2002 Student Conference on Research and Development Proceedings, Shah Alam, Malaysia

SCOReD2002

Development of an Adaptive Business Insolvency Classifier Prototype (AVICENA) using Hybrid Intelligent Algorithms

*Azizi Ab Aziz, Assoc. Prof Fadzilah Siraj and Azizi Zakaria

Artificial Intelligence Special Interest Group
School of Information Technology
Universiti Utara Malaysia
06010 Sintok, Kedah, Malaysia
**E-mail: azizi_ab_aziz@hotmail.com*

Abstract - Confronted by an increasingly competitive environment and chaotic economy conditions, businesses are facing with the need to accept greater risk. Businesses do not become insolvent overnight, rather many times creditors, investors and the financial community will receive either direct or indirect indications that a company is experiencing financial distress. Thus, this paper analyzed the ability of AVICENA in classifying business insolvency performance events. Neural networks (Multi layer Perceptron – Backpropagation) serves as a classifier mechanism while Apriori algorithms (Auto Association Rules) supports the decision made by the neural networks, in which rules are generated. The conventional model in predicting business performances, called as Altman- Z Scores model is used for performance comparison.

Keywords - Neural Networks, Business Insolvency, Apriori Algorithms, Discriminant Analysis, Financial Ratios.

1. INTRODUCTION

The essence of business failure is the inability of owners to sustain their business from financial perspective. Thus, faced with insufficient profits or losses over a sustained period, the business owner is left with little choice but to cease operations [2][4]. It is a major concern among managers, stockholders and creditors. Business insolvency modeling is a subject of great interest to financial practitioners and researchers. It allows for timely decision to be made relative to the reallocation of resources more efficiently.

Being able to forecast potential failure provides an early warning mechanism so that an appropriate adjustment in resource allocation can take place[8]. Most of the research works in this area implement statistical model and financial analysis. Comparison with one of this method is done through performance analysis.

This paper will discuss the implementation of hybrid intelligent algorithms as an alternative mean and improving decision-making processes in modeling business insolvency. The paper is organized, as follows, the next section, will highlight classifier techniques, association rules extraction, experimental design and results. Finally, conclusion concluded this paper in the last section.

2. CLASSIFICATION USING MULTI LAYER PERCEPTRON-BACK PROPAGATION

Literally, neural networks are also referred as *neuro-computers* and *connectionist networks.* It is a massively parallel-distributed network processor that has a natural propensity for storing experiential knowledge and making it available for use[7]. *Learning algorithm* is a function of which to modify the synaptic weights of the network in an orderly fashion so as to attain desired design objective[1].

The Multi Layer Perceptron with back-propagation learning algorithm is depicted below (Figure 1.0):

Start:

Initialize input vectors x(n) , weights, output O, learning parameters, error threshold.

Do until termination criterion = True

$$v_j^i(n) = \sum_{i=0}^{p} w_{ji}^{(i)}(n) y_i^{(l-i)}(n)$$

$$y_i^l(n) = 1 \Big/ 1 + e^{(-v_j^{(l)}(n)/c)}$$

if l = l or neuron j is first hidden node then

$y_j^0(n) = x_j(n)$ *else if (l=L) or neuron j is an output layer. then*

$$y_j^L(n) = o_j(n)$$

end if

Error signal, $E_j(n) = d_j(n) - o_j(n)$

$$\delta_j^L = \tfrac{1}{c}\left[e_j^L(n) o_j(n)\left(1 - o_j(n)\right)\right]$$

$$\delta_j^l = \tfrac{1}{c}\left[y_j^l(n)(1 - y_j^l(n) \sum_k \delta_k^{l+1}(n) w_{kj}^{l+1}(n) \right]$$

$$w_{ji}^l(n) = w_{ji}^l(n) + \alpha\left[w_{ji}^l(n) - w_{ji}^l(n-1)\right] + \eta \delta_j^l(n) y_j^l(n) y_i^{i-l}(n)$$

epoch $\rightarrow$ *counter +1*

Loop

End:

Figure 1.0: MLP Backpropagation Algorithm

This algorithm is a *supervised learning* technique where the network is considerably learned sufficiently if the error rate between desired and actual error is minimized. In neural learning adaptation, the aim is to train the network to achieve a "balance" between the ability to recall perfectly pattern that has been learned and to give a reasonably good output responses [5].

3. ASSOCIATION RULES EXTRACTOR USING APRIORI ALGORITHM

The Apriori algorithm detects correlation between two or more fields in a data record, or among sets of values in a single field [2]. The purpose of the algorithm is to discover association rules of the following form:

$$P \Rightarrow Q, i.e: P_1 \wedge P_2 \wedge ... \wedge P_m \Rightarrow Q_1 \wedge Q_2 \wedge .. \wedge Q_n$$

Figure 2.0: General Association Rules

Apriori discovers patterns in the form of so-called *large itemsets (k=1,2,...)*, which is essentially the sets of items that are often associated within individual record in the database. The generated rules are calculated based on minimum support and confidence. Assumed that the association rule given $X \Rightarrow y$ and minimum support and confidence are calculated as depicted below:

$$Support = \frac{\sigma(X \cup y)}{|T|},\ T = total\ transaction$$

Figure 3.0: Support Level

$$Confidence = \frac{\sigma(X \cup y)}{\sigma(X)}$$

Figure 4.0: Confidence Level

Support measures the frequency of the occurencing patterns while *confidence* indicates the strength of the rules implication. The flow of the algorithm is depicted below (Figure 5.0):

F_1= {frequent 1-item sets}
K=2;
    While(F_{k-1} is not empty)
        Call C_k=AprioriGen(F_{K-1})
        For all transaction in T
            Subset (C_k, T)
        End
        F_k={c in C_k s.t c.count >=min.support)
    Loop
  Answer = $\cup sets(F_k)$
Public Function AprioriGen(F(k-1))
    join F_{k-1} with F_k
    $C_1=(i_1,i_2,...,I_{k-1})$ and $C_2=(j_1,j_2,...j_{k-1})$joined if
        $i_p=j_p$ for p = 1 to k-1
    Evaluate min.confidence
    new candidate, C = $(i_1,i_2,... I_{k-1},j_{k-1})$
  Add C into table structure
End Function

Figure 5.0: Apriori Algorithm Pseudo Code

The algorithm started by finding all sets of possible items that have transaction support above minimum support [2]. Itemsets with minimum support are called *large itemsets*. Later, this itemsets are used to generate association rules.

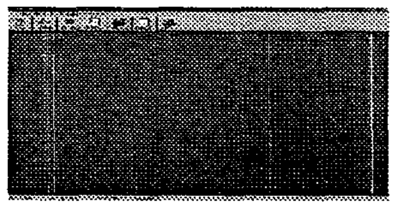
4. PROTOTYPE DESIGN & RESULTS

For experimental works and testing, the prototype named as *AVICENA* is developed using *Visual Basic ver. 6.0.* This prototype runs under Windows platform using Pentium 800 Mhz and 128 RAM.

Figure 6.0: *AVICENA* Main Screen

In this prototype, there are six modules involved:

- Data Entry & Pre-Processing Module
- Adaptive Classifier Module
- Rules-Extraction Module



- Predictor Module
- Administration Support Module
- Reporting Module

In this paper, only three main modules will be discussed (pre-processing, adaptive classifier and rules- extractor).

4.1 Feature Input Vector & Pre-Processing

The financial ratios and macroeconomics data were used as input data. The input parameters are chosen through several references and financial experts advise[1][3][4][8]. These parameters are shown below

- Working Capital/Total Assets
- Retained Earning/Total Assets
- Earning before Income Tax/Total Assets
- Total Sales /Total Assets
- Total Debt / Total Assets
- Inflation Rate
- Inter-Bank Rate
- Operational Size
- National Gross Domestic Product
- Firm's Age
- Types of Industry

Initially, these values are converted through binarization and symbolization processes. Later, these values are then normalized using *Linear Scaling* method:

$$\bar{x} = \left(X_{\min} + (X_{\max} - X_{\min}) * \frac{D - D_{\min}}{D_{\max} - D_{\min}} \right)$$

Figure 7.0: Data Normalization

where D is original dataset and X is pre-processed value.

4.2 Adaptive Classifier Module

This module implements MLP-Backpropagation neural network. The learning parameters (*momentum rate, learning constant, firing angle, error tolerance, sampling, hidden nodes needed and maximum epoch)* are pre-determined by user.

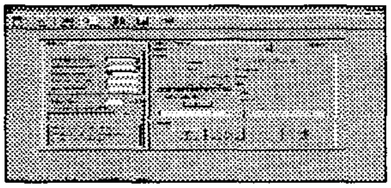


Figure 8.0: Adaptive Classifier GUI

The training iteration is terminated if either maximum epoch or error tolerance achieved. The initial weights are seeded either using *Nguyen-Widrow* normalization or random -0.05 to 0.05 weights initialization. The *Nguyen-Widrow* normalization algorithm is shown in Figure 9.0

Start:

Calculate β ,scale factor for n input nodes and p hidden nodes $\beta = 0.7^{n}\sqrt{p}$

Set random initialization weights value to v_{ij}

Compute Euclidean length of j^{th} column of v, $|v_j|$

Update all the weights, $v_{ij} = \dfrac{\beta v_{ij}(old)}{|v_j|}$

Use a bias input, $v_{oj} = random(-\beta, \beta)$

End:

Figure 9.0: Nguyen-Widrow Normalization

The data is separated into training set and testing set. Training set provides *knowledge* for model's generalization, while testing set is used in validating the generalization performances upon previously unseen data. The accuracy of these sets are measured through:

$$Correct\ Training\ (\%) = \frac{correct\ \ per\ training}{num\ \ of\ \ training\ \ set} \times 100$$

$$Correct\,Testing(\%) = \frac{correct\ per\ testing}{num\ of\ sample - num\ of\ training\ set} \times 100$$

The processing time taken during learning process depends to the complexity of problem space.

4.3 Rules-Extractor Module

This module provides rules-based explanation to the particular classify result. The Apriori algorithm is chosen to chunk out the rules hidden in the financial database.

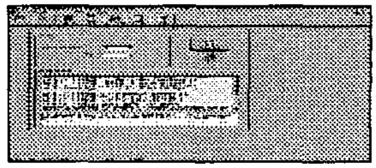


Figure 10. Rule Extractor GUI.

The main purpose of this rule is to support or deny result derived from adaptive classifier module [6]. Initially, during rules formation, the interesting itemsets are converted into binary & symbolical codes in database table. Later, *Syntax-Matching Algorithm* is employed for a better rules interpretation. It enables user to understand how the result was derived.

4.4 Experimental Result

The *AVICENA* is implemented with 300 business data, which consists of 180 healthy firm and 120 distressed firms. Several experiments were executed and the best result profile is shown in Table 1.0

Properties	Result
Dataset	300
Epochs	15000 iterations
Network Topology	11-6-1
Processing Time	20 minutes
Training/Testing size	80% : 20%
Activation Function	Hyperbolic Tangent
Learning Rate	0.12
Momentum	0.56
Error Rate	2.6%
Training Correct	83.3 %
Testing Correct	86.5%
Rules Accuracy	89.3%
Weight Initialization	*Nguyen-Widrow Method*

Table 1.0: The Experimental Result

The predictor module will integrate the neural networks generalized model with rules that has been extracted by Apriori algorithm. For performance comparison purposes, based on prior research works, the Altman Z-Score model is used. Using the same dataset, it shown that *AVICENA* gives a comparable result with 86.5 percent classification accuracy, supported by 89.3 percent of rules accuracy, compared to Altman's Z-Score model with 79.7 percent of classification accuracy.

5. CONCLUSION

Financial related data is ultimately chaotic in nature. The ability of neural networks in discerning chaotic pattern provides a bright future in predicting business insolvency. Integrated with data mining technology, Apriori algorithm is used to generate association rules. These rules will infer on how neural network derived the results. Thus, it will neutralize the risk of wrongly classify cases and eliminate *black-box* nature in neural networks decision making process. The prototype shows promising result in classifying business insolvency.

6.ACKNOWLEDGEMENT

The author would like to convey appreciation to Registrar of Businesses & Companies and Bank Negara Malaysia for data given. Personal thanks to Miss Chua Boon Lay from CAIRO, UTM KL and Dr Simon Scott from APITT for their supports and guidance. Also many thanks to Mr. Mohd. Hafiz Mohd. Hassin for his effort in integrating interface agent to the prototype.

7. REFERENCES

[1]Agarwal.A. *Neural Networks and Their Extensions for Business Decision Making: PhD Dissertation*, College of Applied Mathematics, The Ohio State University (1993).

[2]Agrawal.R and Srikant R. Fast Algorithm for Mining Association Rules, *20th Conference on Very Large Database Analysis*, Santiago, Chile(1994).

[3]Altman.E . *Corporate Financial Distress and Bankruptcy 2nd Edition*, John Willey, New York. (1995).

[4]Anderson.P. *The Economy As An Evolving Complex System*, Addison-Willey, Redwood City, CA (1998).

[5]Aziz .et al. Neural Networks Application in Chaos Theory: Business Failure Prediction Modeling, *2nd Conference of Information Technology in Asia 2001*, Kuching, Sarawak (2001).

[6]Chua B.L .et al. Intelligent Database by Neural Network and Data Mining, *Artificial Intelligence Application in Industry 99*, Kuala Lumpur (1999).

[7]Haykin.S. *Neural Networks: A Comprehensive Foundation*, Springer, New York (1994).

[8]Ismail.S *.Pengenalan Pengurusan Kewangan*, Dewan Bahasa & Pustaka, Kuala Lumpur (1998).