and 1 hidden layer.

Table 1 summarizes the training parameters for all the best model configurations.

Table 1. Training parameters for the best configurations of the neural models

	Temporal learning rate	Fuzziness	Learning rate	
FWD	0.1	0.7	0.5	
	Number of rule nodes	Momentum	Learning rate	Gain coefficient
FuNN	10	0.9	0.001	1
	Number of hidden nodes	Momentum	Learning rate	Gain coefficient
MLP	2	0.9	0.01	1

4.3. Generalization performance

4.3.1. Test accuracy. Table 2 shows the classification results for each of the neural models. The values in the table represent the means obtained after 30 runs of the best configurations. This table shows the classification rate for each class and the total classification, together with the associated standard deviation.

Table 2. Test set accuracy

	Average (%)			Standard deviation		
Model	Total	Good	Bad	Total	Good	Bad
FWD	67.32	71.63	49.90	0.00	0.00	0.00
FuNN	65.83	68.36	55.62	1.15	2.03	2.83
MLP	66.44	69.41	54.43	1.17	2.13	2.79

The neural models presented a very similar test set accuracy with the biggest difference observed in the classification rate of the bad payer class.

4.3.2. ROC curves. In classification problems it is usual to analyze the generalization performance of the learning model considering the test accuracy only. As a result, this measure is very used to compare the performance of several models. However, comparisons based on the test accuracy omits two important aspects that must be considered, especially in real-world problems: usually the class distribution can not be precisely specified and the costs associated with the types of error (Type I and Type II) can be different and can change over time. For this reason, to improve a comparative analysis between the models, it is necessary to apply some technique that is able to determine the best model independently of the class distribution and costs associated with the types of error. The technique selected to this analysis was the ROC (Receiver Operating Characteristics) curves [10].

The ROC curves show the relation of the false positive rate with the true positive rate varying according to a threshold applied to the model outputs. The curves make possible a visual comparison of a set of models as well. A point in the ROC curve is better than another point if it is in a higher northwestern position. Figure 1 depicts the ROC curves for the models studied.

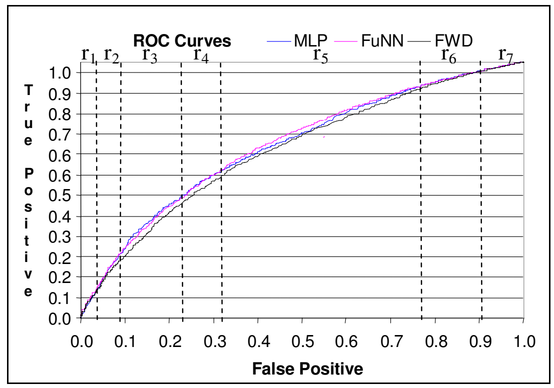

Figure 1. ROC curves for the neural models

The graph with the ROC curves was divided into seven regions (r_1,r_2,...,r_7) for better analysis. In the regions r_1 and r_7, the model curves are overlaped, denoting similar model performance. In the regions r_2, r_4 and r_6, howerver, the results with the MLP and FuNN models are superior to those observed with the FWD model, because their curves are in a higher northwestern location. In the region r_3, the MLP model is better than the models FuNN and FWD. In the region r_5, the FuNN model presents the best performance among the models. In none of the regions, the FWD model can be elected as the best classifier.

4.3.3. Portfolio of maintained clients. A complementary aspect of practical importance to the application domain of credit risk assessment is to observe how the model decision affects the number of clients (both good and bad) in the company's client portfolio when a credit concession decision is taken (approval or rejection). This can be obtained by considering the continuous network responses and varying a threshold (from 0 to 100) as the cutpoint for credit approval. If an applicant receives a score (network response) above the threshold then its credit application is approved, otherwise it is rejected. This procedure allows a comparison between the hybrid neural models themselves and in relation to the previous classification decision taken by the financial company. Two curves are shown in the graph. The first represents the total number of clients (both good and bad) maintained in the portfolio as a function of the cutpoint, whereas the second accounts for the rate of bad payers reduced in the portfolio. The interpretation of the results is that if the curve of the bad payers decreases more consistently that the curve of the total number of clients it means that the classifier is being able to detect and reject many more bad clients than it is producing possible losses of good payers, when compared to the previous method for credit evaluation. In other words, the bad case curve is always below the curve of the total number of clients maintained in the portfolio.

Figures 2, 3 and 4 show the curves of the clients maintained in the portfolio for the models MLP, FuNN and FWD, respectively. It can be observed that in none of the regions the curve of the bad payers is above the curve of the total number of clients, which means that the neural models presented a performance superior to the criterion used by the company to classify the clients.

These results demonstrate that the application of the neural models to the company decision process would result in a better decision, especially when considering that the cost of accepting a bad payer is typically much higher than that of losing a good payer.

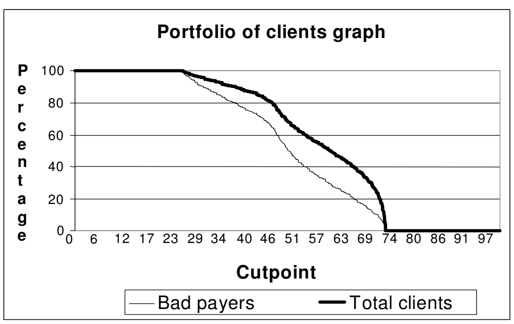

Figure 2. Portfolio of clients with the MLP model

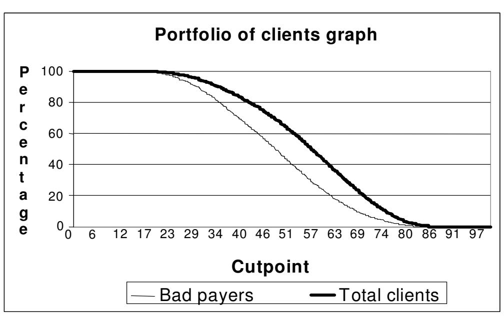

Figure 3. Portfolio of clients with the FuNN model

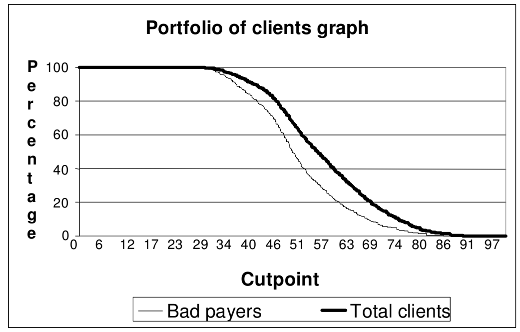

Figure 4. Portfolio of clients with the FWD model

4.4. Feature selection

Two feature selection techniques were selected to be applied in conjunction with the neural models studied. The first method is the feature selection process embedded in the FWD which is accomplished from the analysis of the weight connection values that represent the feature relevance (from 0 to 1) for each class. The features with a relevance higher than a certain threshold (0.3 in the experiments carried out) are taken as relevant. The application of this procedure to the original 68 features resulted in the selection of 26 final relevant features. The second feature selection method makes use of the TREPAN algorithm and is accomplished by defining the features not present in the tree as irrelevant [1]. From the original database composed of 27 features only 8 were selected as relevant. Table 3 shows the test set accuracy of the neural models after feature selection with the two methods examined. Since the test set accuracy has increased in some of the models, these results demonstrated the effectiveness of the feature selection techniques investigated.

Table 3. Test set accuracy after the feature selection

Technique	Model	Average (%)			Standard deviation		
		Total	Good	Bad	Total	Good	Bad
FWD	FWD	67.32	72.10	47.99	0.00	0.00	0.01
	FuNN	66.04	68.84	54.72	1.16	2.16	3.02
	MLP	66.54	69.72	53.68	0.37	0.70	1.04
Using the decision tree	FWD	65.13	67.58	55.19	0.57	0.89	1.07
	FuNN	65.00	67.75	53.89	0.00	0.00	0.00
	MLP	65.15	67.70	54.82	0.66	1.26	2.08

4.5. Symbolic knowledge extraction

After studying the classification accuracy of the neural models, the knowledge extraction capacity of the techniques was evaluated [11]. Table 4 shows the rules extracted from the FWD model. Although the FWD ability to remove irrelevant features contributed to obtaining simpler rules, the FWD rule extraction technique produced extensive rules for the high dimensional credit concession problem. This is because all the relevant features must be present in the rules. Another problem in this model is the semantic representation of Boolean features. According to [7], since the linguistic term *nearly* is associated with each feature, all the features are manipulated as numerical and the semantic representation of Boolean variables becomes inappropriate.

Table 4. Rules extracted from the FWD model

IF sex is *nearly 0.587* AND $marital\_status_1$ is *nearly 0.285* AND … num_additional_cards is *nearly 0.537*
THEN client is good payer

IF sex is *nearly 0.135* AND $marital\_status_1$ is *nearly 0.731* AND ... num_additional_cards is *nearly 0.763*
THEN client is bad payer

Table 5 shows some examples of rules extracted by the AREFuNN and REFuNN techniques applied to the FuNN model. Two types of rules are shown: weighted and simple. In Case 1, degrees of importance and certainty are associated with the conditions and conclusion, respectively. In Case 2, the degrees are omitted.

Table 5. Examples of rules extracted by REFuNN and AREFuNN

Case 1 IF $marital\_status_1$ is *false* (0,81) AND
$marital\_status_4$ is *false* (0,77) AND
$residencial\_city_2$ is *false* (0,53) AND
$residencial\_ddd_1$ is *false* (0,64) AND
zip_code_1 is *high* (0,85) AND
zip_code_2 is *medium* (0,63) AND
$type\_client_1$ is *true* (0,85) AND
spouse_income is *high* (0,66) AND
income is *small* (0,82)
THEN client is *good payer* (0,62)

Case 2 IF $marital\_status_5$ is *true* AND
$residencial\_ddd_4$ is *false* AND
income is *small* AND
zip_code_2 is *medium* AND
employment_time is *small*
THEN client is *bad payer*

REFuNN extracted 5 rules and AREFuNN extracted 9 rules. Although the rule set extracted by AREFuNN was larger than the set extracted by REFuNN, the rules obtained by AREFuNN are simpler. Despite theses techniques use thresholds to simplify the rules, the extracted rule sets were larger than that extracted from the FWD network. This characteristic is not necessarily a disadvantage because the rule set size is not the only aspect to be considered. In order to make the comprehension easier, it is also important that a small number of conditions per rule are generated (a characteristic not observed in the FWD method).

The knowledge extracted by TREPAN can be represented in two forms: decision tree and if-then rules. Initially, TREPAN extracts a decision tree. The main advantage of this representation is the capacity of visually presenting the mined knowledge, making comprehension and application easier. From the decision tree, a rule set can be obtained. Table 6 shows some examples of the rules obtained from the decision tree extracted by TREPAN from the MLP network.

Table 6. Examples of rules extracted by TREPAN

IF age > 0.26 AND flag_residencial_phone ≠ 0
THEN client is good payer (support = 39.99%, confidence = 87.16%)

IF age <= 0.26 AND sex ≠ 0 AND flag_residencial_phone = 0
THEN client is bad payer (support = 2.12%, confidence = 38.09%)

A small and very comprehensible tree was produced from which 13 rules were derived. Although the rule set given by TREPAN was larger than the set extracted by the other techniques, the number of conditions per rule is small and the rule application is very direct.

The fidelity rate between the decision tree and the MLP network and the accuracy of the tree on the test set are showed in the Table 7. The fidelity measures the percentage of examples whose decision tree classification is equal to the network classification.

Table 7. Test set results for the TREPAN technique

Accuracy (%)			Fidelity (%)		
Total	Good	Bad	Total	Good	Bad
66.51	71.20	47.52	85.78	88.81	79.14

5. Hybrid neural solutions proposed for the KDD process

Although the FWD model has presented a similar performance to those achieved by the MLP and FuNN models, it has a serious practical limitation: capacity of solving only linearly separable problems. In addition to this, the rules extracted from a FWD model are very extensive, making difficult the comprehension of the extracted knowledge. However, the feature selection ability of the FWD model is a most relevant functionality. In the credit problem examined, which uses a large scale database, 61% of the features were considered irrelevant. These results demonstrated the effectiveness of the FWD feature selection technique, making it a serious candidate for application in the data selection step of the KDD process.

Differently from the FWD, the MLP and FuNN models are able to solve non-linearly separable problems. These models, therefore, are suitable for application in the data mining step. With respect to the rule extraction of the FuNN model, AREFuNN was superior to REFuNN, presenting simpler rules.

The feature selection made with the decision tree extracted by TREPAN presented satisfactory results when applied to the FuNN and MLP models. Considering the aspect of symbolic knowledge extraction, TREPAN produced a very simple decision tree. As a result, the interpretation and direct application of the knowledge extracted are made easier, making TREPAN appropriate for use in both the knowledge presentation step and data selection step.

The observations of the advantages and disadvantages of each technique lead to the proposition of two hybrid neural solutions for the KDD process. In the first solution, the FWD feature selection technique is used in the data selection step. After, the reduced database is applied to the FuNN model as the data mining algorithm. Later, the rule extraction techniques REFuNN and AREFuNN are used individually or unified in the knowledge presentation step. In the second solution, the feature selection technique based on the decision tree defined by TREPAN is used in the data selection step. The MLP and FuNN models are used in the data mining step. In the knowledge presentation step, four alternatives can be applied. The TREPAN technique can be used with the FuNN or MLP models, or the FuNN model can be employed together with the rule extraction techniques REFuNN and AREFuNN (alone or unified).

6. Conclusions

This paper has presented a thorough investigation of several feature selection, data mining and knowledge presentation techniques for the development of hybrid neural solutions for KDD applications. These solutions resulted from a detailed experimental investigation of three neural models (MLP, FuNN and FWD), four symbolic knowledge extraction techniques (AREFuNN, REFuNN, TREPAN and FWD) and two feature selection techniques (FWD and the decision tree extracted by TREPAN). The experiments were performed using a large scale credit assessment database extracted from a real-world operational situation. The experimental investigation provided a practical contribution since it has demonstrated that the use of HNS for accomplishing several steps of a KDD solution is both feasible and attractive. Therefore, HNS can be considered an alternative to the traditional neural models, without performance loss and with the additional functionality of representing knowledge in a comprehensible form.

There are many options for further works. Some examples are the test and validation of the proposed solutions in other real-world problems, and the investigation of other neural models, techniques for feature selection and extraction of symbolic knowledge from ANNs. An additional important research would be the investigation of extensions to the FWD model in order to make it able to solve non-linearly separable problems, to produce more than one rule per class and to support more than one membership function per feature.

References

[1] Han, J., and M. Kamber, *Data Mining: concepts and techniques*, Morgan Kaufmann, 2001

[2] I.G.L. Silva, B.P. Amorim, P.G. Campos, and L.M. Brasil, “Integration of Data Mining and Hybrid Expert System”, In *Fifteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, Florida, 2002, pp. 267-271

[3] Jang, J.R., C. Sun, and E. Mizutani, *Neuro-Fuzzy and soft computing: a computational approach to learning and machine intelligence*, Prentice Hall, 1997

[4] M.W. Craven, and J.W. Shavlik, “Extracting tree-structured representations of trained networks”, In: D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo (eds.): *Advances in Neural Information Processing Systems*, MIT Press, Denver, 1996, pp. 37-45

[5] I.A. Taha, and J. Ghosh, “Symbolic Interpretation of Artificial Neural Networks”, *IEEE Transactions on Knowledge and Data Engineering,* Vol. 11, n. 3, 1999, pp. 448-461

[6] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, “Learning Representations by Backpropagation Errors”, *Nature*, Vol. 323, 1986, pp. 533-536

[7] R. Li, M. Mukaidono, and I.B. Turksen, “A fuzzy neural network for pattern classification and feature selection”, *Fuzzy Sets and Systems*, Vol. 130, 2002, pp. 101-108

[8] N. Kasabov, J.S. Kim, M. Watts, and A. Gray, “FuNN/2 – A Fuzzy Neural Network Architecture for Adaptative Learning and Knowledge Acquisition”, *Information Sciences - Applications*, Vol. 101, n. 3-4, 1997, pp. 155-175

[9] N. Kasabov, J. Kim, R. Kozma, and, T. Cohen, “Rule Extraction from Fuzzy Neural Networks FuNN: A Method and a Real-World Application”, *Journal of Advanced Computational Intelligence*, Vol. 5, n. 4, 2001, pp. 193-200

[10] F. Provost, and T. F. Fawcett, “Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions”. In *Proceedings of the Third Int. Conf. on Knowledge Discovery and Data Mining (KDD-97)*, AAAI Press. Menlo Park, 1997, n. 3, pp. 43-48

[11] B.P. Amorim, G.C. Vasconcelos, and L.M. Brasil, “A Comparative Analysis of Techniques for Extraction of Fuzzy and Boolean Rules from Artificial Neural Networks”, In *International Conference on Fuzzy Systems (AFSS)*, Hanoi, Vietnam, 2004